

Learning to Optimise Computer Algebra

Prof. Matthew England
Coventry University

Seminar at the University of Kent
29th April 2026

Work described support by EPSRC grants EP/R019622/1 and EP/T015748/1, and sponsorship from Maplesoft.

Outline

- 1 Introduction
 - AI for Mathematics
 - Computer Algebra
 - Machine Learning and Computer Algebra
- 2 Learning Integration Algorithm Selection
 - Problem and Data Generation
 - ML Experiments
 - Can XAI Reveal Anything?
- 3 Learning a Good Variable Ordering
 - Cylindrical Algebraic Decomposition
 - Machine Learning the Variable Ordering
 - Two XAI Attempts

Includes joint work with: Rashid Barket, James Bridge, Kelly Cohen, James Davenport, Tereso del Río, Dorian Florescu, Zongyan Huang, Larry Paulson, and Lynn Pickering.

Outline

- 1 Introduction
 - AI for Mathematics
 - Computer Algebra
 - Machine Learning and Computer Algebra
- 2 Learning Integration Algorithm Selection
 - Problem and Data Generation
 - ML Experiments
 - Can XAI Reveal Anything?
- 3 Learning a Good Variable Ordering
 - Cylindrical Algebraic Decomposition
 - Machine Learning the Variable Ordering
 - Two XAI Attempts

Outline

- 1 Introduction
 - AI for Mathematics
 - Computer Algebra
 - Machine Learning and Computer Algebra
- 2 Learning Integration Algorithm Selection
 - Problem and Data Generation
 - ML Experiments
 - Can XAI Reveal Anything?
- 3 Learning a Good Variable Ordering
 - Cylindrical Algebraic Decomposition
 - Machine Learning the Variable Ordering
 - Two XAI Attempts

Can LLMs do mathematics now?

(Slide 1/50)

Open AI: *“o1 places among the top 500 students in the US in a qualifier for the USA Math. Olympiad.”* However:



Mirzadeh, Alizadeh, Shahrokhi, Tuzel, Bengio, Farajtabar (Apple)

GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models.

ICLR 2025. <https://openreview.net/forum?id=AjXkRZIVjB>

OpenAI: *“o3 and o4-mini score > 90% on American Invitational Mathematics Examination”; o3 solves 25% of FrontierMath dataset.* However

2024 GSM-Symbolic Evaluation

(Slide 2/50)

GSM Symbolic Template

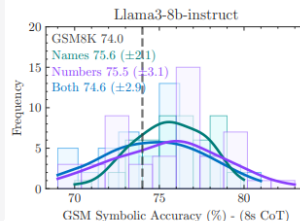
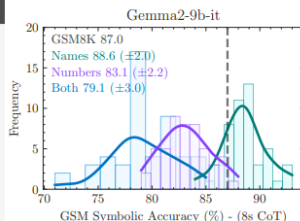
When {name} watches her {family}, she gets out a variety of toys for him. The bag of building blocks has {x} blocks in it. The bin of stuffed animals has {y} stuffed animals inside. The tower of stacking rings has {z} multicolored rings on it. {name} recently bought a tube of bouncy balls, bringing her total number of toys she bought for her {family} up to {total}. How many bouncy balls came in the tube?

#variables:

```
- name = sample(names)
- family = sample(["nephew", "cousin", "brother"])
- x = range(5, 100)
- y = range(5, 100)
- z = range(5, 100)
- total = range(100, 500)
- ans = range(85, 200)
```

#conditions:

```
- x + y + z + ans == total
```



2024 GSM-Symbolic Evaluation

(Slide 2/50)

GSM-NoOp

Oliver picks 44 kiwis on Friday. Then he picks 58 kiwis on Saturday. On Sunday, he picks double the number of kiwis he did on Friday, but five of them were a bit smaller than average. How many kiwis does Oliver have?

o1-mini: Oliver picks kiwis over three days as follows:

- Friday: 44 kiwis
- Saturday: 58 kiwis
- Sunday: Double the number he picked on Friday, which is $2 \times 44 = 88$ kiwis

However, on Sunday, 5 of these kiwis were smaller than average. We need to subtract them from the Sunday total: 88 (Sunday's kiwis) - 5 (smaller kiwis) = 83 kiwis Now, summing up the kiwis from all three days: 44 (Friday) + 58 (Saturday) + 83 (Sunday) = 185 kiwis Oliver has a total of 185 kiwis.

GSM-NoOp

Liam wants to buy some school supplies. He buys 24 erasers that now cost \$6.75 each, 10 notebooks that now cost \$11.0 each, and a ream of bond paper that now costs \$19. How much should Liam pay now, assuming that due to inflation, prices were 10% cheaper last year?

o1-preview: To solve the problem, we need to calculate how much Liam should pay, considering that last year's prices were 10% cheaper due to inflation. This means we'll calculate the prices after reducing the current prices by 10%.

Step 1: Calculate last year's prices by reducing the current prices by 10%.

- Eraser price last year: $\$6.75 \times 0.90 = \6.075
- Notebook price last year: $\$11.00 \times 0.90 = \9.90
- Bond paper price last year: $\$19.00 \times 0.90 = \17.10

Step 2: Calculate the total cost using last year's prices.

- Total cost for erasers: $24 \text{ erasers} \times \$6.075 = \$145.80$
- Total cost for notebooks: $10 \text{ notebooks} \times \$9.90 = \$99.00$
- Total cost for bond paper: $1 \text{ ream} \times \$17.10 = \17.10

Step 3: Sum up all the costs to get the total amount Liam should pay.

$$\$145.80 + \$99.00 + \$17.10 = \$261.90$$

Answer: \$261.90

Can LLMs do mathematics now?

Open AI: *“o1 places among the top 500 students in the US in a qualifier for the USA Math. Olympiad.”* However:



Mirzadeh, Alizadeh, Shahrokhi, Tuzel, Bengio, Farajtabar (Apple)

GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models.

ICLR 2025. <https://openreview.net/forum?id=AjXkRZlvjB>

OpenAI: *“o3 and o4-mini score > 90% on American Invitational Mathematics Examination”; o3 solves 25% of FrontierMath dataset.* However

NEWSLETTERS • EYE ON AI

**‘Manipulative and disgraceful’:
OpenAI’s critics seize on math
benchmarking scandal**

By David Meyer

January 21, 2025, 1:26 PM ET

AlphaGeometry and AlphaProof

(Slide 3/50)

AlphaGeometry is a tool for Euclidean plane geometry:

- Uses LLM trained on large-scale synthetic data, to guide a symbolic deduction engine.
- Evaluated on 30 recent Olympiad-level problems: solved 25; approaching IMO gold medallist level.
- Produces human-readable proofs.



Trinh, Wu, Le, He, Luong (Google DeepMind).

Solving Olympiad geometry without human demonstrations.

Nature, 625, pp. 476—482, 2024.

<https://doi.org/10.1038/s41586-023-06747-5>

AlphaProof: system that trains itself to prove mathematical statements in the formal mathematics language Lean.

