Real Quantifier Elimination Technology and Optimisations via Machine Learning

Matthew England
Coventry University
Centre for Computational Science and Mathematical Modelling

Computational Algebraic Geometry Research Network
University of Warwick
8th March 2024

Speaker supported by EPSRC grant EP/T015748/1.

Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
   - Alternatives to Traditional CAD
2. Optimising CAD via Machine Learning
   - ML for Computer Algebra
   - ML for CAD Variable Ordering
   - Beyond Efficiency Gains
3. Applications (time permitting)
   - Biology
   - Economics

Overview

Summary: the talk will describe the Real Quantifier Elimination Problem and the Cylindrical Algebraic Decomposition method to tackle it. We will introduce what these things are, how machine learning may be used to optimise them, and how explainable AI may be used to gain ideas to improve software without embedding machine learning models.

Key message: potential for fruitful interplay between computational mathematics and machine learning.

Joint work with: Rashid Barket, James Bridge, James Davenport, Tereso del Río, Forian Florescu, Zongyang Huang, Laurence Paulson, and Lynn Pickering.

Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
   - Alternatives to Traditional CAD
2. Optimising CAD via Machine Learning
   - ML for Computer Algebra
   - ML for CAD Variable Ordering
   - Beyond Efficiency Gains
3. Applications (time permitting)
   - Biology
   - Economics

Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
   - Alternatives to Traditional CAD
2. Optimising CAD via Machine Learning
   - ML for Computer Algebra
   - ML for CAD Variable Ordering
   - Beyond Efficiency Gains
3. Applications (time permitting)
   - Biology
   - Economics

Real QE

Real Quantifier Elimination (Real QE)

Given: A quantified formulae (in prenex normal form) whose atoms are (integral) polynomial constraints;
Produce: a quantifier free formula logically equivalent over $\mathbb{R}$.

Fully quantified examples:

In: $\exists x, x^2 + 3x + 1 > 0$
Out: $\texttt{True}$ e.g. when $x = 0$
In: $\forall x, x^2 + 3x + 1 > 0$
Out: $\texttt{False}$ e.g. when $x = -1$
In: $\forall x, x^2 + 1 > 0$
Out: $\texttt{True}$

Partially quantified example:

In: $\forall x, x^2 + bx + 1 > 0$
Out: ???

The answer depends on the unquantified variable, b.

Real QE

Real Quantifier Elimination (Real QE)

Given: A quantified formulae (in prenex normal form) whose atoms are (integral) polynomial constraints;
Produce: a quantifier free formula logically equivalent over $\mathbb{R}$.

Fully quantified examples:

In: $\exists x, x^2 + 3x + 1 > 0$
Out: True e.g. when $x = 0$
In: $\forall x, x^2 + 3x + 1 > 0$
Out: False e.g. when $x = -1$
In: $\forall x, x^2 + 1 > 0$
Out: True

Partially quantified example:

In: $\forall x, x^2 + bx + 1 > 0$
Out: ???

The answer depends on the unquantified variable, b.

Partially quantified Tarski formulae example

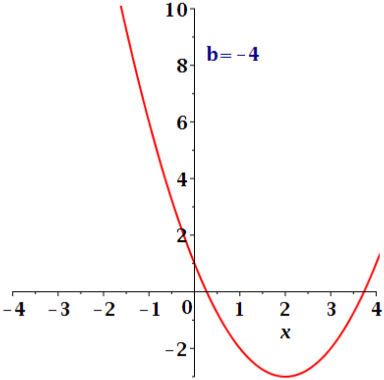
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is } \texttt{True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is } \texttt{False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions automatically.



Partially quantified Tarski formulae example

Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is } \texttt{True} ,$$

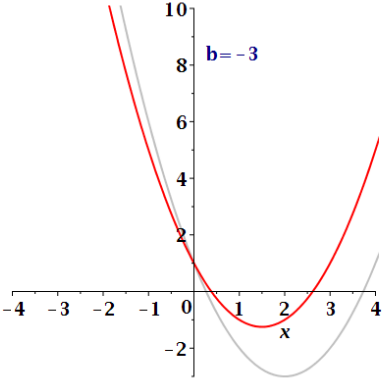
$$\forall x, x^2 + 3x + 1 > 0 \text{ is } \texttt{False} .$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$

Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions automatically.



Partially quantified Tarski formulae example

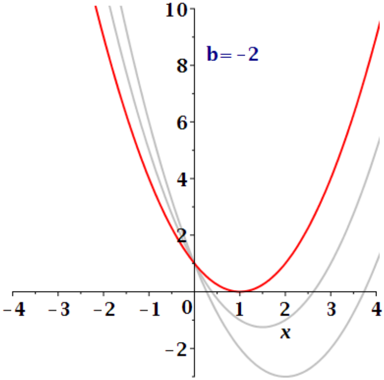
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is } \texttt{True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is } \texttt{False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \land (b < 2)$

The technology we look at today can answer such questions automatically.



Partially quantified Tarski formulae example

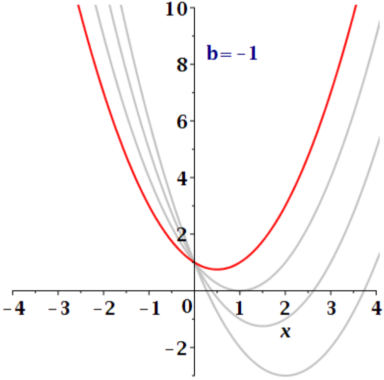
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is } \texttt{True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is } \texttt{False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions automatically.



Partially quantified Tarski formulae example

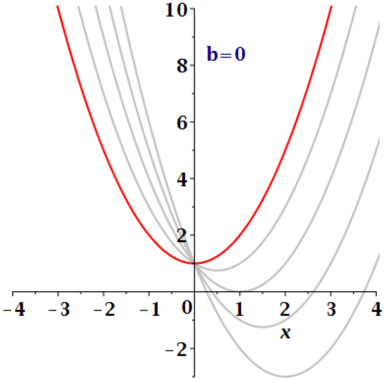
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions automatically.



Partially quantified Tarski formulae example

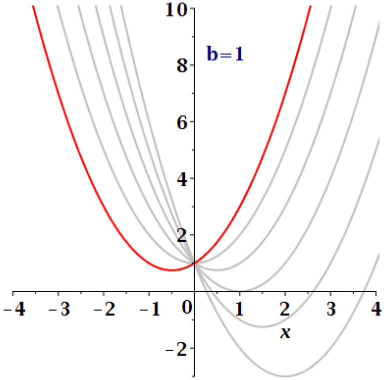
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is } \texttt{True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is } \texttt{False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions automatically.



Partially quantified Tarski formulae example

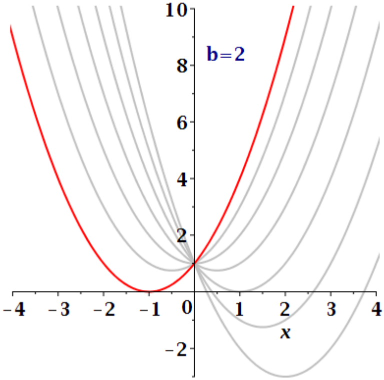
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is } \texttt{True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is } \texttt{False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions automatically.



Partially quantified Tarski formulae example

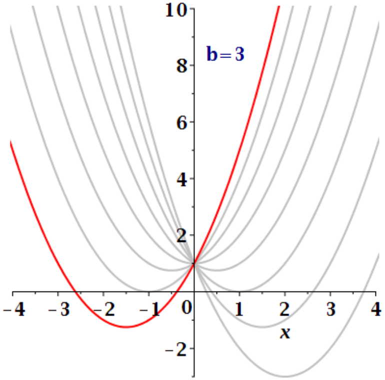
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions automatically.



Partially quantified Tarski formulae example

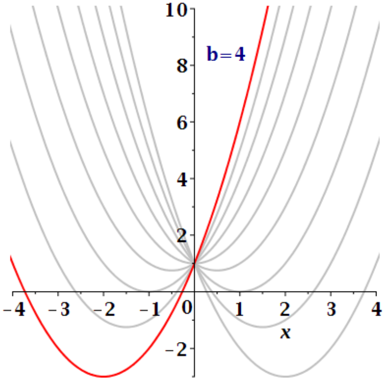
Consider $\forall x, x^2 + bx + 1 > 0$.
When $b = 0$ and $b = 3$:

$$\forall x, x^2 + 1 > 0 \text{ is True},$$
$$\forall x, x^2 + 3x + 1 > 0 \text{ is False}.$$

So in general the answer depends on b.

Input: $\forall x, x^2 + bx + 1 > 0$
Output: $(-2 < b) \wedge (b < 2)$

The technology we look at today can answer such questions automatically.



Numeric vs. Symbolic solution?

The solution in the previous slide was essentially numeric: try what happens for many different values and interpolate a solution.

The alternative is to use **Symbolic Computation:** algorithms and data-structures for manipulating exact mathematical expressions and objects. Traditionally implemented in *Computer Algebra Systems* (e.g. Macaulay2, Maple, Mathematica, Sage).

Advantages of Symbolic Computation:

- Can solve numerically ill-conditioned problems.
- Provide guarantees for safety critical systems.
- Provide fundamental insight into the system.

Disadvantage of Symbolic Computation: much more expensive!

Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
   - Alternatives to Traditional CAD
2. Optimising CAD via Machine Learning
   - ML for Computer Algebra
   - ML for CAD Variable Ordering
   - Beyond Efficiency Gains
3. Applications (time permitting)
   - Biology
   - Economics

Cylindrical Algebraic Decomposition

A **Cylindrical Algebraic Decomposition (CAD)** is:

- a **decomposition** of $\mathbb{R}^n$ into a set of cells C_i (i.e. $\bigcup_i C_i = \mathbb{R}^n$ and $C_i \cap C_j = \emptyset$ if $i \neq j$) such that:
- cells are **semi-algebraic** meaning they may be described by a finite sequence of polynomial constraints;
- cells are **cylindrical** meaning the projection of any two onto a lower coordinate space *in the variable ordering* are identical or disjoint (i.e. the cells in $\mathbb{R}^m$ stack up in cylinders over cells from CAD in $\mathbb{R}^{m-1}$).

G.E. Collins.
Quantifier elimination for real closed fields by cylindrical algebraic decomposition.
In Proc. 2nd GI Conf. Automata Theory and Formal Languages, pp. 134–183. Springer-Verlag, 1975.

CAD Invariance Property

CAD may also refer to an algorithm that produces the CAD object.

The traditional CAD algorithm introduced by Collins in the 1970s takes a set of input polynomials and produces a CAD such that each polynomial has constant sign in each cell: this additional property is called **sign-invariance**.
Such a CAD allows us to uncover properties of polynomials over infinite space by examining finite set of sample points.

Most applications (e.g. QE) actually provide as input a logical formulae built from polynomial constraints and require as output a **truth-invariant CAD**: one such that each formula has constant truth value in each cell.
Such a CAD allows us to find solution sets from the descriptions of true cells: semi-algebraic; easy to visualise and check membership.

CAD Invariance Property

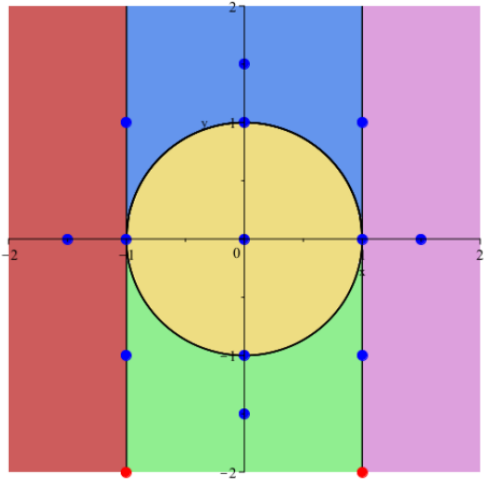
CAD may also refer to an algorithm that produces the CAD object.

The traditional CAD algorithm introduced by Collins in the 1970s takes a set of input polynomials and produces a CAD such that each polynomial has constant sign in each cell: this additional property is called **sign-invariance**.
Such a CAD allows us to uncover properties of polynomials over infinite space by examining finite set of sample points.

Most applications (e.g. QE) actually provide as input a logical formulae built from polynomial constraints and require as output a **truth-invariant CAD**: one such that each formula has constant truth value in each cell.
Such a CAD allows us to find solution sets from the descriptions of true cells: semi-algebraic; easy to visualise and check membership.