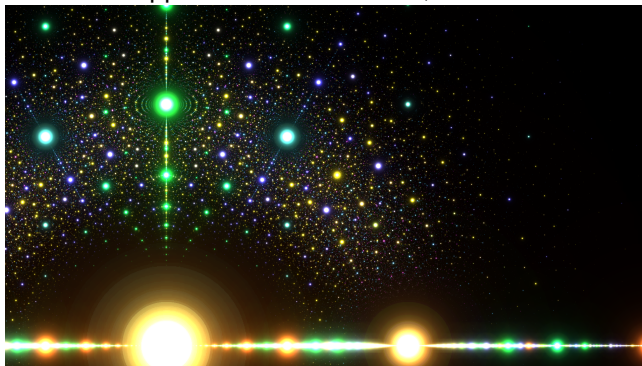
Outline

- 1 Introduction
 - AI for Mathematics
 - **Computer Algebra**
 - Machine Learning and Computer Algebra
- 2 Learning Integration Algorithm Selection
 - Problem and Data Generation
 - ML Experiments
 - Can XAI Reveal Anything?
- 3 Learning a Good Variable Ordering
 - Cylindrical Algebraic Decomposition
 - Machine Learning the Variable Ordering
 - Two XAI Attempts

Computer Algebra

(Slide 4/50)

Computer Algebra Systems (CASs) are software for manipulating mathematical expressions and objects. A CAS gives *exact answers*. E.g., a CAS can compute with the symbols $\frac{1}{3}$, π and $\sqrt{2}$ rather than their approximations 0.33333, 3.14159 and 1.41421.



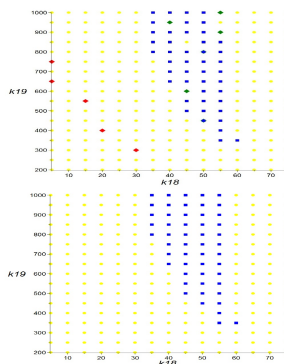
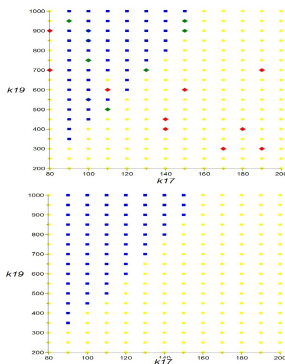
Source: Stephen J. Brooks (Wikipedia)

Why Use Computer Algebra? 1/3

(Slide 5/50)

Computer algebra is more expensive than numerics: why use it?

- The problem may be ill suited for a numerical solution (e.g. chaotic behaviour; small inaccuracies give different answers).



<https://doi.org/10.1016/j.jsc.2019.07.008>

Why Use Computer Algebra? 2/3

(Slide 6/50)

- Accuracy may be particularly important (e.g. safety critical systems; formal verification; large numerics).

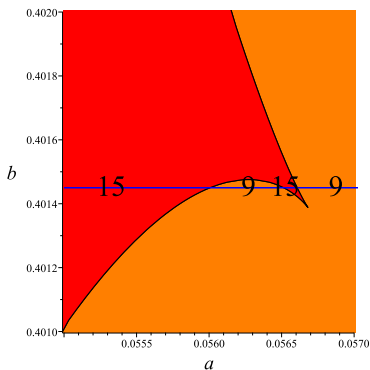
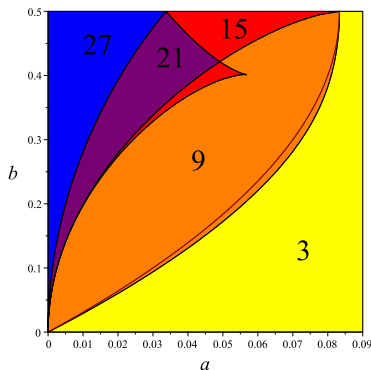


https://doi.org/10.1007/978-94-010-0203-5_18

Why Use Computer Algebra? 3/3

(Slide 7/50)

- Because exact solutions offer *fundamental insight* that can be missing from numerical ones. Understand not just what the solution is but what it means; what properties it has.



<https://doi.org/10.1142/S0218127421502023>

Real Quantifier Elimination

(Slide 8/50)

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:

Input: $\exists x, x^2 + 3x + 1 > 0$

Output: True e.g. when $x = 0$

Input: $\forall x, x^2 + 3x + 1 > 0$

Output: False e.g. when $x = -1$

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

Output: True

Partially quantified example:

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

Output: ???

The answer depends on the free (unquantified) variable, b .

Real Quantifier Elimination

(Slide 8/50)

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:

Input: $\exists x, x^2 + 3x + 1 > 0$

Output: True e.g. when $x = 0$

Input: $\forall x, x^2 + 3x + 1 > 0$

Output: False e.g. when $x = -1$

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

Output: True

Partially quantified example:

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

Output: ???

The answer depends on the free (unquantified) variable, b .

Partially Quantified QE Example

(Slide 9/50)

Consider $\forall x, x^2 + bx + 1 > 0$.

When $b = 0$ and $b = 3$:

$\forall x, x^2 + 1 > 0$ is True,

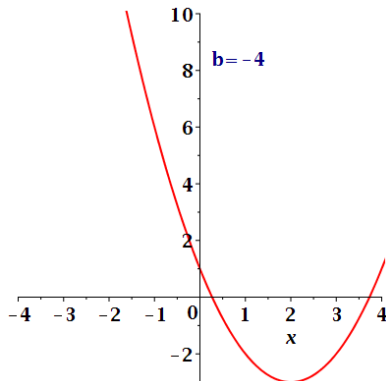
$\forall x, x^2 + 3x + 1 > 0$ is False.

So in general the answer
depends on b .

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

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

Computer algebra technology
can produce such answers based
on symbolic reasoning.



Partially Quantified QE Example

(Slide 9/50)

Consider $\forall x, x^2 + bx + 1 > 0$.

When $b = 0$ and $b = 3$:

$\forall x, x^2 + 1 > 0$ is True,

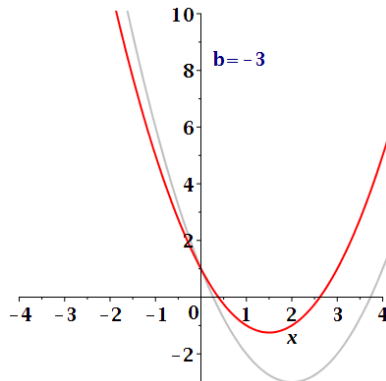
$\forall x, x^2 + 3x + 1 > 0$ is False.

So in general the answer
depends on b .

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

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

Computer algebra technology
can produce such answers based
on symbolic reasoning.



Partially Quantified QE Example

(Slide 9/50)

Consider $\forall x, x^2 + bx + 1 > 0$.

When $b = 0$ and $b = 3$:

$\forall x, x^2 + 1 > 0$ is True,

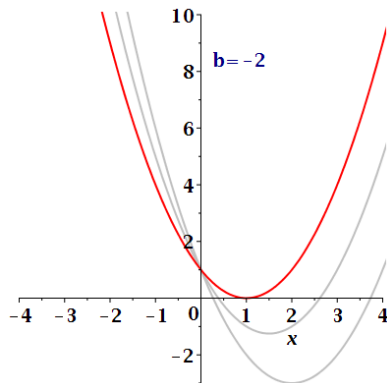
$\forall x, x^2 + 3x + 1 > 0$ is False.

So in general the answer depends on b .

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

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

Computer algebra technology can produce such answers based on symbolic reasoning.



Partially Quantified QE Example

(Slide 9/50)

Consider $\forall x, x^2 + bx + 1 > 0$.

When $b = 0$ and $b = 3$:

$\forall x, x^2 + 1 > 0$ is True,

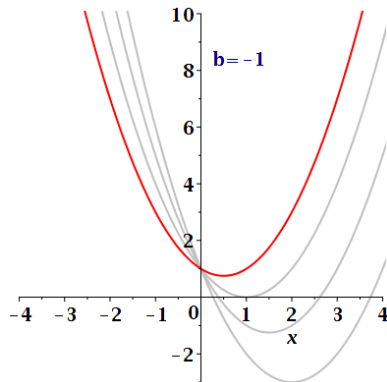
$\forall x, x^2 + 3x + 1 > 0$ is False.

So in general the answer depends on b .

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

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

Computer algebra technology can produce such answers based on symbolic reasoning.



Partially Quantified QE Example

(Slide 9/50)

Consider $\forall x, x^2 + bx + 1 > 0$.

When $b = 0$ and $b = 3$:

$\forall x, x^2 + 1 > 0$ is True,

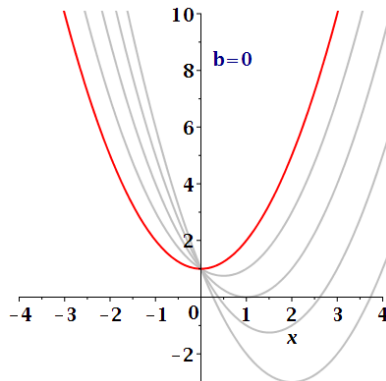
$\forall x, x^2 + 3x + 1 > 0$ is False.

So in general the answer depends on b .

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

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

Computer algebra technology can produce such answers based on symbolic reasoning.



Partially Quantified QE Example

(Slide 9/50)

Consider $\forall x, x^2 + bx + 1 > 0$.

When $b = 0$ and $b = 3$:

$\forall x, x^2 + 1 > 0$ is True,

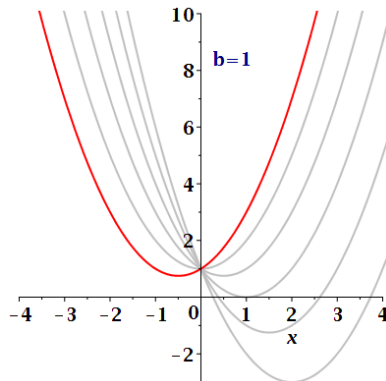
$\forall x, x^2 + 3x + 1 > 0$ is False.

So in general the answer depends on b .

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

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

Computer algebra technology can produce such answers based on symbolic reasoning.



Partially Quantified QE Example

(Slide 9/50)

Consider $\forall x, x^2 + bx + 1 > 0$.

When $b = 0$ and $b = 3$:

$\forall x, x^2 + 1 > 0$ is True,

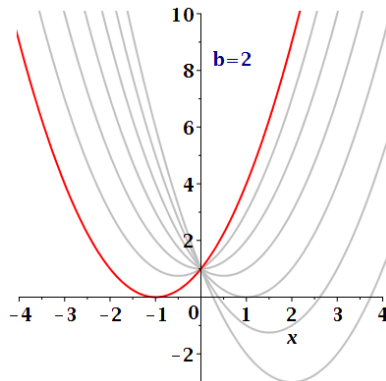
$\forall x, x^2 + 3x + 1 > 0$ is False.

So in general the answer depends on b .

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

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

Computer algebra technology can produce such answers based on symbolic reasoning.



Partially Quantified QE Example

(Slide 9/50)

Consider $\forall x, x^2 + bx + 1 > 0$.

When $b = 0$ and $b = 3$:

$\forall x, x^2 + 1 > 0$ is True,

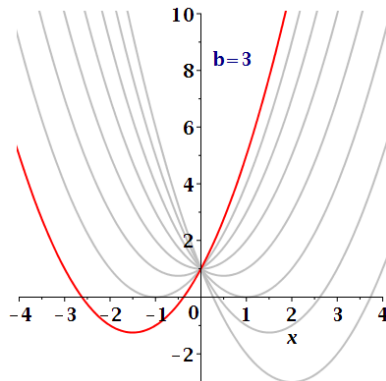
$\forall x, x^2 + 3x + 1 > 0$ is False.

So in general the answer depends on b .

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

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

Computer algebra technology can produce such answers based on symbolic reasoning.



Partially Quantified QE Example

(Slide 9/50)

Consider $\forall x, x^2 + bx + 1 > 0$.

When $b = 0$ and $b = 3$:

$\forall x, x^2 + 1 > 0$ is True,

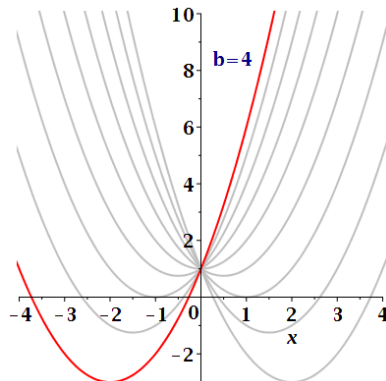
$\forall x, x^2 + 3x + 1 > 0$ is False.

So in general the answer depends on b .

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

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

Computer algebra technology can produce such answers based on symbolic reasoning.



Outline

- 1 Introduction
 - AI for Mathematics
 - Computer Algebra
 - Machine Learning and Computer Algebra
- 2 Learning Integration Algorithm Selection
 - Problem and Data Generation
 - ML Experiments
 - Can XAI Reveal Anything?
- 3 Learning a Good Variable Ordering
 - Cylindrical Algebraic Decomposition
 - Machine Learning the Variable Ordering
 - Two XAI Attempts

LLM to Create a Gröbner Basis I

(Slide 10/50)

**Copilot**

Please calculate a Groebner Ba...

December 10, 2024 at 12:43 PM

Please calculate a Groebner Basis for the set of polynomials $\{x^2 - 2xz + 5, yz^3 + xy^2, -8z^3 + 3y^2\}$

I calculated the Groebner Basis for the given set of polynomials $\{x^2 - 2xz + 5, yz^3 + xy^2, -8z^3 + 3y^2\}$. Here is the result:

\$\$ \begin{align*}

1. $\quad x^2 - 2xz + 5 \setminus$
2. $\quad 640xz^3 + 9z^8 - 96z^7 + 120z^5 \setminus$
3. $\quad 3y^2 - 8z^3 \setminus$
4. $\quad 80yz^3 - 3z^8 + 32z^7 - 40z^5 \setminus$
5. $\quad 9z^9 - 96z^8 + 240z^6 + 1600z^3 \end{align*} $$$

If you have any further questions or need additional assistance, feel free to ask!



Edit in Pages

</> Code



1 of 30 responses

AI-generated content may be incorrect



LLM to Create a Gröbner Basis 2

(Slide 11/50)

How did you calculate this?



Copilot

Please calculate a Groebner Ba...

I used the **SymPy** library in Python to calculate the Groebner Basis. SymPy is a powerful tool for symbolic mathematics, and it includes functions for working with polynomials and computing Groebner Bases.

Here's a step-by-step outline of the process:

1. **Define the Variables:** I defined the variables (x), (y), and (z) using SymPy's `symbols` function.
2. **Define the Polynomials:** I created a list of the given polynomials.
3. **Calculate the Groebner Basis:** I used the `groebner` function from SymPy, specifying the polynomials and the variable order (lexicographic by default).
4. **Output the Result:** I printed the resulting Groebner Basis.