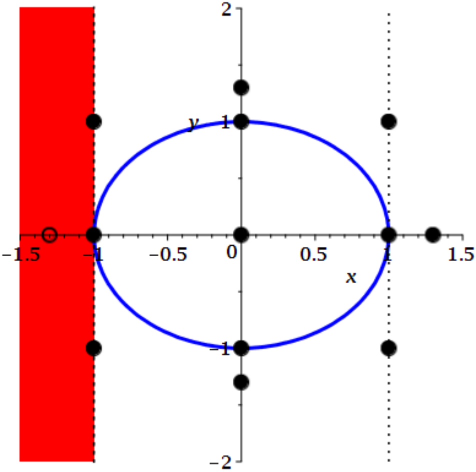
Example: Circle — visualisation

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$,
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



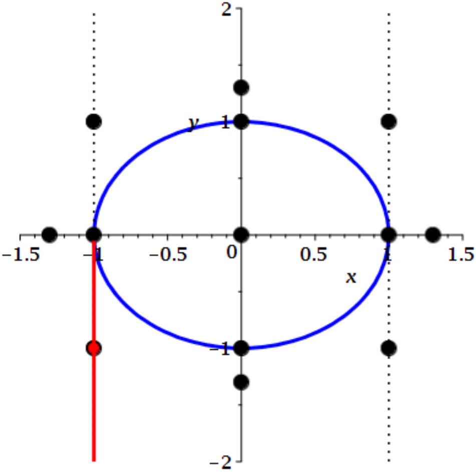
Example: Circle — visualisation

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$, $y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



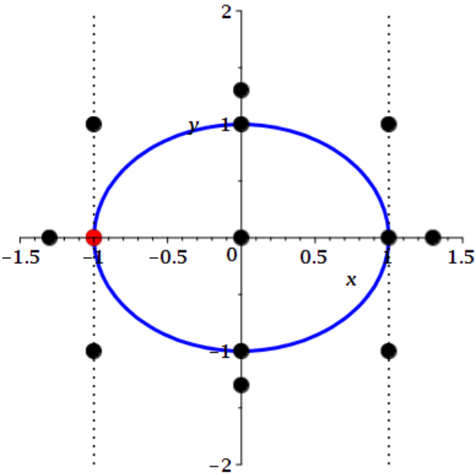
Example: Circle — visualisation

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$, $y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



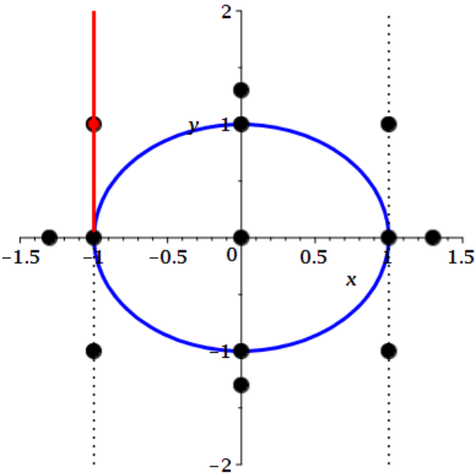
Example: Circle – visualisation

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$,
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



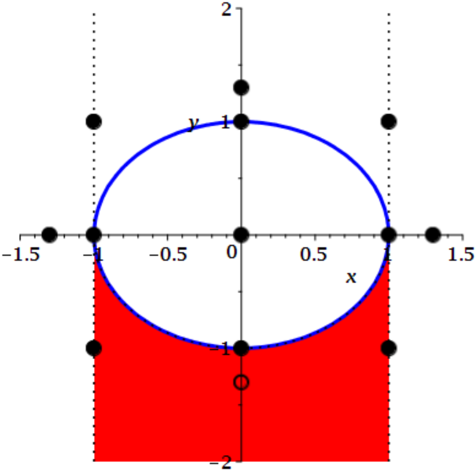
Example: Circle — visualisation

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$, $y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



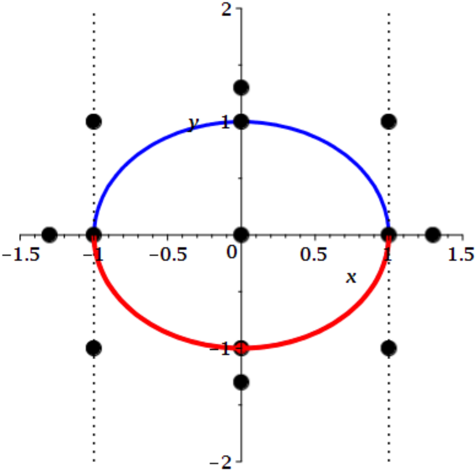
Example: Circle – visualisation

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$, $y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



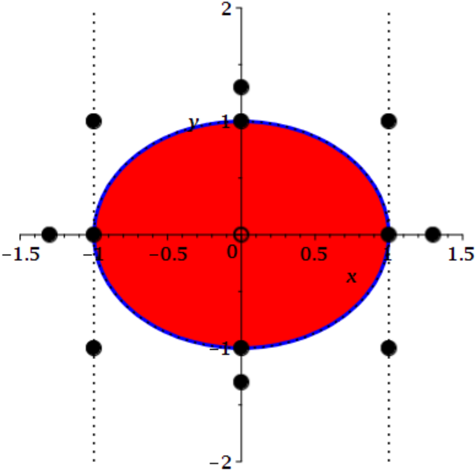
Example: Circle — visualisation

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$, $y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



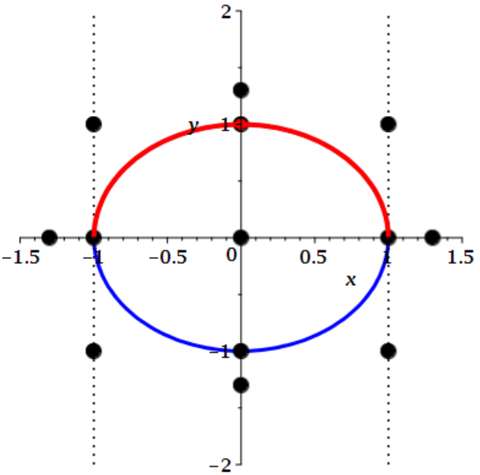
Example: Circle — visualisation

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$,
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



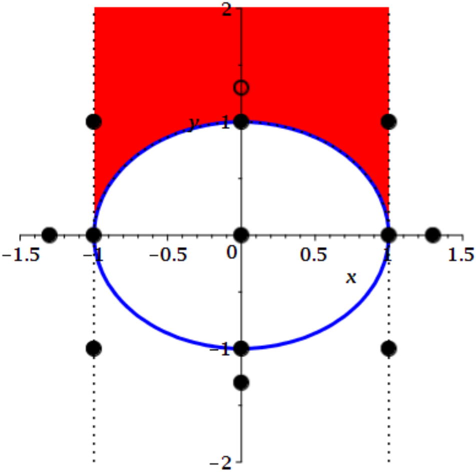
Example: Circle – visualisation

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$, $y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



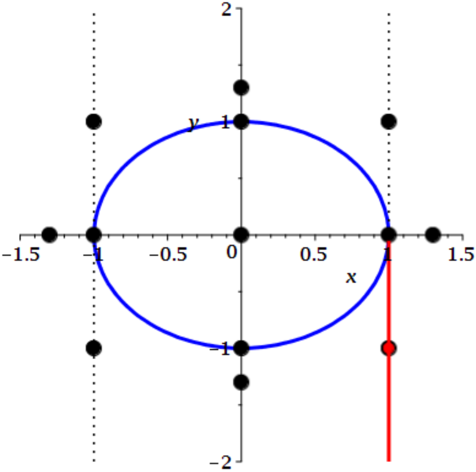
Example: Circle – visualisation

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$,
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



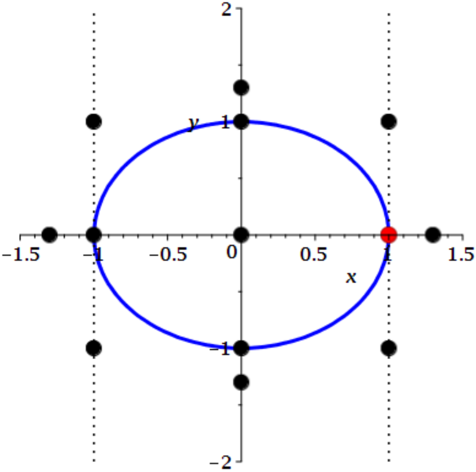
Example: Circle – visualisation

Cell 1: $x < -1,\ y$ free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1,$
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1,$
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1,$
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1,$
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1,$
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1,\ y$ free



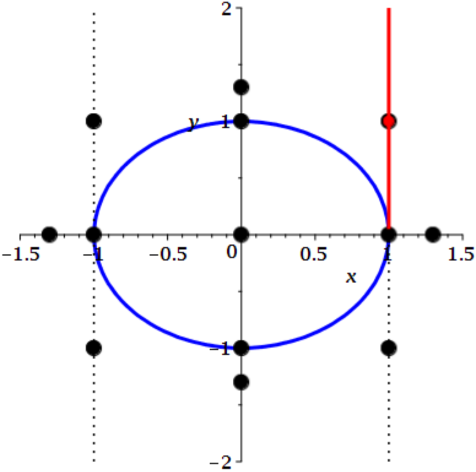
Example: Circle — visualisation

Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$, $y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$, $y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$, $y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



Example: Circle — visualisation

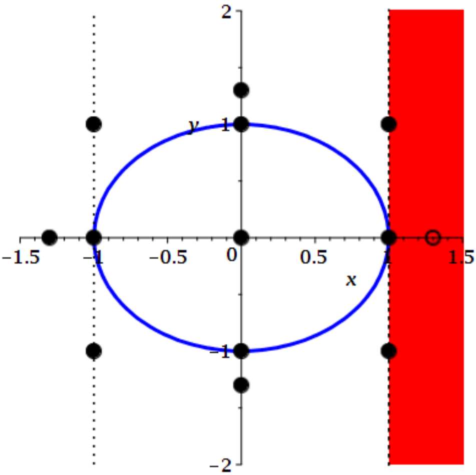
Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$,
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$
Cell 13: $x > 1$, y free



Example: Circle – visualisation

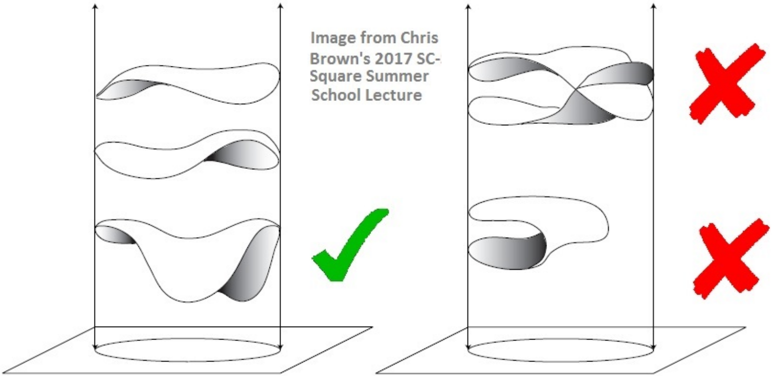
Cell 1: $x < -1$, y free
Cell 2: $x = -1, y < 0$
Cell 3: $x = -1, y = 0$
Cell 4: $x = -1, y > 0$
Cell 5: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y < 0$
Cell 6: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y < 0$
Cell 7: $-1 < x < 1$,
$y^2 + x^2 - 1 < 0$
Cell 8: $-1 < x < 1$,
$y^2 + x^2 - 1 = 0, y > 0$
Cell 9: $-1 < x < 1$,
$y^2 + x^2 - 1 > 0, y > 0$
Cell 10: $x = 1, y < 0$
Cell 11: $x = 1, y = 0$
Cell 12: $x = 1, y > 0$

Cell 13: $x > 1$, y free



How to build a CAD?

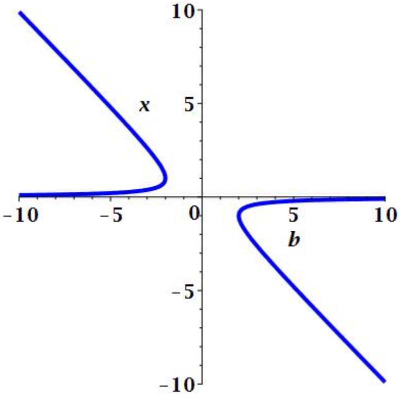
The usual approach is to calculate projection polynomials whose roots indicate changes in the behaviour of the input set; decompose with respect to these and then lift back working at a sample point. Sound if the projection provides **delineability**.