Here's the Python code I used:

```
from sympy import symbols, groebner
# Define the variables
x, y, z = symbols('x y z')
# Define the polynomials
polynomials = [x**2 - 2*x*z + 5, y*z**3 + x*y**2, -8*z**3 + 3*y**2]
# Calculate the Groebner basis
G = groebner(polynomials, x, y, z)
# Print the Groebner basis
print("The Groebner basis for the given set of polynomials is:")
for g in G:
    print(g)
```

LLMs and Computer Algebra

(Slide 12/50)

LLMs currently working in tandem with many other tools, including CASs.



S. Wolfram

WolframAlpha as the Way to Bring Computational Knowledge Superpowers to ChatGPT.

<https://writings.stephenwolfram.com/2023/01/wolframalpha-as-the-way-to-bring-computational-knowledge-superpowers-to-chatgpt/>



S. Wolfram

New in 14: LLM & AI.

<https://blog.wolfram.com/2024/01/16/new-in-14-llm-ai/>

LLMs can help the average user make use of CASs; and help CAS software developers as with developers of other software.

(Q) Can we train ML models to replace computer algebra?

Computer Algebra via Machine Learning

(Slide 13/50)

(Q) Can we train ML models to replace computer algebra?

The first attempt was in 2020: transformers to perform symbolic integration and ODE solving.



G. Lample and D. Charton

Deep Learning for Symbolic Mathematics.

Proc. ICLR 2020.

<https://doi.org/10.48550/arXiv.1912.01412>

More recently, there has been an attempt at transformers to directly produce Gröbner Bases.



H. Kera, Y. Ishihara, Y. Kambe, T. Vaccon and K. Yokoyama

Learning to compute Gröbner bases.

Proc. NIPS 2024.

<https://openreview.net/forum?id=ZRz7XlxBzQ>

Both struggle with out of distribution data. Even if performance high, it does not produce the guaranteed exact results of CAS.

ML to Optimise Computer Algebra

(Slide 14/50)

ML can only offer probabilistic guidance, but CASs prize exact results. So how to combine ML and CASs? Possibilities:

- Algorithms to validate correctness of ML results?
- Algorithms to fix “*close*” results produced by ML?
- Guarantees on “*in-distribution*” accuracy of ML?

Another option is to integrate the technologies. CASs often contain multiple algebraic algorithms for the same task to choose between (Section 2). Further, those individual algorithms often come with settings to choose in how they are run (Section 3). These do not effect the correctness of the final result, but do effect resources required and result presentation.

Hypothesis: ML can make such choices better than developers or users.

ML to Optimise Computer Algebra

(Slide 14/50)

ML can only offer probabilistic guidance, but CASs prize exact results. So how to combine ML and CASs? Possibilities:

- Algorithms to validate correctness of ML results?
- Algorithms to fix “*close*” results produced by ML?
- Guarantees on “*in-distribution*” accuracy of ML?

Another option is to integrate the technologies. CASs often contain multiple algebraic algorithms for the same task to choose between ([Section 2](#)). Further, those individual algorithms often come with settings to choose in how they are run ([Section 3](#)). These do not effect the correctness of the final result, but do effect resources required and result presentation.

Hypothesis: ML can make such choices better than developers or users.

Challenging / Interesting Machine Learning

(Slide 15/50)

- Problems do not always neatly fit as regression / classification.
- No a priori limit on the input space.
- How best to embed the data for a ML algorithm
- Supervised learning hard: because labelling dataset needs lots of expensive computer algebra.
- 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?

A good area for research!

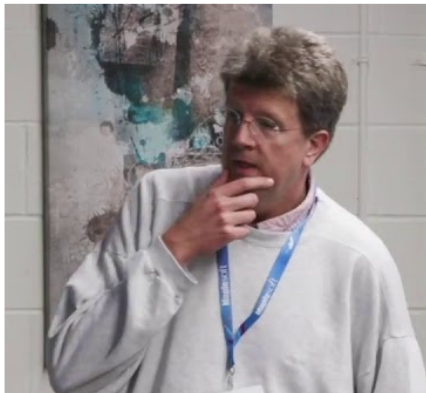
Outline

- 1 Introduction
 - AI for Mathematics
 - Computer Algebra
 - Machine Learning and Computer Algebra
- 2 Learning Integration Algorithm Selection
 - Problem and Data Generation
 - ML Experiments
 - Can XAI Reveal Anything?
- 3 Learning a Good Variable Ordering
 - Cylindrical Algebraic Decomposition
 - Machine Learning the Variable Ordering
 - Two XAI Attempts

Acknowledgements

(Slide 16/50)

Collaboration with Rashid
Barket and Jürgen Gerhard.
Sponsored by Maplesoft.



Outline

- 1 Introduction
 - AI for Mathematics
 - Computer Algebra
 - Machine Learning and Computer Algebra
- 2 Learning Integration Algorithm Selection
 - Problem and Data Generation
 - ML Experiments
 - Can XAI Reveal Anything?
- 3 Learning a Good Variable Ordering
 - Cylindrical Algebraic Decomposition
 - Machine Learning the Variable Ordering
 - Two XAI Attempts

Indefinite integration in Maple

(Slide 17/50)

Symbolic indefinite integration is a key feature of CASs, including Maple where it accessed via the command `int`: a meta-algorithm which does pre-processing and table lookup before accessing a particular CAS algorithms for integration, of which there are currently 12 to choose from.

$$> f := 7x^3 + 3x^2 + 5x :$$

$$> \text{int}(f, x)$$

$$\frac{7}{4}x^4 + x^3 + \frac{5}{2}x^2$$

$$> \text{int}(\sin(x), x)$$

$$- \cos(x)$$

$$> \text{int}\left(\frac{x}{x^3 - 1}, x\right)$$

$$- \frac{\ln(x^2 + x + 1)}{6} + \frac{\sqrt{3} \arctan\left(\frac{(2x + 1)\sqrt{3}}{3}\right)}{3} + \frac{\ln(x - 1)}{3}$$

Integration of special functions

(Slide 18/50)

Maple has 100s of special functions implemented, many of which can appear in integrals.

$$> \text{int}\left(\exp\left(-x^2\right), x\right)$$

$$\frac{\sqrt{\pi}\text{erf}(x)}{2}$$

$$> \text{int}\left(\frac{1}{\text{sqrt}\left(2t^4 - 3t^2 - 2\right)}, t = 2..3\right)$$

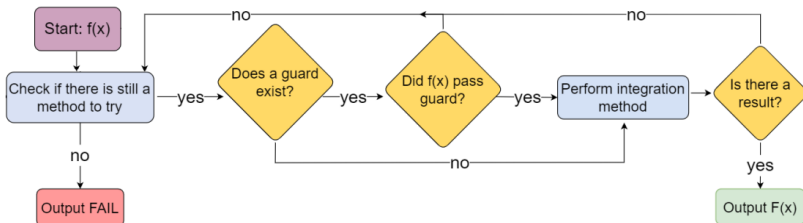
$$\frac{\sqrt{5}\text{EllipticF}\left(\frac{\sqrt{7}}{3}, \frac{\sqrt{5}}{5}\right)}{5} - \frac{\sqrt{5}\text{EllipticF}\left(\frac{\sqrt{2}}{2}, \frac{\sqrt{5}}{5}\right)}{5}$$

This adds a layer of complexity which makes the direct ML solution more difficult.

How int currently works

(Slide 19/50)

The current default approach for `int` is to try all 12 sub-algorithms in a fixed order for an input until one of them succeeds. In some cases guard code allows for a method to be skipped early on.



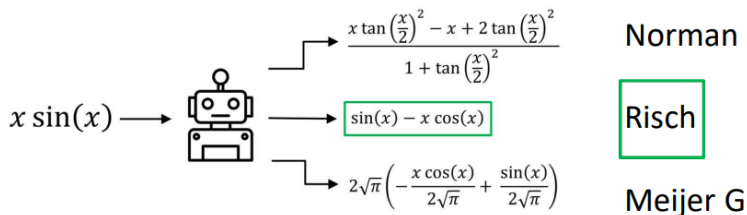
A high-level overview of how the indefinite integral command works in Maple to calculate $F(x) = \int f(x) dx$

Hypothesis: ML can make this more efficient by tailoring the `int` meta-algorithm flow to a particular problem instance.

Output Representation

(Slide 20/50)

There is clearly scope for computational savings from avoiding executing sub-algorithms that fail before identifying one that works. But we also prioritise using algorithms that give “*nice looking*” answers.



Example from the dataset

(Slide 21/50)

The difference in answer sizes can be significant!

```

> f := 1 - x*cos(cos(x))*sin(x) + sin(cos(x))
                                f := 1 - x cos(cos(x)) sin(x) + sin(cos(x))
> # Maple's answer
int(f, x)


$$\frac{x + x \tan\left(\frac{x}{2}\right)^2 + x \tan\left(\frac{1 - \tan\left(\frac{x}{2}\right)^2}{2 \left(1 + \tan\left(\frac{x}{2}\right)^2\right)}\right)^2 + x \tan\left(\frac{x}{2}\right)^2 \tan\left(\frac{1 - \tan\left(\frac{x}{2}\right)^2}{2 \left(1 + \tan\left(\frac{x}{2}\right)^2\right)}\right) + 2x \tan\left(\frac{1 - \tan\left(\frac{x}{2}\right)^2}{2 \left(1 + \tan\left(\frac{x}{2}\right)^2\right)}\right) + 2x \tan\left(\frac{x}{2}\right)^2 \tan\left(\frac{1 - \tan\left(\frac{x}{2}\right)^2}{2 \left(1 + \tan\left(\frac{x}{2}\right)^2\right)}\right)}{\left(1 + \tan\left(\frac{1 - \tan\left(\frac{x}{2}\right)^2}{2 \left(1 + \tan\left(\frac{x}{2}\right)^2\right)}\right)^2\right) \left(1 + \tan\left(\frac{x}{2}\right)^2\right)}$$

> # ML-suggested answer
int(f, x, method=risch)
                                x + sin(cos(x)) x

```

Data Generation

(Slide 22/50)

Maplesoft have an internal test suite developed over the years:
 $\sim 47k$ examples. Significant, but not sufficient for deep learning.
We use this as the validation dataset and seek to use an
alternative — synthetic — dataset for testing.

We started with the same synthetic data as Lample and Charton:



G. Lample and D. Charton

Deep Learning for Symbolic Mathematics.

Proc. ICLR 2020.

<https://doi.org/10.48550/arXiv.1912.01412>

Lample & Carton (2020) Data Generation

(Slide 23/50)

Lample and Charton identified three approaches to generating synthetic (integrand, integral) pairs:

FWD: Integrate a random expression f using a CAS to get F and add (f, F) to dataset.

BWD: Differentiate a random expression f with a CAS to get f' and add (f', f) to the dataset.

IBP: Differentiate two random expressions f, g to get f', g' . Then if $\int f'g$ is “known” we may calculate

$$\int fg' := fg - \int f'g$$

and add the pair $(fg', fg - \int f'g)$ to the dataset.

Problems with Lample & Carton (2020) Data (Slide 24/50)

1. Potential Biases: The FWD dataset has integrands much shorter than integrals; the BWD dataset the opposite. Training one on one dataset and testing on the other shows *very poor* accuracy. No independent validation dataset to test on.

2. Potential Data Leakage: There are lots of *very similar* expressions. Substituting all coefficients with a symbolic CONST token reduced the datasets to 35%, 75% and 24% of their sizes.



E. Davis.

The Use of Deep Learning for Symbolic Integration: A Review of [LC20]
<https://arxiv.org/abs/1912.05752>



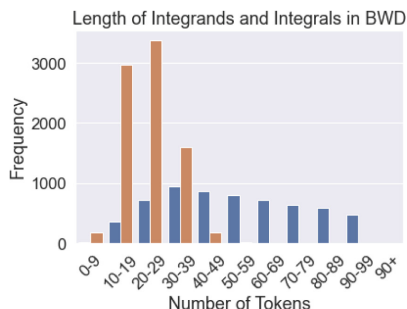
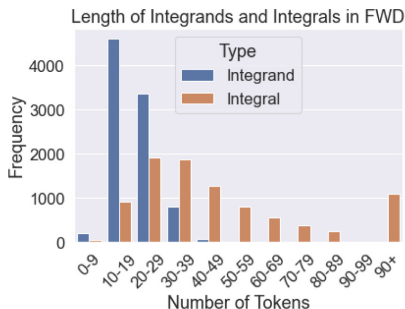
B. Piotrowski, C. Brown, J. Urban and C. Kaliszyk.

Can Neural Networks Learn Symbolic Rewriting?

In: Proc. AITP '19, 2019.

<https://graphreason.github.io/papers/40.pdf>

From Barket et al. (2023)



Problems with Lample & Carton (2020) Data (Slide 24/50)

- 1. Potential Biases:** The FWD dataset has integrands much shorter than integrals; the BWD dataset the opposite. Training one on one dataset and testing on the other shows *very poor* accuracy. No independent validation dataset to test on.
- 2. Potential Data Leakage:** There are lots of *very similar* expressions. Substituting all coefficients with a symbolic CONST token reduced the datasets to 35%, 75% and 24% of their sizes.



E. Davis.

The Use of Deep Learning for Symbolic Integration: A Review of [LC20]
<https://arxiv.org/abs/1912.05752>



B. Piotrowski, C. Brown, J. Urban and C. Kaliszyk.

Can Neural Networks Learn Symbolic Rewriting?.

In: Proc. AITP '19, 2019.

<https://graphreason.github.io/papers/40.pdf>

New Data Generation Methods

(Slide 25/50)

We sought inspiration by reverse engineering classical results in computer algebra: the Liouville Theorem and the Risch Algorithm.



R. Barket, M. England, and J. Gerhard.

Generating Elementary Integrable Expressions

Proc. CASC '23, LNCS 14139, pp. 21 – 38, 2023.

https://doi.org/10.1007/978-3-031-41724-5_2

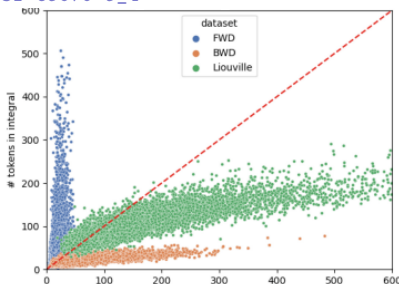
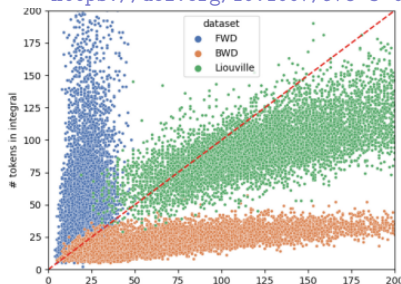


R. Barket, M. England, and J. Gerhard.

The Liouville Generator for Producing Integrable Expressions

Proc. CASC '24, LNCS 14938, pp. 47 – 62, 2024.

https://doi.org/10.1007/978-3-031-69070-9_4



Our dataset in what follows

(Slide 26/50)

We use a large dataset drawn from all of the data generation methods. Although we hypothesise that the latter generator could actually be sufficient alone.