QE via CAD Example

How to determine with CAD?

$$\exists x, x^2 + bx + 1 \leq 0$$



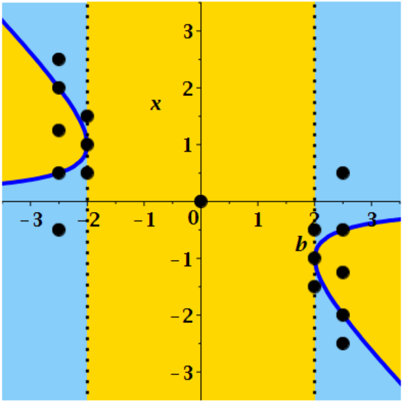
QE via CAD Example

How to determine with CAD?

$$\exists x, x^2 + bx + 1 \leq 0$$

To solve we:

Build a sign-invariant CAD for $f = x^2 + bx + 1$.



QE via CAD Example

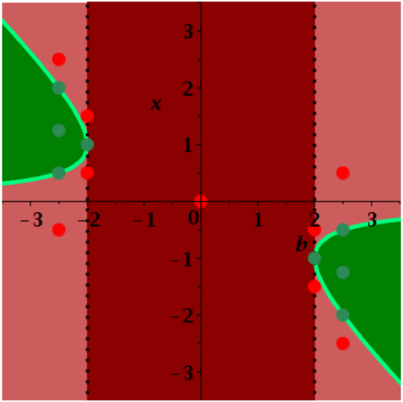
How to determine with CAD?

$$\exists x, x^2 + bx + 1 \leq 0$$

To solve we:

Build a sign-invariant CAD for $f = x^2 + bx + 1$.

Tag each cell true or false according to $f \leq 0$.



QE via CAD Example

How to determine with CAD?

$$\exists x, x^2 + bx + 1 \leq 0$$

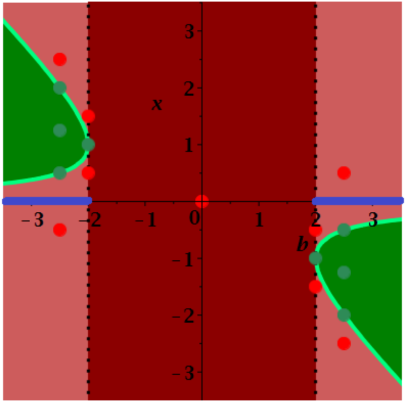
To solve we:

Build a sign-invariant CAD for $f = x^2 + bx + 1$.

Tag each cell true or false according to $f \leq 0$.

Take disjunction of projections of true cells:

$$b < -2 \vee b = -2 \\ \vee b = 2 \vee b > 2$$



QE via CAD Example

How to determine with CAD?

$$\exists x, x^2 + bx + 1 \leq 0$$

To solve we:

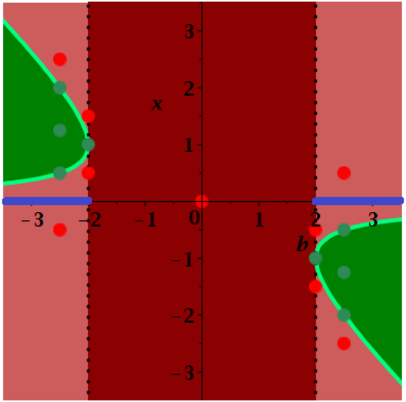
Build a sign-invariant CAD for $f = x^2 + bx + 1$.

Tag each cell true or false according to $f \leq 0$.

Take disjunction of projections of true cells:

$$\Longrightarrow$$

$$b \leq -2 \vee b \geq 2$$



QE via CAD in General

In general we can perform Real QE on a decomposition as follows.

- Eliminate existential quantifiers by projecting the true cells, as in the previous example.
- Eliminate universal quantifiers by using the relation

$$\forall x P(x) = \neg \exists x \neg P(x)$$

and then proceeding with existential QE.

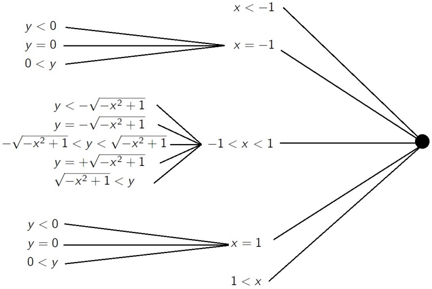
Recall our original example was $\forall x, x^2 + bx + 1 > 0$. The above leads us to study $\exists x, x^2 + bx + 1 \leq 0$ which from the previous slide we know has solution $b \leq -2 \vee b \geq +2$. The solution to our universally quantified problem is then the negation of this: $-2 < b \wedge b < +2$ (as we found numerically earlier).

Cylindricity Property

Decompositions allow us to understand infinite space with a finite number of samples and (semi-) algebraic cells allows us to construct solution formulae easily. But why cylindricity?

The Real QE solution process requires us to project cells (and combine those projections) and to calculate the complement of cells: both of these things are trivial for cells arranged cylindrically.

Cylindricity means we can think of CAD as a tree branching by variable.



Warning: CAD Complexity

By the end of projection you have doubly exponentially many polynomials of doubly exponential degree (in the number of projections, i.e. variables). Hence also the number of real roots, cells and time to compute them grows doubly exponentially!

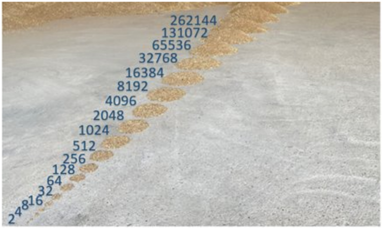
C. Brown and J.H. Davenport.
The complexity of quantifier elimination and cylindrical algebraic decomposition.
In Proc. ISSAC '07, pages 54–60. ACM, 2007.

Thus in practice applications are only realistic for 4 variables, unless specific optimisations are utilised:

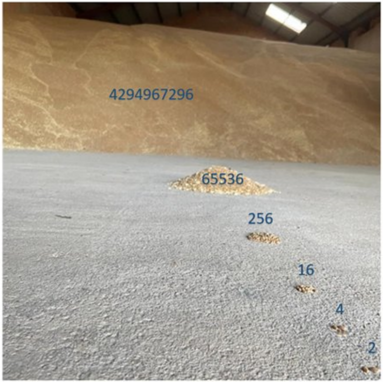
$$2^{2^3} = 256 \qquad 2^{2^4} = 65,536 \qquad 2^{2^5} = 4,294,967,296$$

The Doubly Exponential Wall

Images by Tereso del Rio



Exponential Growth



Doubly Exponential Growth

Real QE Applications

QE can solve problems throughout engineering & science. E.g.

- derivation of optimal numerical schemes.
- artificial intelligence to pass university entrance exam.
- automated theorem proving.
- automated loop parellisation.
- structural design (minimising the weight of trusses).
- ⋮

There is no lack of potential applications: the problem is scaling up in the face of high complexity!

QE Applications References

M. Erascu and H. Hong.
Real quantifier elimination for the synthesis of optimal numerical algorithms (Case study: Square root computation).
Journal of Symbolic Computation, 75:110–126, 2016.

N.H. Arai, T. Matsuzaki, H. Iwane, and H. Anai.
Mathematics by machine.
In Proc. ISSAC '14, pages 1–8. ACM, 2014.

L.C. Paulson.
Metitarski: Past and future.
In Proc ITP '12, pages 1–10. Springer, 2012.

A. Grosslinger, M. Griebl, and C. Lengauer.
Quantifier elimination in automatic loop parallelization.
Journal of Symbolic Computation, 41(11):1206–1221, 2006.

A.E. Charalampakis and I. Chatzigiannelis.
Analytical solutions for the minimum weight design of trusses by cylindrical algebraic decomposition.
Archive of Applied Mechanics, 88(1):39–49, 2018.

QE via CAD Implementations

Both of the big proprietary Computer Algebra Systems have QE implementations: MATHEMATICA (the `Resolve` command) and MAPLE (`RegularChains:-SemiAlgebraicSetTools`; `QuantifierElimination:-CylindricalAlgebraicDecompose`; and `RootFinding:-Parametric:-CellDecomposition`).

The specialist computer algebra system QEPCAD-B is dedicated to QE via CAD and is available for free. It can be used via an intelligent interface TARSKI, is available as a sub-package of SAGE, and can now even run in your browser!

Z. Kovács, C.W. Brown, T. Recio, and R. Vajda
A web version of Tarski, a system for computing with Tarski formulas and semialgebraic sets.
In Proc. SYNASC 2022, pages 59–62, IEEE, 2022.
https://doi.org/10.1109/SYNASC57785.2022.00019

Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
   - Alternatives to Traditional CAD
2. Optimising CAD via Machine Learning
   - ML for Computer Algebra
   - ML for CAD Variable Ordering
   - Beyond Efficiency Gains
3. Applications (time permitting)
   - Biology
   - Economics

Alternatives to CAD for QE

There exists algorithms to achieve Real QE other than via CAD:

- Virtual Term Substitution: more efficient but degree limitations. Implementations in MATHEMATICA and REDLOG.
- QE via Comprehensive Gröbner Bases: more efficient, especially if there are many equations. Implementation in SYNRAC package for MAPLE.
- Family of algorithms for Real QE with complexity doubly exponential in the number of quantifier eliminations. No implementations and analysis suggests that the crossover point where the asymptomatic growth becomes relevant is to far away for practical use at the moment.
- Variety of specialist Real QE algorithms for input of certain shape.

Alternatives to CAD for QE

There exists algorithms to achieve Real QE other than via CAD:

- Virtual Term Substitution: more efficient but degree limitations. Implementations in MATHEMATICA and REDLOG.
- QE via Comprehensive Gröbner Bases: more efficient, especially if there are many equations. Implementation in SYNRAC package for MAPLE.
- Family of algorithms for Real QE with complexity doubly exponential in the number of quantifier eliminations. No implementations and analysis suggests that the crossover point where the asymptomatic growth becomes relevant is to far away for practical use at the moment.
- Variety of specialist Real QE algorithms for input of certain shape.

Alternatives to CAD for QE References

T. Sturm.
Thirty years of virtual substitution: Foundations, techniques, applications.
In Proc. ISSAC 2018, pages 11–16, ACM, 2018.

R. Fukasaku, H. Iwane, and Y. Sato.
Real quantifier elimination by computation of comprehensive Gröbner systems.
In Proc. ISSAC '15, pages 173–180. ACM, 2015.

J. Renegar.
Recent progress on the complexity of the decision problem for the reals.
In B.F. Caviness and J.R. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, pages 220–241. Springer-Verlag, 1998.

H.P. Le and M. Safey El Din.
Faster one block quantifier elimination for regular polynomial systems of equations.
In Proc. ISSAC '21, pages 265–272. ACM, 2021.

Alternatives to Traditional CAD Computation I (Slide 17/48)

There are now various alternative computational schemes for CAD:

1. **CAD adapted for SMT:** CAD as a theory solver in the CDCL(T) paradigm for problems where all variables are existentially quantified.
   The logic space is searched efficiently via a SAT-solver with CAD used to analyse whether subsets of constraints can be satisfied together.
   Open source implementations exist in the SMT-solvers, SMT-RAT, Yices2, Z3 and cvc5.

G. Kremer and E. Ábrahám.
Fully incremental cylindrical algebraic decomposition.
J. of Symbolic Computation, 100, pages 11–37. Elsevier, 2020.
https://doi.org/10.1016/j.jsc.2019.07.018

Alternatives to Traditional CAD Computation II (Slide 18/48)

2. **CAD theory redesigned for SMT:** the theorems that underpin CAD reused to validate search-based algorithms that generalise bad guesses to CAD-cells.

D. Jovanovic and L. de Moura
Solving Non-linear Arithmetic
Proc. IJCAR 2012, LNCS 7364, pp. 339–354.
https://doi.org/10.1007/978-3-642-31365-3_27

E. Ábrahám, J.H. Davenport, M. England and G. Kremer.
Deciding the consistency of non-linear real arithmetic constraints with a conflict driven search using cylindrical algebraic coverings.
JLAMP 119, pages 2352-2208. Elsevier, 2021.
https://doi.org/10.1016/j.jlamp.2020.100633

Alternatives to Traditional CAD Computation III (Slide 19/48)

3. **SMT inspired CAD algorithms for QE:** Generalisations of the new SMT-based algorithms to tackle the full Real QE problem.

C. W. Brown
Open Non-uniform Cylindrical Algebraic Decompositions
Proc. ISSAC 2015, pp. 85–92. ACM, 2015
https://doi.org/10.1145/2755996.2756654

G. Kremer and J. Nalbach.
Cylindrical Algebraic Coverings for Quantifiers.
Proc. SC^2 2022, CEUR-WS 3458, pp. 1–9, 2023.
https://ceur-ws.org/Vol-3458

Alternatives to Traditional CAD Computation IV (Slide 20/48)

4. **CAD via Regular Chains:** First performs a decomposition in $\mathbb{C}^n$ using regular chains (triangular sets) theory and then refines this to a decomposition of $\mathbb{R}^n$.