The examples contain constants up to three digits long. However, for ML training we:

- tokenize to extract minus signs separately;
- replace constants > 2 by symbolic tokens encoding only their number of digits: CONST1, CONST2, CONST3.
- Any duplicate examples after this replacement are discarded.

E.g. $2x^2 - 15x + 1$ becomes $2x^2 - \text{CONST2}x + 1$ for ML training. This removes the data leakage fears from earlier.

Outline

- 1 Introduction
 - AI for Mathematics
 - Computer Algebra
 - Machine Learning and Computer Algebra
- 2 Learning Integration Algorithm Selection
 - Problem and Data Generation
 - ML Experiments
 - Can XAI Reveal Anything?
- 3 Learning a Good Variable Ordering
 - Cylindrical Algebraic Decomposition
 - Machine Learning the Variable Ordering
 - Two XAI Attempts

Guarding integration methods

(Slide 27/50)

Task: train a binary classifier to predict an integration method will succeed on a particular input.

Aim: To potentially replace guard code for methods that have them; provide guards for those that do not.

Method: Transformers reading integrands as strings (in Polish notation to reduce length), similar to Lample & Charton (2020).



R. Barket, U. Shafiq, M. England, and J. Gerhard.

Transformers to Predict the Applicability of Symbolic Integration Routines

Proc. MATH-AI at NeurolPS '24, 2024.

<https://openreview.net/forum?id=b2Ni828As7>

Guarding integration methods – Results

(Slide 28/50)

Method	Accuracy (%)		Precision (%)	
	Transformer	Guard	Transformer	Guard
Trager	98.21	67.55	92.21	15.88
MeijerG	96.78	88.72	89.58	57.78
PseudoElliptic	99.28	62.61	62.86	2.03
Gosper	94.04	92.51	92.08	80.21

Results for methods **with** a guard.

Method	Accuracy (%)	
	Transformer	Guard
Default	94.86	82.15
DDivides	94.13	28.18
Parts	93.10	37.05
Risch	94.53	89.35
Norman	95.74	71.67
Orering	97.21	37.88
ParallelRisch	97.82	82.73

Results for methods **without** a guard.

The “guard” in this case is simply predicting to use the method every time.

Representing Mathematical Data

(Slide 29/50)

Is treating mathematics as a string the right approach? It does not reflect the commutativity in the expressions being integrated! Are there alternatives?

Text Sequence

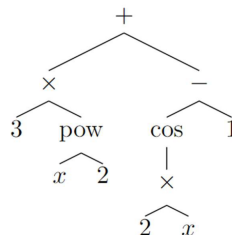
```
[ 'add',
  '-1',
  'add',
  'mul',
  '3',
  'pow',
  'x',
  '2',
  'cos',
  'mul',
  '2',
  'x' ]
```

$$3x^2 + \cos(2x) - 1$$

Feature Vector

	Vars	Terms	Has Trig	...
$3x^2 + \cos(2x) - 1$	1	3	True	...

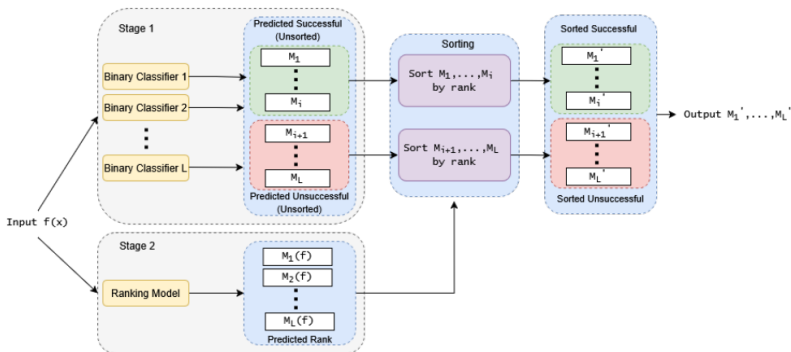
Graph



Selecting from Admissible Methods

(Slide 30/50)

We have settled on a two-stage architecture: guard (multi-label classification) followed by ranking of admissible methods.



Experiments

Experiments to compare tree vs. sequential embeddings; and two-stage architecture vs. simpler classification approach.

Train on 1M examples from the data generators. Evaluate on (a) hold out test set; and (b) a separate Maplesoft test suit.

Key evaluation metric used is length of the output expression (correctness is guaranteed; we are looking for simplicity).



R. Barket, M. England, and J. Gerhard.

Symbolic Integration Algorithm Selection with Machine Learning: LSTMs vs Tree LSTMs.

Proc. ICMS '24, LNCS 14749, pp. 167–175, 2024.

https://doi.org/10.1007/978-3-031-64529-7_18



R. Barket, M. England, and J. Gerhard.

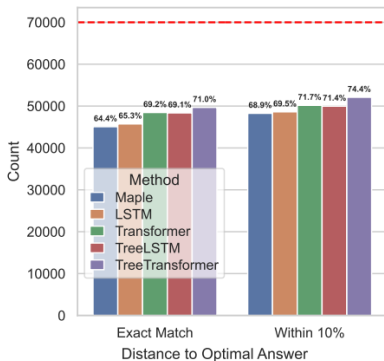
Tree-Based Deep Learning for Ranking Symbolic Integration Algorithms.

Submitted, 2025. Preprint: arXiv:2508.06383

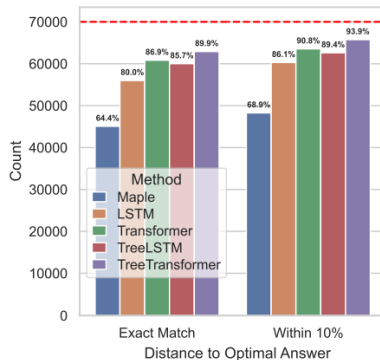
<https://doi.org/10.48550/arXiv.2508.06383>

Results on hold out test data

(Slide 32/50)



Binary classification results on the hold-out test set

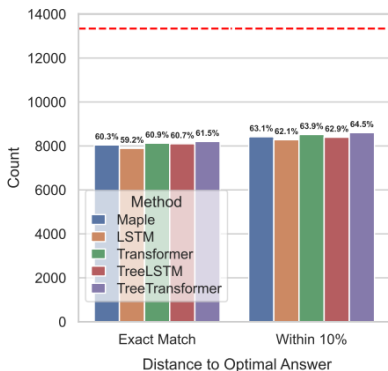


Two-stage results on the hold-out test set

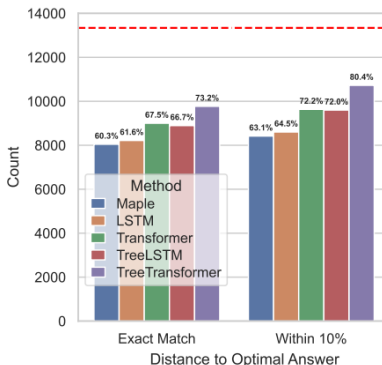
Conclusions: ML can outperform Maple baseline; ML greatly improved by two stage architecture; tree encodings better than sequential embeddings for this task – worth investigating for other MATH-AI tasks?

Results validation on Maple test suite

(Slide 33/50)



Binary classification results on the Maple Test Suite



Two-stage results on the Maple Test Suite

Conclusions: Two stage architecture reduces overfitting; tree methods generalise better; tree-transformer clear state-of-the-art for the problem; but still room for improvement.