C. Chen and M. Moreno Maza and B. Xia and L. Yang.
Computing cylindrical algebraic decomposition via triangular decomposition.
Proc. ISSAC 2009, pp. 95–102. ACM, 2009.
https://doi.org/10.1145/1576702.1576718

Alternatives to Traditional CAD Computation V (Slide 21/48)

5. **Geometric CAD:** A new view of CAD proposed made using Geometric Fiber Classification that suggests a computation scheme using Gröbner Bases in place of iterated resultants.

Rizeng Chen.
Geometric Fiber Classification of Morphisms and a Geometric Approach to Cylindrical Algebraic Decomposition.
Preprint, arXiv:2311.10515 (Dec 2023).
https://doi.org/10.48550/arXiv.2311.10515

Optimising CAD

All of the alternative CAD computation schemes above still have doubly exponential complexity in the worst case.
They also all have a sensitivity to the variable ordering which:

- Determines the meaning of cylindricity in the definition;
- Determines the order of operations in the algorithm.

It is well observed that this choice can dramatically effect the efficiency or even tractability of CAD, but there does not exist any theoretical method to make the choice – it is commonly decided by heuristics. This is one key area in which a CAD implementation may be optimised.

We will consider whether this can be achieved via machine learning.

Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
   - Alternatives to Traditional CAD
2. Optimising CAD via Machine Learning
   - ML for Computer Algebra
   - ML for CAD Variable Ordering
   - Beyond Efficiency Gains
3. Applications (time permitting)
   - Biology
   - Economics

Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
   - Alternatives to Traditional CAD
2. Optimising CAD via Machine Learning
   - ML for Computer Algebra
   - ML for CAD Variable Ordering
   - Beyond Efficiency Gains
3. Applications (time permitting)
   - Biology
   - Economics

Machine Learning Mathematics

Machine Learning (ML) uses statistics upon data to learn how to perform tasks that have not been explicitly programmed. ML technology is behind recent Artificial Intelligence (AI) applications. It is increasingly used in industry and academia.

(Q) How can ML assist mathematicians?

There have been a variety of papers suggesting that ML can guide pure mathematicians to potential conjectures.

Davies, Alex and Veličković, Petar and Buesing, Lars and Blackwell, Sam and Zheng, Daniel and Tomašev, Nenad and Tanburn, Richard and Battaglia, Peter and Blundell, Charles and Juhász, András and Lackenby, Marc and Williamson, Geordie and Hassabis, Demis and Kohli, Pushmeet
Advancing mathematics by guiding human intuition with AI.
Nature, 600, pp. 70——74.
https://doi.org/10.1038/s41586-021-04086-x

Machine Learning Mathematics

Machine Learning (ML) uses statistics upon data to learn how to perform tasks that have not been explicitly programmed. ML technology is behind recent Artificial Intelligence (AI) applications. It is increasingly used in industry and academia.

(Q) How can ML assist mathematicians?

There have been a variety of papers suggesting that ML can guide pure mathematicians to potential conjectures.

Davies, Alex and Veličković, Petar and Buesing, Lars and Blackwell, Sam and Zheng, Daniel and Tomašev, Nenad and Tanburn, Richard and Battaglia, Peter and Blundell, Charles and Juhász, András and Lackenby, Marc and Williamson, Geordie and Hassabis, Demis and Kohli, Pushmeet
Advancing mathematics by guiding human intuition with AI.
Nature, 600, pp. 70––74.
https://doi.org/10.1038/s41586-021-04086-x

Machine Learning Matrix Multiplication

Reinforcement learning found an improvement on Strassen's algorithm to multiply matrices:

A. Fawzi and M. Balog and A. Huang and T. Hubert and B. Romera-Paredes and M. Barekatain and A. Novikov and F.J.R. Ruiz and J. Schrittwieser and G. Swirszcz and D. Silver and D. Hassabis and P. Kohli.
Discovering faster matrix multiplication algorithms with reinforcement learning.
Nature, 610, pp. 47——53, 2022.
https://doi.org/10.1038/s41586-022-05172-4

That solution seeded a new random walk based algorithm which found an even better one!

Manuel Kauers and Jakob Moosbauer.
Flip Graphs for Matrix Multiplication.
Proc. ISSAC 2023, pp. 381 – –388. ACM, 2023.
https://doi.org/10.1145/3597066.3597120

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & a_{4,5} \end{pmatrix} \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} & b_{1,5} \\ b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} & b_{2,5} \\ b_{3,1} & b_{3,2} & b_{3,3} & b_{3,4} & b_{3,5} \\ b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} & b_{4,5} \\ b_{5,1} & b_{5,2} & b_{5,3} & b_{5,4} & b_{5,5} \end{pmatrix} = \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} & c_{1,5} \\ c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} & c_{2,5} \\ c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} & c_{3,5} \\ c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} & c_{4,5} \end{pmatrix}$$

$h_1 = a_{3,2}(-b_{2,1} - b_{2,5} - b_{3,1})$
$h_2 = (a_{2,2} + a_{2,5} - a_{3,5})(-b_{2,5} - b_{5,1})$
$h_3 = (-a_{3,1} - a_{4,1} + a_{4,2})(-b_{1,1} + b_{2,5})$
$h_4 = (a_{1,2} + a_{1,4} + a_{3,4})(-b_{2,5} - b_{4,1})$
$h_5 = (a_{1,5} + a_{2,2} + a_{2,5})(-b_{2,4} + b_{5,1})$
$h_6 = (-a_{2,2} - a_{2,5} - a_{4,5})(b_{2,3} + b_{5,1})$
$h_7 = (-a_{1,1} + a_{4,1} - a_{4,2})(b_{1,1} + b_{2,4})$
$h_8 = (a_{3,2} - a_{3,3} - a_{4,3})(-b_{2,3} + b_{3,1})$
$h_9 = (-a_{1,2} - a_{1,4} + a_{4,4})(b_{2,3} + b_{4,1})$
$h_{10} = (a_{2,2} + a_{2,5})b_{5,1}$
$h_{11} = (-a_{2,1} - a_{4,1} + a_{4,2})(-b_{1,1} + b_{2,2})$
$h_{12} = (a_{4,1} - a_{4,2})b_{1,1}$
$h_{13} = (a_{1,2} + a_{1,4} + a_{2,4})(b_{2,2} + b_{4,1})$
$h_{14} = (a_{1,3} - a_{3,2} + a_{3,3})(b_{2,4} + b_{3,1})$
$h_{15} = (-a_{1,2} - a_{1,4})b_{4,1}$
$h_{16} = (-a_{3,2} + a_{3,3})b_{3,1}$
$h_{17} = (a_{1,2} + a_{1,4} - a_{2,1} + a_{2,2} - a_{2,3} + a_{2,4} - a_{3,2} + a_{3,3} - a_{4,1} + a_{4,2})b_{2,2}$
$h_{18} = a_{2,1}(b_{1,1} + b_{1,2} + b_{5,2})$
$h_{19} = -a_{2,3}(b_{3,1} + b_{3,2} + b_{5,2})$
$h_{20} = (-a_{1,5} + a_{2,1} + a_{2,3} - a_{2,5})(-b_{1,1} - b_{1,2} + b_{1,4} - b_{5,2})$
$h_{21} = (a_{2,1} + a_{2,3} - a_{2,5})b_{5,2}$
$h_{22} = (a_{1,3} - a_{1,4} - a_{2,4})(b_{1,1} + b_{1,2} - b_{1,4} - b_{3,1} - b_{3,2} + b_{3,4} + b_{4,4})$
$h_{23} = a_{1,3}(-b_{3,1} + b_{3,4} + b_{4,4})$
$h_{24} = a_{1,5}(-b_{4,4} - b_{5,1} + b_{5,4})$
$h_{25} = -a_{1,1}(b_{1,1} - b_{1,4})$
$h_{26} = (-a_{1,3} + a_{1,4} + a_{1,5})b_{4,4}$
$h_{27} = (a_{1,3} - a_{3,1} + a_{3,3})(b_{1,1} - b_{1,4} + b_{1,5} + b_{3,5})$
$h_{28} = -a_{3,4}(-b_{3,5} - b_{4,1} - b_{4,5})$
$h_{29} = a_{3,1}(b_{1,1} + b_{1,5} + b_{3,5})$
$h_{30} = (a_{3,1} - a_{3,3} + a_{3,4})b_{3,5}$
$h_{31} = (-a_{1,4} - a_{1,5} - a_{3,4})(-b_{4,4} - b_{5,1} + b_{5,4} - b_{5,5})$
$h_{32} = (a_{2,1} + a_{4,1} + a_{4,4})(b_{1,3} - b_{4,1} - b_{4,2} - b_{4,3})$
$h_{33} = a_{4,3}(-b_{3,1} - b_{3,3})$
$h_{34} = a_{4,4}(-b_{1,3} + b_{4,1} + b_{4,3})$
$h_{35} = -a_{4,5}(b_{1,3} + b_{5,1} + b_{5,3})$
$h_{36} = (a_{2,3} - a_{2,5} - a_{4,5})(b_{3,1} + b_{3,2} + b_{3,3} + b_{5,2})$
$h_{37} = (-a_{4,1} - a_{4,4} + a_{4,5})b_{1,3}$
$h_{38} = (-a_{2,3} - a_{3,1} + a_{3,3} - a_{3,4})(b_{3,5} + b_{4,1} + b_{4,2} + b_{4,5})$
$h_{39} = (-a_{3,1} - a_{4,1} - a_{4,4} + a_{4,5})(b_{1,3} + b_{5,1} + b_{5,3} + b_{5,5})$
$h_{40} = (-a_{1,3} + a_{1,4} + a_{1,5} - a_{4,4})(-b_{3,1} - b_{3,3} + b_{3,4} + b_{4,4})$
$h_{41} = (-a_{1,1} + a_{4,1} - a_{4,5})(b_{1,3} + b_{3,1} + b_{3,3} - b_{3,4} + b_{5,1} + b_{5,3} - b_{5,4})$
$h_{42} = (-a_{2,1} + a_{2,5} - a_{3,5})(-b_{1,1} - b_{1,2} - b_{1,5} + b_{4,1} + b_{4,2} + b_{4,5} - b_{5,2})$
$h_{43} = a_{2,4}(b_{4,1} + b_{4,2})$
$h_{44} = (a_{2,3} + a_{3,2} - a_{3,3})(b_{2,2} - b_{3,1})$
$h_{45} = (-a_{3,3} + a_{3,4} - a_{4,3})(b_{3,5} + b_{4,1} + b_{4,3} + b_{4,5} + b_{5,1} + b_{5,3} + b_{5,5})$
$h_{46} = -a_{3,5}(-b_{5,1} - b_{5,5})$
$h_{47} = (a_{2,1} - a_{2,5} - a_{3,1} + a_{3,5})(b_{1,1} + b_{1,2} + b_{1,5} - b_{4,1} - b_{4,2} - b_{4,5})$
$h_{48} = (-a_{2,3} + a_{3,3})(b_{2,2} + b_{3,2} + b_{3,5} + b_{4,1} + b_{4,2} + b_{4,5})$
$h_{49} = (-a_{1,1} - a_{1,3} + a_{1,4} + a_{1,5} - a_{2,1} - a_{2,3} + a_{2,4} + a_{2,5})(-b_{1,1} - b_{1,2} + b_{1,4})$
$h_{50} = (-a_{1,4} - a_{2,4})(b_{2,2} - b_{3,1} - b_{3,2} + b_{3,4} - b_{4,2} + b_{4,4})$

$h_{51} = a_{2,2}(b_{2,1} + b_{2,2} - b_{5,1})$
$h_{52} = a_{4,2}(b_{1,1} + b_{2,1} + b_{2,3})$
$h_{53} = -a_{1,2}(-b_{2,1} + b_{2,4} + b_{4,1})$
$h_{54} = (a_{1,2} + a_{1,4} - a_{2,2} - a_{2,5} - a_{3,2} + a_{3,3} - a_{4,2} + a_{4,3} - a_{4,4} - a_{4,5})b_{2,3}$
$h_{55} = (a_{1,4} - a_{4,4})(-b_{2,3} + b_{3,1} + b_{3,3} - b_{3,4} + b_{4,3} - b_{4,4})$
$h_{56} = (a_{1,1} - a_{1,5} - a_{4,1} + a_{4,5})(b_{3,1} + b_{3,3} - b_{3,4} + b_{5,1} + b_{5,3} - b_{5,4})$
$h_{57} = (-a_{3,1} - a_{4,1})(-b_{1,3} - b_{1,5} - b_{2,5} - b_{5,1} - b_{5,3} - b_{5,5})$
$h_{58} = (-a_{1,4} - a_{1,5} - a_{3,4} - a_{3,5})(-b_{5,1} + b_{5,4} - b_{5,5})$
$h_{59} = (-a_{3,3} + a_{3,4} - a_{4,3} + a_{4,4})(b_{4,1} + b_{4,3} + b_{4,5} + b_{5,1} + b_{5,3} + b_{5,5})$
$h_{60} = (a_{2,5} + a_{4,5})(b_{2,3} - b_{3,1} - b_{3,2} - b_{3,3} - b_{5,2} - b_{5,3})$
$h_{61} = (a_{1,4} + a_{3,4})(b_{1,1} - b_{1,4} + b_{1,5} - b_{2,5} - b_{4,4} + b_{4,5} - b_{5,1} + b_{5,4} - b_{5,5})$
$h_{62} = (a_{2,1} + a_{4,1})(b_{1,2} + b_{1,3} + b_{2,2} - b_{4,1} - b_{4,2} - b_{4,3})$
$h_{63} = (-a_{3,3} - a_{4,3})(-b_{2,3} - b_{3,3} - b_{3,5} - b_{4,1} - b_{4,3} - b_{4,5})$
$h_{64} = (a_{1,1} - a_{1,3} - a_{1,4} + a_{3,1} - a_{3,3} - a_{3,4})(b_{1,1} - b_{1,4} + b_{1,5})$
$h_{65} = (-a_{1,1} + a_{4,1})(-b_{1,3} + b_{1,4} + b_{2,4} - b_{5,1} - b_{5,3} + b_{5,4})$
$h_{66} = (a_{1,1} - a_{1,2} + a_{1,3} - a_{1,5} - a_{2,2} - a_{2,5} - a_{3,2} + a_{3,3} - a_{4,1} + a_{4,2})b_{2,4}$
$h_{67} = (a_{2,5} - a_{3,5})(b_{1,1} + b_{1,2} + b_{1,5} - b_{2,5} - b_{4,1} - b_{4,2} - b_{4,5} + b_{5,2} + b_{5,5})$
$h_{68} = (a_{1,1} + a_{1,3} - a_{1,4} - a_{1,5} - a_{4,1} - a_{4,3} + a_{4,4} + a_{4,5})(-b_{3,1} - b_{3,3} + b_{3,4})$
$h_{69} = (-a_{1,3} + a_{1,4} - a_{2,3} + a_{2,4})(-b_{2,4} - b_{3,1} - b_{3,2} + b_{3,4} - b_{5,2} + b_{5,4})$
$h_{70} = (a_{2,3} - a_{2,5} + a_{4,3} - a_{4,5})(-b_{3,1} - b_{3,2} - b_{3,3})$
$h_{71} = (-a_{3,1} + a_{3,3} - a_{3,4} + a_{3,5} - a_{4,1} + a_{4,3} - a_{4,4} + a_{4,5})(-b_{5,1} - b_{5,3} - b_{5,5})$
$h_{72} = (-a_{2,1} - a_{2,4} - a_{4,1} - a_{4,4})(b_{4,1} + b_{4,2} + b_{4,3})$
$h_{73} = (a_{1,3} - a_{1,4} - a_{1,5} + a_{2,3} - a_{2,4} - a_{2,5})(b_{1,1} + b_{1,2} - b_{1,4} + b_{2,4} + b_{5,2} - b_{5,4})$
$h_{74} = (a_{2,1} - a_{2,3} + a_{2,4} - a_{3,1} + a_{3,3} - a_{3,4})(b_{4,1} + b_{4,2} + b_{4,5})$
$h_{75} = -(a_{1,2} + a_{1,4} - a_{2,2} - a_{2,5} - a_{3,1} + a_{3,2} + a_{3,4} + a_{3,5} - a_{4,1} + a_{4,2})b_{2,5}$
$h_{76} = (a_{1,3} + a_{3,3})(-b_{1,1} + b_{1,4} - b_{1,5} + b_{2,4} + b_{3,4} - b_{3,5})$
$c_{1,1} = -h_{10} + h_{12} + h_{14} - h_{15} - h_{16} + h_{53} + h_5 - h_{66} - h_7$
$c_{2,1} = h_{10} + h_{11} - h_{12} + h_{13} + h_{15} + h_{16} - h_{17} - h_{44} + h_{51}$
$c_{3,1} = h_{10} - h_{12} + h_{15} + h_{16} - h_1 + h_2 + h_3 - h_4 + h_{75}$
$c_{4,1} = -h_{10} + h_{12} - h_{15} - h_{16} + h_{52} + h_{54} - h_6 - h_8 + h_9$
$c_{1,2} = h_{13} + h_{15} + h_{20} + h_{21} - h_{22} + h_{23} + h_{25} - h_{43} + h_{49} + h_{50}$
$c_{2,2} = -h_{11} + h_{12} - h_{13} - h_{15} - h_{16} + h_{17} + h_{18} - h_{19} - h_{21} + h_{43} + h_{44}$
$c_{3,2} = -h_{16} - h_{19} - h_{21} - h_{28} - h_{29} - h_{38} + h_{42} + h_{44} - h_{47} + h_{48}$
$c_{4,2} = h_{11} - h_{12} - h_{18} + h_{21} - h_{32} + h_{33} - h_{34} - h_{36} + h_{62} - h_{70}$
$c_{1,3} = h_{15} + h_{23} + h_{24} + h_{34} - h_{37} + h_{40} - h_{41} + h_{55} - h_{56} - h_9$
$c_{2,3} = -h_{10} + h_{19} + h_{32} + h_{35} + h_{36} + h_{37} - h_{43} - h_{60} - h_6 - h_{72}$
$c_{3,3} = -h_{16} - h_{28} + h_{33} + h_{37} - h_{39} + h_{45} - h_{46} + h_{63} - h_{71} - h_8$
$c_{4,3} = h_{10} + h_{15} + h_{16} - h_{33} + h_{34} - h_{35} - h_{37} - h_{54} + h_6 + h_8 - h_9$
$c_{1,4} = -h_{10} + h_{12} + h_{14} - h_{16} + h_{23} + h_{24} + h_{25} + h_{26} + h_5 - h_{66} - h_7$
$c_{2,4} = h_{10} + h_{18} - h_{19} + h_{20} - h_{22} - h_{24} - h_{26} - h_5 - h_{69} + h_{73}$
$c_{3,4} = -h_{14} + h_{16} - h_{23} - h_{26} + h_{27} + h_{29} + h_{31} + h_{46} - h_{58} + h_{76}$
$c_{4,4} = h_{12} + h_{25} + h_{26} - h_{33} - h_{35} - h_{40} + h_{41} + h_{65} - h_{68} - h_7$
$c_{1,5} = h_{15} + h_{24} + h_{25} + h_{27} - h_{28} + h_{30} + h_{31} - h_4 + h_{61} + h_{64}$
$c_{2,5} = -h_{10} - h_{18} - h_2 - h_{30} - h_{38} + h_{42} - h_{43} + h_{46} + h_{67} + h_{74}$
$c_{3,5} = -h_{10} + h_{12} - h_{15} + h_{28} + h_{29} - h_2 - h_{30} - h_3 + h_{46} + h_4 - h_{75}$
$c_{4,5} = -h_{12} - h_{29} + h_{30} - h_{34} + h_{35} + h_{39} + h_3 - h_{45} + h_{57} + h_{59}$

Extended Data Fig. 2 | Algorithm for multiplying 4 × 5 by 5 × 5 matrices in standard arithmetic with 76 multiplications. This outperforms the previously best known algorithm, which involves 80 multiplications.

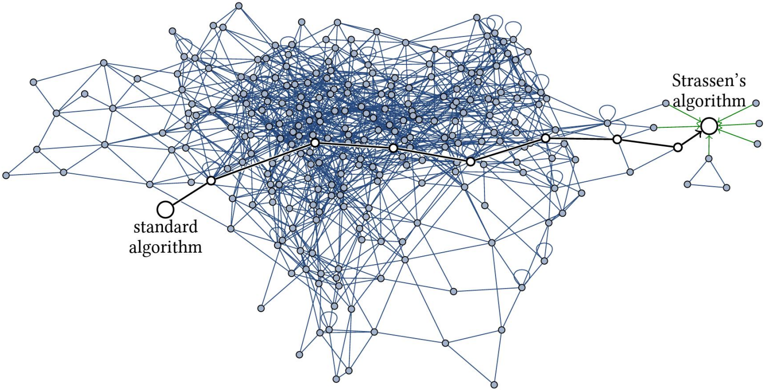
Machine Learning Matrix Multiplication

Reinforcement learning found an improvement on Strassen's algorithm to multiply matrices:

A. Fawzi and M. Balog and A. Huang and T. Hubert and B. Romera-Paredes and M. Barekatain and A. Novikov and F.J.R. Ruiz and J. Schrittwieser and G. Swirszcz and D. Silver and D. Hassabis and P. Kohli.
Discovering faster matrix multiplication algorithms with reinforcement learning.
Nature, 610, pp. 47——53, 2022.
https://doi.org/10.1038/s41586-022-05172-4

That solution seeded a new random walk based algorithm which found an even better one!

Manuel Kauers and Jakob Moosbauer.
Flip Graphs for Matrix Multiplication.
Proc. ISSAC 2023, pp. 381 – –388. ACM, 2023.
https://doi.org/10.1145/3597066.3597120

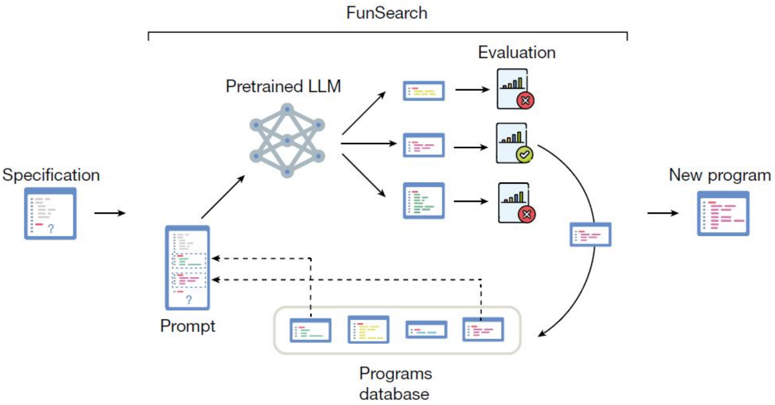

Strassen's algorithm
standard algorithm

Machine Learning to make Algorithms

Romera-Paredes, B. and Barekatain, M. and Novikov, A. and Balog, M. and Kumar, M.P. and Dupont, E. and Ruiz, F.J.R. and Ellenberg, J.S. and Wang, P. and Fawzi, O. and Kohli, P. and Fawzi, A.
Mathematical discoveries from program search with large language models.
Nature, 625, pp. 468-–475, 2023.
https://doi.org/10.1038/s41586-023-06924-6

Starts with a database of heuristic methods for the problems and evolves them by feeding two at a time as a prompt to an LLM (Codey) which is asked to merge them; with the outputs evaluated for fitness and the best added into the database.

Does not train the LLM themselves! The LLM is simply the merger tool. They produce programs that describe how to solve a problem, rather than the problem solution itself.


FunSearch
Evaluation
Pretrained LLM
Specification
New program
Prompt
Programs
database

Symbolic Computation vs Machine Learning (Slide 26/48)

(Q) So can ML replace symbolic computation?

There is a growing body of research on the use of ML in place of expensive symbolic computation. E.g. for integration and the solution of differential equations.

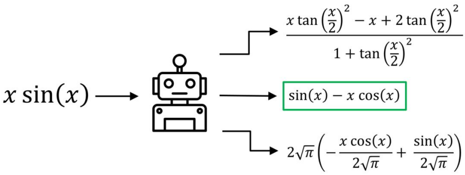
G. Lample and D. Charton
Deep Learning for Symbolic Mathematics.
Proc. ICLR 2020.
https://doi.org/10.48550/arXiv.1912.01412

These tasks are well-suited because it is cheap to symbolically check the correctness of the answer! However, this is not the case for most symbolic computation.

ML for Symbolic Algorithm Selection

The types of integrals dealt with in that paper were very simple. The integrator in a Computer Algebra System like Maple can integrate far more general functions but does so using a wide variety of different algorithms.

(Q) How to select the right algorithm to use for an input?



Maplesoft

Coventry University PhD student Rashid Barket is sponsored by Maplesoft to build an ML-based algorithm selector.

Symbolic Computation WITH Machine Learning (Slide 28/48)

ML can only offer probabilistic guidance, but symbolic computation prizes exact results. 99% accuracy is great for image recognition but would not be acceptable for a mathematical proof.

However, ML can be applied to symbolic computation and still ensure exact results; by having it select or otherwise guide existing algorithms rather than replace them entirely.