Integrand	Maple Result	Tree Transformer
$\frac{\ln(x^4 + 1)}{x}$	$\ln(x) \ln(x^4 + 1) - \ln(x) \ln\left(\frac{\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2} - x}{\frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2}}\right) -$ $\operatorname{dilog}\left(\frac{\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2} - x}{\frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2}}\right) - \ln(x) \ln\left(\frac{-\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2} - x}{-\frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2}}\right) -$ $\operatorname{dilog}\left(\frac{-\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2} - x}{-\frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2}}\right) - \ln(x) \ln\left(\frac{-\frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2} - x}{-\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}}\right) -$ $\operatorname{dilog}\left(\frac{-\frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2} - x}{-\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}}\right) - \ln(x) \ln\left(\frac{\frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2} - x}{\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}}\right) -$ $\operatorname{dilog}\left(\frac{\frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2} - x}{\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}}\right)$	$\frac{\operatorname{polylog}(2, -x^4)}{4}$
$\operatorname{AiryAi}(x)^2$	$- \frac{18^{2/3} 2^{1/3} 3^{2/3} x^4 \operatorname{BesselI}\left(\frac{2}{3}, \frac{2x^{3/2}}{3}\right)^2}{162(x^{3/2})^{4/3}} +$ $\frac{2 \cdot 18^{2/3} 2^{1/3} 3^{2/3} x \operatorname{BesselI}\left(\frac{2}{3}, \frac{2x^{3/2}}{3}\right)^2}{81(x^{3/2})^{4/3}} +$ $\frac{18^{2/3} 2^{1/3} 3^{2/3} x^4 \operatorname{BesselI}\left(\frac{5}{3}, \frac{2x^{3/2}}{3}\right)^2}{162(x^{3/2})^{4/3}} -$ $\frac{18^{2/3} 2^{1/3} 3^{2/3} \sqrt{x} \operatorname{BesselI}\left(\frac{1}{3}, \frac{2x^{3/2}}{3}\right) \operatorname{BesselI}\left(\frac{2}{3}, \frac{2x^{3/2}}{3}\right)}{81} -$ $\frac{18^{2/3} 2^{1/3} 3^{2/3} \operatorname{BesselI}\left(\frac{1}{3}, \frac{2x^{3/2}}{3}\right) \operatorname{BesselI}\left(\frac{5}{3}, \frac{2x^{3/2}}{3}\right)}{81} +$ $\frac{18^{2/3} 2^{1/3} 3^{2/3} x^2 \operatorname{BesselI}\left(\frac{4}{3}, \frac{2x^{3/2}}{3}\right) \operatorname{BesselI}\left(\frac{2}{3}, \frac{2x^{3/2}}{3}\right)}{81} +$ $\frac{3^{5/6} 2^{2/3} \Gamma\left(\frac{5}{6}\right) x^3 \operatorname{hypergeom}\left(\left[\frac{5}{6}, 1\right], \left[\frac{4}{3}, \frac{5}{3}, 2\right], \frac{4x^3}{9}\right)}{36\pi^{3/2}}$	$\operatorname{AiryAi}(x)^2 x - \operatorname{AiryAi}(1, x)^2$

Outline

- 1 Introduction
 - AI for Mathematics
 - Computer Algebra
 - Machine Learning and Computer Algebra
- 2 Learning Integration Algorithm Selection
 - Problem and Data Generation
 - ML Experiments
 - Can XAI Reveal Anything?
- 3 Learning a Good Variable Ordering
 - Cylindrical Algebraic Decomposition
 - Machine Learning the Variable Ordering
 - Two XAI Attempts

LIGs on Transformer Guards

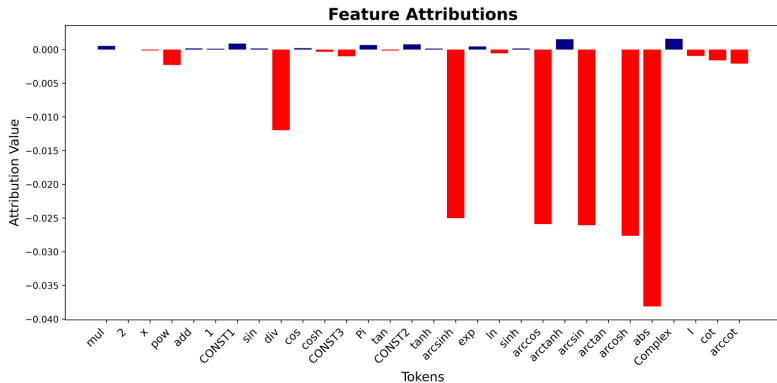
(Slide 34/50)

We experimented with Layered Integrated Gradients (LIGs) applied to our embedding layer to compute an attribution score for each input token.

One pattern spotted in these explanations is the high scores applied to the `abs` token for the Risch sub-algorithm classifier.

[CLS] mul INT+ CONST1 mul x pow abs cos mul INT+ 2 pow x INT+ 2 INT- 1

Upon inspection: this sub-algorithm cannot be applied when there is `abs` in the input, but the current Maple guard code did not check for this. An improvement offered to the CAS developers independent of whether they embed the ML or not.



LIGs on Transformer Guards

(Slide 34/50)

We experimented with Layered Integrated Gradients (LIGs) applied to our embedding layer to compute an attribution score for each input token.

One pattern spotted in these explanations is the high scores applied to the `abs` token for the Risch sub-algorithm classifier.

[CLS] mul INT+ CONST1 mul x pow abs cos mul INT+ 2 pow x INT+ 2 INT- 1

Upon inspection: this sub-algorithm cannot be applied when there is `abs` in the input, but the current Maple guard code did not check for this. An improvement offered to the CAS developers independent of whether they embed the ML or not.

Outline

- 1 Introduction
 - AI for Mathematics
 - Computer Algebra
 - Machine Learning and Computer Algebra
- 2 Learning Integration Algorithm Selection
 - Problem and Data Generation
 - ML Experiments
 - Can XAI Reveal Anything?
- 3 Learning a Good Variable Ordering
 - Cylindrical Algebraic Decomposition
 - Machine Learning the Variable Ordering
 - Two XAI Attempts

Outline

- 1 Introduction
 - AI for Mathematics
 - Computer Algebra
 - Machine Learning and Computer Algebra
- 2 Learning Integration Algorithm Selection
 - Problem and Data Generation
 - ML Experiments
 - Can XAI Reveal Anything?
- 3 Learning a Good Variable Ordering
 - Cylindrical Algebraic Decomposition
 - Machine Learning the Variable Ordering
 - Two XAI Attempts

Cylindrical Algebraic Decomposition

(Slide 35/50)

Recall the Real Quantifier Elimination Problem from the introduction. The main algebraic tool to solve that is the **Cylindrical Algebraic Decomposition (CAD)**. This:

- Decomposes the variable space according to the truth of the polynomial constraints involved (but is often much finer than needed). Thus can work with one sample point per cell.
- Provides algebraic formulae for each cell.
- Has cells arranged in cylinders, making projection easy.



G.E. Collins.

Quantifier elimination for real closed fields by cylindrical algebraic decomposition.