Symbolic Computation algorithms often come with choices that need to be made, which do not effect the mathematical correctness of the final result, but which do effect the resources required to find that result (and often how the result is presented).

Such choices are often either left to the user, hard coded by the developer, or made based on a simple heuristic. ML may be able to offer a superior choice.

This is INTERESTING Machine Learning

Summary: ML should be of interest to those developing mathematical algorithms.

But also, ML researchers should also be interested in this application. It is a particularly challenging (interesting) domain:

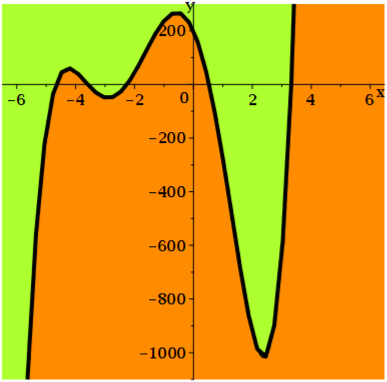
- No a priori limit on the input space.
- Supervised learning hard: because labelling dataset needs lots of expensive symbolic computation.
- Unsupervised learning is hard: because it is unclear if a particular outcome is good or bad without seeing the rest.
- What constitutes a meaningful and representative data set?
- Insufficient quantities of real world data for deep learning.
- How to perform data augmentation / synthetic data generation to allow for generalisability on problems of interest?

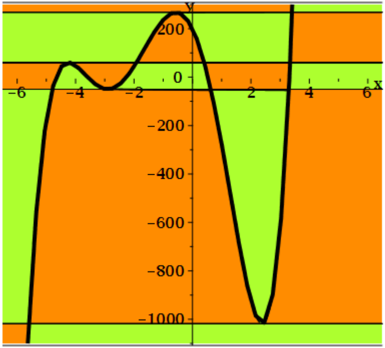
Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
   - Alternatives to Traditional CAD
2. Optimising CAD via Machine Learning
   - ML for Computer Algebra
   - ML for CAD Variable Ordering
   - Beyond Efficiency Gains
3. Applications (time permitting)
   - Biology
   - Economics

Variable Ordering for CAD

CAD requires a variable ordering: there can be multiple valid orderings which lead to an acceptable decomposition, but some lead to smaller decompositions via less computation.





CAD Variable Ordering Choice

Depending on our application we may have a free or constrained choice in variable ordering. When using CAD for QE we must project variables in the order of quantification, but we are free to change order within quantifier blocks (and with the free variables).

The variable ordering has been long known to effect the number of cells produced in a CAD or even the tractability of a problem.

There are various *human-designed heuristics* to make the choice:

- Simple heuristics which use only simple measures of polynomials (degrees, sparsity etc.)
- Expensive heuristics which use more involved algebraic computations.

Simple human-designed heuristics

Brown's Heuristic: chooses variable based on overall degree; breaking ties with total degree of the terms with that variable; breaking ties with number of terms the variable is in.

C. Brown
Companion to the Tutorial: Cylindrical Algebraic Decomposition, presented at ISSAC 2004.
www.usna.edu/Users/cs/wcbrown/research/ISSAC04/handout.pdf

gmods: chooses variable based on the sum of its degree in each polynomial.

T. del Río and M. England
New Heuristic to Choose a Cylindrical Algebraic Decomposition Variable Ordering Motivated by Complexity Analysis
Proc. CASC 2022, LNCS 13366, pp. 300––317. Springer, 2022.
https://doi.org/10.1007/978-3-031-14788-3_17

First use of ML for CAD variable ordering choice (Slide 33/48)

First attempt used a support vector machine classifier to choose which of the three variable ordering heuristics we had implemented in our CAD code to follow for a given problem instance: no one heuristic dominated the others; ML choice did better than any one.

Z. Huang, M. England, D. Wilson, J. Davenport, L. Paulson and J. Bridge.
Applying machine learning to the problem of choosing a heuristic to select the variable ordering for cylindrical algebraic decomposition.
Intelligent Computer Mathematics (LNCS 8543), pp. 92-107. Springer Berlin Heidelberg, 2014.
http://dx.doi.org/10.1007/978-3-319-08434-3_8

The first use of Machine Learning to optimise computer algebra.

EPSRC ML4QE Project

Experiments with ML methodology to choose the ordering directly. In particular:

- Classifier model choice.
- New feature extraction methods to represent polynomials to Machine Learning models: brute force combinations of simple statistics followed by selection
- Tailored hyper-parameter selection to runtime rather than classification accuracy.

D. Florescu and M. England.
A Machine Learning Based Software Pipeline to Pick the Variable Ordering for Algorithms with Polynomial Inputs.
Mathematical Software (LNCS 12097), pp. 302–311. Springer International Publishing, 2020.
https://doi.org/10.1007/978-3-030-52200-1_30

Dataset Considerations

The above work used the `QF_NRA` SMT-LIB benchmarks: a collection of satisfiability problems whose atoms are non-linear polynomial constraints over the reals.

Chosen as the largest (only substantial) existing set of benchmarks for CAD. Problems are meaningful: come from "*industry*" (theorem provers MetiTarski and Keymaera), biology, economics, dynamic geometry proofs, etc. **However:**

- Dataset is highly imbalanced with respect to CAD variable ordering. Risk of over-fitting given no reason to expect imbalance in general data.
- Data is highly dominated by one source of problems.
- Dataset contains many problems that are very similar (e.g. differ by a constant). Thus a risk of data leakage between testing and training.

Dataset Lessons

1. Use variable permutations on problem instances to:
   - Balance dataset (random permutation on each instance).
   - Augment dataset (from each problem instance create a set of instances from all possible permutations).
2. Merge problem instances who have the same CAD tree structure in each variable ordering.

We find the loss of "accuracy" from (1a) is offset from (1b); while (2) allows for similar learning from a dataset a fraction of the size.

Tereso del Río and M. England.
Data Augmentation for Mathematical Objects.
Proc. SC^2 2023, CEUR-WS 3455, pp. 29––38, 2023.
https://ceur-ws.org/Vol-3455/

From Classification to Regression

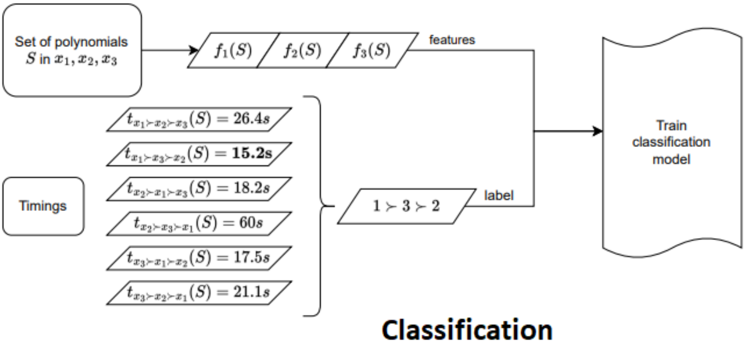
Prior work framed this as ML *Classification* (choose from discrete set of variable orderings). May be reframed as ML *Regression* predict the time taken with a particular ordering: multiple predictions then compared to choose the final ordering. Why?

- A model trained with regression has access to more information: not just which ordering did best but also which came second, which third etc.
- However, regression is a more difficult task (but we only need to be good enough at it to make a ranking).

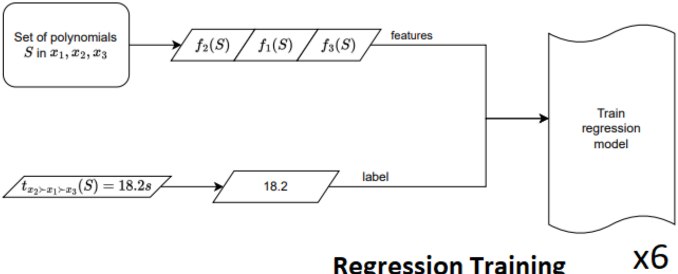
So we hypothesised that regression would outperform classification.

T. del Río and M. England.
Lessons on Datasets and Paradigms in Machine Learning for Symbolic Computation: A Case Study on CAD..
Submitted, 2024. https://doi.org/10.48550/arXiv.2401.13343

Classification vs Regression Flowcharts

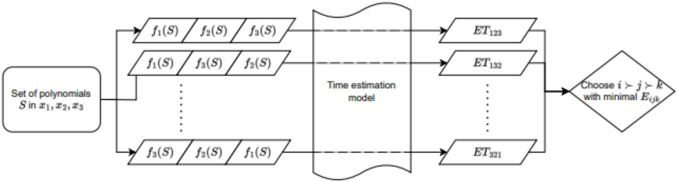


Classification vs. Regression Flowcharts



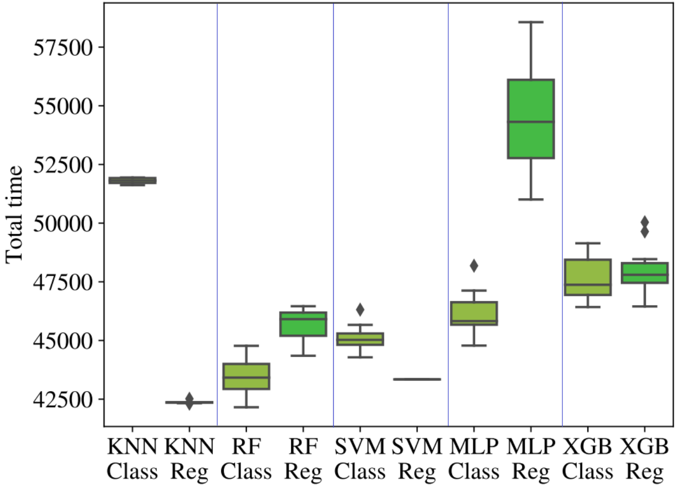
Regression Training

Regression Testing



Classification vs. Regression Results

Regression not universally beneficial, but new state-of-the-art ML.



Remaining Challenges

None of the above used Deep Learning. The dataset size, even after augmentation, did not really support this.

Key remaining challenges:

1. How to create synthetic data sufficient for training deep learning but representative of the problems to be solved in reality?
2. Better representations of the mathematics to the ML models than feature extraction. E.g. Graph Neural Networks.

Some progress on similar challenges in the context of symbolic integration algorithm selection.

Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
   - Alternatives to Traditional CAD
2. Optimising CAD via Machine Learning
   - ML for Computer Algebra
   - ML for CAD Variable Ordering
   - Beyond Efficiency Gains
3. Applications (time permitting)
   - Biology
   - Economics

Inspiration

D. Peifer, M. Stillman, and D. Halpern-Leistner.
Learning selection strategies in Buchberger's algorithm,
In: Proc. ICML 2020, pages 7575-7585, PMLR, 2020.

Applied reinforcement learning to choose the order in which to process S-pairs in Buchberger's algorithm for a Gröbner Basis.

Human analysis of their model revealed a simple, "*human level*" strategy that explained most of the choices. Simple, but very different the human-designed heuristics.

Suggests that ML can reveal new mathematical truths about the algorithm, and direct future non-ML algorithm development.

How to automate such analysis?

Explainable AI (XAI)

The field of **Explainable AI (XAI)** covers ML techniques whose decisions can be explained to a human expert. XAI is increasingly important: both as an error check on the ML process; and to build trust in new AI technologies.

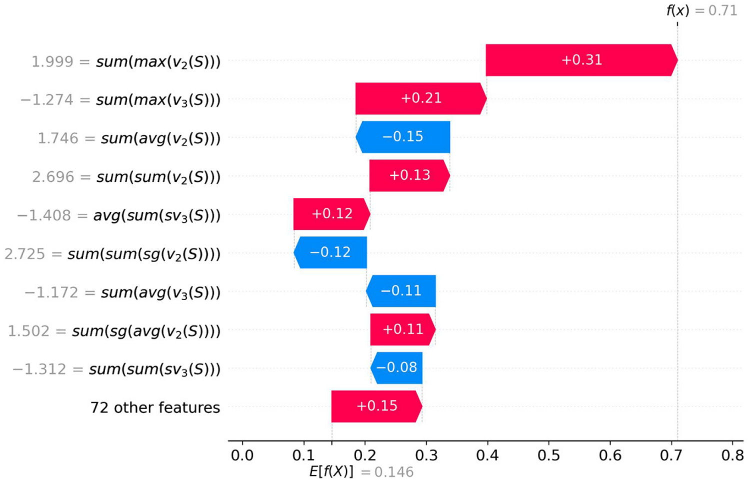
- **Ante-hoc explainable models** are transparent by design in their decisions, e.g. linear regression and support vector machines which fit a line to maximise an objective.
- **Post-hoc explainability** is where an ML model is trained as normal, and then a secondary analysis provides an interpretation.

Post-hoc XAI for CAD Variable Ordering

L. Pickering, T. Del Rio Almajano, M. England and K. Cohen.
Explainable AI Insights for Symbolic Computation: A case study on selecting the variable ordering for cylindrical algebraic decomposition.
Journal of Symbolic Computation, **123**, Article Number 102276, 2024.

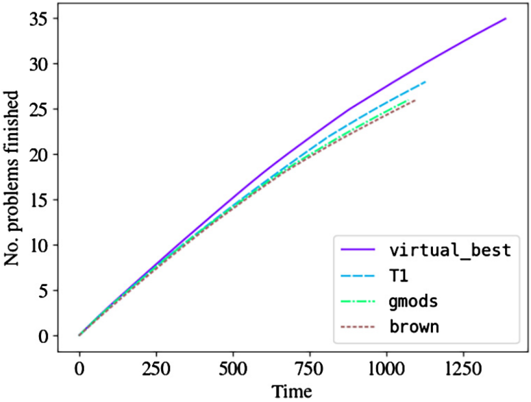
We applied the SHAP XAI tool to analyse the features used by ML classifiers to choose the CAD variable ordering. Some features identified as impactful had been known before; others were new.

Example SHAP Waterfall Plot



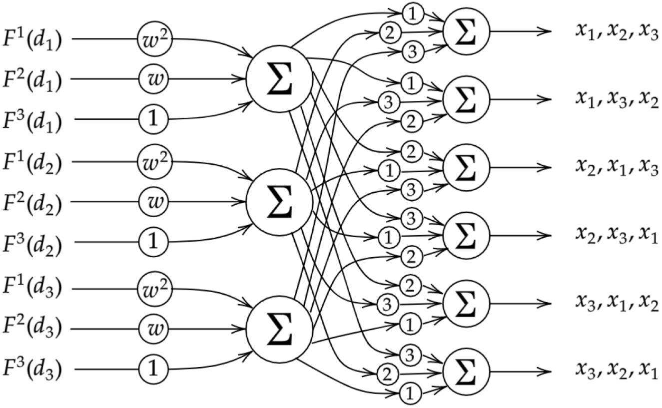
Designing a heuristics from SHAP output

We aggregated across the dataset; had models vote on features to create overall feature ranking; took the top 6 voted features and studied ordered triples of them (similar to Brown's heuristic).



Ante-hoc XAI for CAD Variable Ordering

Started by formulating Brown's heuristic as a neural network:

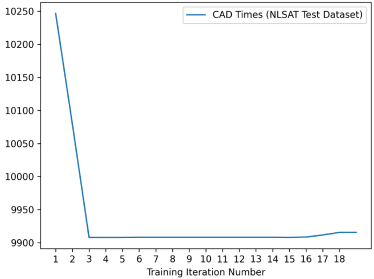


Where w is chosen so that $F^i(d_v) < 2 - 1$ for all features and variables.

Searching the space of similar networks

Starting from the Brown NN we optimise by ML training to:

- use different features F^i (i.e. different feature triples);
- use different weights (more complicated combinations of the same three pieces of information).



What results is a heuristic informed by a ML process but no more complicated than the human designed one.

This one is ongoing work with Dorian Florescu (not yet published).

Summary of the Two XAI Experiments

Both experiments took a *human-designed* heuristic as a starting point and then used ML processes to produce another heuristic.

The latter is not human-designed but is *human-level*:

- can be expressed in natural language in a similar amount of text as a heuristic designed by a human;
- can be implemented without any ML architecture.

Potentially these could be new methodologies for heuristic design.

Thanks for Listening

Contact Details

Matthew.England@coventry.ac.uk

https://matthewengland.coventry.domains/

Outline

1 The Real QE Problem
- Quantifier Elimination
- Real QE via CAD
- Alternatives to Traditional CAD

2 Optimising CAD via Machine Learning
- ML for Computer Algebra
- ML for CAD Variable Ordering
- Beyond Efficiency Gains

3 Applications (time permitting)
- Biology
- Economics

Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
   - Alternatives to Traditional CAD
2. Optimising CAD via Machine Learning
   - ML for Computer Algebra
   - ML for CAD Variable Ordering
   - Beyond Efficiency Gains
3. Applications (time permitting)
   - Biology
   - Economics

Chemical Reaction Networks

A **Chemical Reaction Network** (CRN) models the behaviour of a chemical system. Toy Example:

$$2X_1 + X_2 \xrightarrow{k_1} 3X_1 \xrightarrow{k_2} X_1 + 2X_2 \xrightarrow{k_3} 3X_2 \, , \qquad 3X_2 \xrightarrow{k_4} 2X_1 + X_2 .$$

From this we define a dynamical system:

$$\begin{cases} \dot{x_1} = (3-2)k_1 x_1 x_1 x_2 - 2k_2 x_1^3 - k_3 x_1 x_2^2 + 2k_4 x_2^3, \\ \dot{x_2} = -k_1 x_1^2 x_2 + 2k_2 x_1^3 + k_3 x_1 x_2^2 - 2k_4 x_2^3. \end{cases}$$

Steady states when all the derivatives are zero:

$$\begin{cases} k_1 x_1^2 x_2 - 2k_2 x_1^3 - k_3 x_1 x_2^2 + 2k_4 x_2^3 = 0, \\ x_1 + x_2 - k_5 = 0. \end{cases}$$

Parametric real semi-algebraic geometry problem.

Chemical Reaction Networks

A **Chemical Reaction Network** (CRN) models the behaviour of a chemical system. Toy Example:

$$2X_1 + X_2 \xrightarrow{k_1} 3X_1 \xrightarrow{k_2} X_1 + 2X_2 \xrightarrow{k_3} 3X_2 \, . \qquad 3X_2 \xrightarrow{k_4} 2X_1 + X_2$$

From this we define a dynamical system:

$$\begin{cases} \dot{x_1} = (3-2)k_1 x_1 x_1 x_2 - 2k_2 x_1^3 - k_3 x_1 x_2^2 + 2k_4 x_2^3, \\ \dot{x_2} = -k_1 x_1^2 x_2 + 2k_2 x_1^3 + k_3 x_1 x_2^2 - 2k_4 x_2^3. \end{cases}$$

Steady states when all the derivatives are zero:

$$\begin{cases} k_1 x_1^2 x_2 - 2k_2 x_1^3 - k_3 x_1 x_2^2 + 2k_4 x_2^3 = 0, \\ x_1 + x_2 - k_5 = 0. \end{cases}$$

Parametric real semi-algebraic geometry problem.

Chemical Reaction Networks

A **Chemical Reaction Network** (CRN) models the behaviour of a chemical system. Toy Example:

$$2X_1 + X_2 \xrightarrow{k_1} 3X_1 \xrightarrow{k_2} X_1 + 2X_2 \xrightarrow{k_3} 3X_2 \; , \qquad 3X_2 \xrightarrow{k_4} 2X_1 + X_2 .$$

From this we define a dynamical system:

$$\begin{cases} \dot{x_1} = (3-2)k_1 x_1 x_1 x_2 - 2k_2 x_1^3 - k_3 x_1 x_2^2 + 2k_4 x_2^3, \\ \dot{x_2} = -k_1 x_1^2 x_2 + 2k_2 x_1^3 + k_3 x_1 x_2^2 - 2k_4 x_2^3. \end{cases}$$

Steady states when all the derivatives are zero:

$$\begin{cases} k_1 x_1^2 x_2 - 2k_2 x_1^3 - k_3 x_1 x_2^2 + 2k_4 x_2^3 = 0, \\ x_1 + x_2 - k_5 = 0. \end{cases}$$

Parametric real semi-algebraic geometry problem.

Multistationarity

A CRN exhibits **multistationarity** if there exists a choice of parameter values for which the system has more than one (positive real) solution.

Why care about multistationarity?

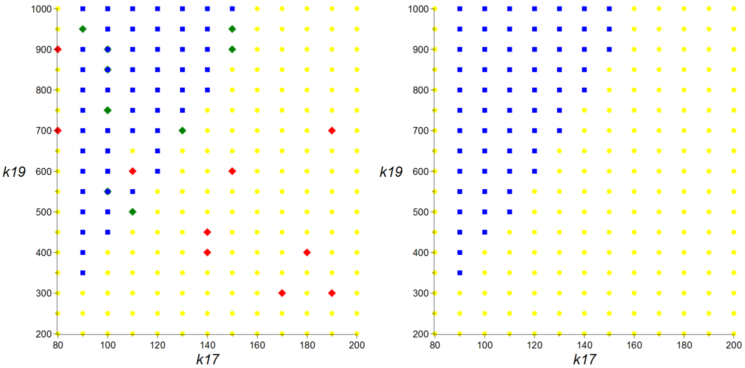
- Used for switch behaviour.
- Instrumental to cellular memory and cell differentiation during development.
- Used by micro organisms in survival strategies.
- Used in decision making processes in the cell division cycle.

There are a variety of efficient methods for answering the Boolean question as to whether or not a system can exhibit multistationarity. Less studied is the question of determining the actual parameter values where multistationarity occurs.

Numeric Methods for Multistationarity 1

Numeric sampling of the parameter space is used. However:

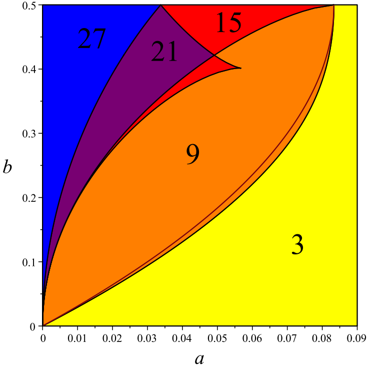
- Incorrect results can be obtained at ill-conditioned points.
- No guarantee all areas of interest will be sampled.



Numeric Methods for Multistationarity 2

Numeric sampling of the parameter space is used. However:

- Incorrect results can be obtained at ill-conditioned points.
- No guarantee all areas of interest will be sampled.



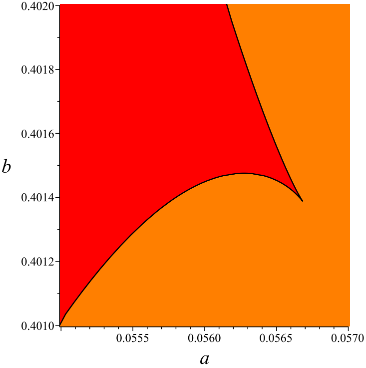


Image courtesy of AmirHosein SadeghiManesh.

Biology Example References

R. Bradford, J.H. Davenport, M. England, H. Errami, V. Gerdt, D. Grigoriev, C. Hoyt, M. Košta, O. Radulescu, T. Sturm, and A. Weber.
Identifying the parametric occurrence of multiple steady states for some biological networks.
Journal of Symbolic Computation, 98:84–119, 2020.

D. Lazard and F. Rouillier.
Solving parametric polynomial systems.
Journal of Symbolic Computation, 42(6):636–667, 2007.

G. Röst and A. Sadeghimanesh.
Exotic bifurcations in three connected populations with Allee effects.
International Journal of Bifurcation and Chaos, 31(13):2150202, 2021.

A. Sadeghimanesh and M. England.
Polynomial Superlevel Set Representation of the Multistationarity Region of Chemical Reaction Networks.
BMC Bioinformatics, 23 Article number 391, 26 pages, 2022.

Outline

1. The Real QE Problem
   - Quantifier Elimination
   - Real QE via CAD
   - Alternatives to Traditional CAD
2. Optimising CAD via Machine Learning
   - ML for Computer Algebra
   - ML for CAD Variable Ordering
   - Beyond Efficiency Gains
3. Applications (time permitting)
   - Biology
   - Economics

Framework for Reasoning in Economics

Determine whether, with variables $\Lambda = (v_1, \ldots, v_n)$, the hypotheses $H(\Lambda)$ follow from the assumptions $A(\Lambda)$. I.e. answer

$$\forall \Lambda \,.\, A(\Lambda) \Rightarrow H(\Lambda)?$$

Logically the answer must be `True` or `False`. But economists are interested also in the following:

- Are the assumptions themselves contradictory?
- If `False`, can additional assumptions be made to give `True`?
- If `True`, can any assumptions be removed?

Such questions can be answered by Real QE in many cases.

TheoryGuru Framework

TheoryGuru package for Mathematica led by Casey Milligan (Chicago). For a proposed economics theorem, we check both:

- the existence of an example
  $\exists \Lambda, A \wedge H$,
- and the existence of a counterexample
  $\exists \Lambda, A \wedge \neg H$.

Then categorize the proposed theorem as:

	$\neg\exists \Lambda, A \wedge \neg H$	$\exists \Lambda, A \wedge \neg H$
$\exists \Lambda, A \wedge H$	`True`	`Mixed`
$\neg\exists \Lambda, A \wedge H$	Contradictory Assumptions	`False`

An economist can explore: e.g. strengthen assumptions of `Mixed` result, or weaken assumptions of `True` result.

Example #0013 Context

Context: Disagreement between Mulligan and Krugman on causes of recession. In particular, latter asserted that whenever taxes on labour supply are primarily responsible for a recession then wages increase. Benchmark Example #0013 considers this claim.

Two scenarios to track: what actually happens (act) when taxes (t) and demand forces (a) together create a recession, and what would have happened (hyp) if taxes on labour supply had been the only factor affecting the market.

The labour demand and supply functions $D(w, a)$ and $S(w, t)$ meet at the labour market equilibrium to supply quantity of labour q.

Example #0013 Context

Context: Disagreement between Mulligan and Krugman on causes of recession. In particular, latter asserted that whenever taxes on labour supply are primarily responsible for a recession then wages increase. Benchmark Example #0013 considers this claim.

Two scenarios to track: what actually happens (act) when taxes (t) and demand forces (a) together create a recession, and what would have happened (hyp) if taxes on labour supply had been the only factor affecting the market.

The labour demand and supply functions $D(w, a)$ and $S(w, t)$ meet at the labour market equilibrium to supply quantity of labour q.

Example #0013 Logic

$$
\begin{aligned}
A \equiv \frac{\partial D(w,a)}{\partial w} < 0 \wedge \frac{\partial S(w,t)}{\partial w} > 0 \\
\wedge \frac{\partial D(w,a)}{\partial a} = 1 \wedge \frac{\partial S(w,t)}{\partial t} = 1 \\
\wedge \frac{d}{dact}\left(D(w,a) = q = S(w,t)\right) \\
\wedge \frac{d}{dhyp}\left(D(w,a) = q = S(w,t)\right) \\
\wedge \frac{dt}{dact} = \frac{dt}{dhyp} \wedge \frac{da}{dhyp} = 0 \\
\wedge \frac{dq}{dhyp} < \frac{1}{2}\frac{dq}{dact} < 0 \\
H \equiv \frac{dw}{dact} > 0.
\end{aligned}
$$

Assumptions:

- Usual slope restrictions.
- Standard normalizations.
- Scenarios both move labour market equilibrium over course of recession.
- Scenarios have the same tax change but only the *act* scenario has a demand shift.
- Majority of the reduction in labour was due to supply.

Hypothesis: wages are higher at the end of the recession.

Example #0013 Logic

$$
\begin{aligned}
A \equiv\ & \frac{\partial D(w,a)}{\partial w} < 0 \wedge \frac{\partial S(w,t)}{\partial w} > 0 \\
& \wedge \frac{\partial D(w,a)}{\partial a} = 1 \wedge \frac{\partial S(w,t)}{\partial t} = 1 \\
& \wedge \frac{d}{dact}(D(w,a) = q = S(w,t)) \\
& \wedge \frac{d}{dhyp}(D(w,a) = q = S(w,t)) \\
& \wedge \frac{dt}{dact} = \frac{dt}{dhyp} \wedge \frac{da}{dhyp} = 0 \\
& \wedge \frac{dq}{dhyp} < \frac{1}{2}\frac{dq}{dact} < 0 \\
H \equiv\ & \frac{dw}{dact} > 0.
\end{aligned}
$$

Assumptions:

- Usual slope restrictions.
- Standard normalizations.
- Scenarios both move labour market equilibrium over course of recession.
- Scenarios have the same tax change but only the *act* scenario has a demand shift.
- Majority of the reduction in labour was due to supply.

Hypothesis: wages are higher at the end of the recession.

Example #0013 Logic

$$A \equiv \frac{\partial D(w,a)}{\partial w} < 0 \wedge \frac{\partial S(w,t)}{\partial w} > 0$$
$$\wedge \frac{\partial D(w,a)}{\partial a} = 1 \wedge \frac{\partial S(w,t)}{\partial t} = 1$$
$$\wedge \frac{d}{dact}(D(w,a) = q = S(w,t))$$
$$\wedge \frac{d}{dhyp}(D(w,a) = q = S(w,t))$$
$$\wedge \frac{dt}{dact} = \frac{dt}{dhyp} \wedge \frac{da}{dhyp} = 0$$
$$\wedge \frac{dq}{dhyp} < \frac{1}{2}\frac{dq}{dact} < 0$$
$$H \equiv \frac{dw}{dact} > 0.$$

Assumptions:

- Usual slope restrictions.
- Standard normalizations.
- Scenarios both move labour market equilibrium over course of recession.
- Scenarios have the same tax change but only the *act* scenario has a demand shift.
- Majority of the reduction in labour was due to supply.

Hypothesis: wages are higher at the end of the recession.

Example #0013 Logic

$$A \equiv \frac{\partial D(w,a)}{\partial w} < 0 \wedge \frac{\partial S(w,t)}{\partial w} > 0$$
$$\wedge \frac{\partial D(w,a)}{\partial a} = 1 \wedge \frac{\partial S(w,t)}{\partial t} = 1$$
$$\wedge \frac{d}{dact}(D(w,a) = q = S(w,t))$$
$$\wedge \frac{d}{dhyp}(D(w,a) = q = S(w,t))$$
$$\wedge \frac{dt}{dact} = \frac{dt}{dhyp} \wedge \frac{da}{dhyp} = 0$$
$$\wedge \frac{dq}{dhyp} < \frac{1}{2}\frac{dq}{dact} < 0$$
$$H \equiv \frac{dw}{dact} > 0.$$

Assumptions:

- Usual slope restrictions.
- Standard normalizations.
- Scenarios both move labour market equilibrium over course of recession.
- Scenarios have the same tax change but only the *act* scenario has a demand shift.
- Majority of the reduction in labour was due to supply.

Hypothesis: wages are higher at the end of the recession.

Example #0013 Logic

$$
\begin{aligned}
A \equiv\ & \frac{\partial D(w,a)}{\partial w} < 0 \wedge \frac{\partial S(w,t)}{\partial w} > 0 \\
& \wedge \frac{\partial D(w,a)}{\partial a} = 1 \wedge \frac{\partial S(w,t)}{\partial t} = 1 \\
& \wedge \frac{d}{dact}(D(w,a) = q = S(w,t)) \\
& \wedge \frac{d}{dhyp}(D(w,a) = q = S(w,t)) \\
& \wedge \frac{dt}{dact} = \frac{dt}{dhyp} \wedge \frac{da}{dhyp} = 0 \\
& \wedge \frac{dq}{dhyp} < \frac{1}{2}\frac{dq}{dact} < 0 \\
H \equiv\ & \frac{dw}{dact} > 0.
\end{aligned}
$$

Assumptions:

- Usual slope restrictions.
- Standard normalizations.
- Scenarios both move labour market equilibrium over course of recession.
- Scenarios have the same tax change but only the *act* scenario has a demand shift.
- Majority of the reduction in labour was due to supply.

Hypothesis: wages are higher at the end of the recession.

Example #0013 Logic

$$
\begin{aligned}
A \equiv\ & \frac{\partial D(w,a)}{\partial w} < 0 \wedge \frac{\partial S(w,t)}{\partial w} > 0 \\
& \wedge \frac{\partial D(w,a)}{\partial a} = 1 \wedge \frac{\partial S(w,t)}{\partial t} = 1 \\
& \wedge \frac{d}{dact}(D(w,a) = q = S(w,t)) \\
& \wedge \frac{d}{dhyp}(D(w,a) = q = S(w,t)) \\
& \wedge \frac{dt}{dact} = \frac{dt}{dhyp} \wedge \frac{da}{dhyp} = 0 \\
& \wedge \frac{dq}{dhyp} < \frac{1}{2}\frac{dq}{dact} < 0 \\
H \equiv\ & \frac{dw}{dact} > 0.
\end{aligned}
$$

Assumptions:

- Usual slope restrictions.
- Standard normalizations.
- Scenarios both move labour market equilibrium over course of recession.
- Scenarios have the same tax change but only the *act* scenario has a demand shift.
- Majority of the reduction in labour was due to supply.

Hypothesis: wages are higher at the end of the recession.

Example #0013 Logic

$$
\begin{aligned}
A \equiv\ & \frac{\partial D(w,a)}{\partial w} < 0 \wedge \frac{\partial S(w,t)}{\partial w} > 0 \\
& \wedge \frac{\partial D(w,a)}{\partial a} = 1 \wedge \frac{\partial S(w,t)}{\partial t} = 1 \\
& \wedge \frac{d}{dact}\left(D(w,a) = q = S(w,t)\right) \\
& \wedge \frac{d}{dhyp}\left(D(w,a) = q = S(w,t)\right) \\
& \wedge \frac{dt}{dact} = \frac{dt}{dhyp} \wedge \frac{da}{dhyp} = 0 \\
& \wedge \frac{dq}{dhyp} < \frac{1}{2}\frac{dq}{dact} < 0 \\
H \equiv\ & \frac{dw}{dact} > 0.
\end{aligned}
$$

Assumptions:

- Usual slope restrictions.
- Standard normalizations.
- Scenarios both move labour market equilibrium over course of recession.
- Scenarios have the same tax change but only the *act* scenario has a demand shift.
- Majority of the reduction in labour was due to supply.

Hypothesis: wages are higher at the end of the recession.

Example #0013 Logic

$$
\begin{aligned}
A \equiv\ & \frac{\partial D(w,a)}{\partial w} < 0 \wedge \frac{\partial S(w,t)}{\partial w} > 0 \\
& \wedge \frac{\partial D(w,a)}{\partial a} = 1 \wedge \frac{\partial S(w,t)}{\partial t} = 1 \\
& \wedge \frac{d}{dact}\left(D(w,a) = q = S(w,t)\right) \\
& \wedge \frac{d}{dhyp}\left(D(w,a) = q = S(w,t)\right) \\
& \wedge \frac{dt}{dact} = \frac{dt}{dhyp} \wedge \frac{da}{dhyp} = 0 \\
& \wedge \frac{dq}{dhyp} < \frac{1}{2}\frac{dq}{dact} < 0
\end{aligned}
$$

$$
H \equiv \frac{dw}{dact} > 0.
$$

Assumptions:

- Usual slope restrictions.
- Standard normalizations.
- Scenarios both move labour market equilibrium over course of recession.
- Scenarios have the same tax change but only the *act* scenario has a demand shift.
- Majority of the reduction in labour was due to supply.

Hypothesis: wages are higher at the end of the recession.

Example #0013 Result

We may view this as a Tarski formula in 12 variables,

$$\Lambda = \left\{ \frac{da}{dact}, \frac{da}{dhyp}, \frac{dt}{dact}, \frac{dt}{dhyp}, \frac{dq}{dact}, \frac{dq}{dhyp}, \frac{dw}{dact}, \frac{dw}{dhyp}, \frac{\partial D(w,a)}{\partial a}, \frac{\partial S(w,t)}{\partial t}, \frac{\partial D(w,a)}{\partial w}, \frac{\partial S(w,t)}{\partial w} \right\},$$

Each is representing a partial derivative describing the supply and demand function or a total derivative indicating a change over time within a scenario.
Evaluating the two existential problems shows that both examples and counterexamples exist: the theorem is not universally true.

Example #0013 Exploration

If we leave $\frac{\partial D(w,a)}{\partial w}$ and $\frac{\partial S(w,t)}{\partial w}$ as free variables then QE recovers a disjunction of three quantifier-free formulae. Two of them contradict the assumptions, but the third, below, could be added to A to guarantee the truth of H.

$$\frac{\partial S(w,t)}{\partial w} \geq -\frac{\partial D(w,a)}{\partial w} > 0.$$

The added assumption states that labour supply is at least as sensitive to wages as labour demand.

The algebra and logical reasoning above is not political! Any politics comes with whether the assumptions hold. We cannot remove the politics, but at least gain clarity over which arguments are political and which not.

Economics Example References

C. Mulligan, R. Bradford, J.H. Davenport, M. England, and Z. Tonks.
Non-linear real arithmetic benchmarks derived from automated reasoning in economics.
In Proc. SC^2 '1), CEUR-WS 2189, pages 48–60, 2018.
URL: http://ceur-ws.org/Vol-2189/.

C.B. Mulligan, J.H. Davenport, and M. England.
TheoryGuru: A Mathematica package to apply quantifier elimination technology to economics.
In Proc. ICMS 2018, LNCS 10931, pages 369–378. Springer, 2018.
URL: https://doi.org/10.1007/978-3-319-96418-8_44.

Thanks for Listening

Contact Details

Matthew.England@coventry.ac.uk

https://matthewengland.coventry.domains/