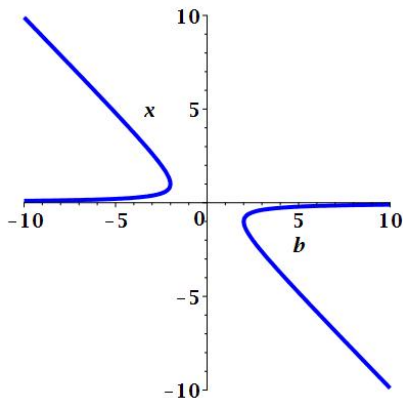
In Proc. 2nd GI Conf. Automata Theory and Formal Languages, pp. 134–183. Springer-Verlag, 1975.

QE via CAD Example

(Slide 36/50)

How to determine with CAD?

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



QE via CAD Example

(Slide 36/50)

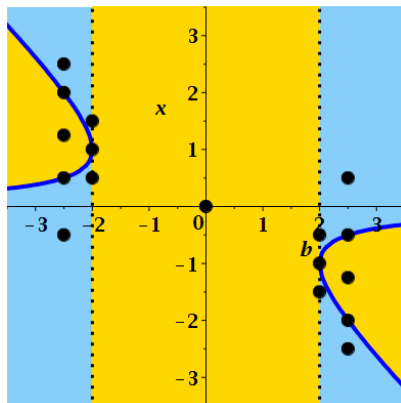
How to determine with CAD?

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

To solve we:

Build a sign-invariant CAD for

$$f = x^2 + bx + 1.$$



QE via CAD Example

(Slide 36/50)

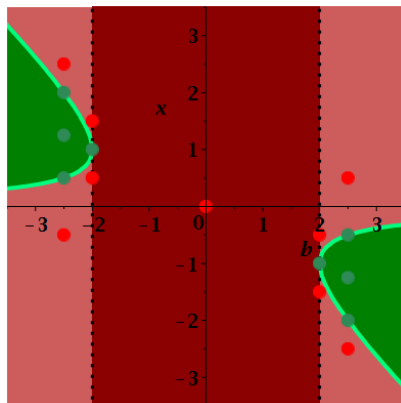
How to determine with CAD?

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

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

(Slide 36/50)

How to determine with CAD?

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

To solve we:

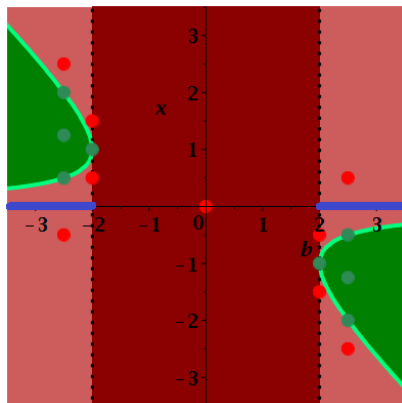
Build a sign-invariant CAD for
 $f = x^2 + bx + 1$.

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

(Slide 36/50)

How to determine with CAD?

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

To solve we:

Build a sign-invariant CAD for

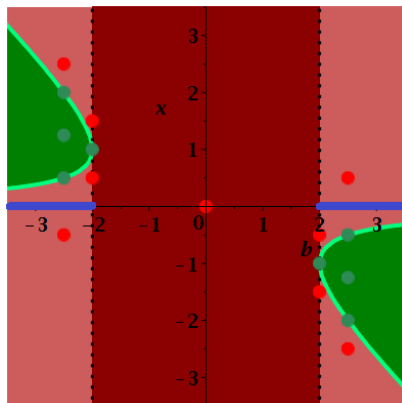
$$f = x^2 + bx + 1.$$

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

Take disjunction of projections of true cells:

$\Rightarrow$

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



QE via CAD in General

(Slide 37/50)

In general we can perform Real QE via CAD as follows.

- Eliminate existential quantifiers by projecting the true cells.
- Convert universal quantifiers to existential by using

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

and then proceeding with existential and negating the answer.

Recall our original example was $\forall x, x^2 + bx + 1 > 0$.

- The above leads us to study $\exists x, x^2 + bx + 1 \leq 0$.
- From previous slide this has solution $b \leq -2 \vee b \geq +2$.
- negate for solution to original problem: $-2 < b \wedge b < +2$
- Which is what we found numerically earlier.

Note: Projection and negation are both easy with cylindrical cells!

QE via CAD in General

(Slide 37/50)

In general we can perform Real QE via CAD as follows.

- Eliminate existential quantifiers by projecting the true cells.
- Convert universal quantifiers to existential by using

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

and then proceeding with existential and negating the answer.

Recall our original example was $\forall x, x^2 + bx + 1 > 0$.

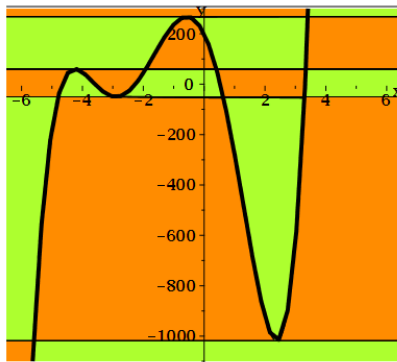
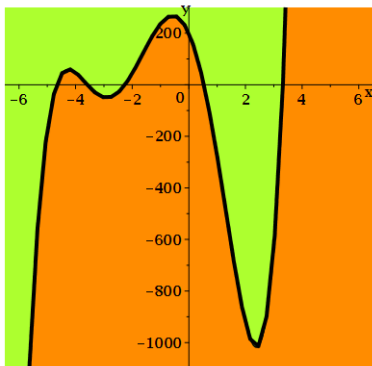
- The above leads us to study $\exists x, x^2 + bx + 1 \leq 0$.
- From previous slide this has solution $b \leq -2 \vee b \geq +2$.
- negate for solution to original problem: $-2 < b \wedge b < +2$
- Which is what we found numerically earlier.

Note: Projection and negation are both easy with cylindrical cells!

Variable Ordering for CAD

(Slide 38/50)

CAD requires a stated variable ordering; which affects decomposition size and computation time.



CAD Variable Ordering Choice

(Slide 39/50)

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 algebraic computations.

CASs tend to apply one of the former; and we focus on those.

Outline

- 1 Introduction
 - AI for Mathematics
 - Computer Algebra
 - Machine Learning and Computer Algebra
- 2 Learning Integration Algorithm Selection
 - Problem and Data Generation
 - ML Experiments
 - Can XAI Reveal Anything?
- 3 Learning a Good Variable Ordering
 - Cylindrical Algebraic Decomposition
 - Machine Learning the Variable Ordering
 - Two XAI Attempts

First Attempt of ML for CAD in 2014

(Slide 40/50)

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

Actually, the first use of Machine Learning to optimise any kind of computer algebra.



EPSRC ML4QE Project 2018–2020

(Slide 41/50)

Experimented with ML methodology to choose the ordering:

- 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

PhD Project 2021–2024

(Slide 42/50)

Tereso del Río:

- Identified and solved problems with the dataset in use (the SMT-LIB).
- Explored data augmentation methods for polynomials.
- Moved from a classification to a regression paradigm.



T. del Río and M. England.

Lessons on Datasets and Paradigms in Machine Learning for Symbolic Computation: A Case Study on CAD.

Mathematics in Computer Science, 18, article 17, Springer, 2024.

Preprint: <https://doi.org/10.48550/arXiv.2401.13343>

Generalisation Problems of ML for CAD

(Slide 43/50)

Recent work on the same problem using GNNs and RL:



Jia, Dong, Liu, Huang, Ma, Zhang.

Suggesting Variable Order for Cylindrical Algebraic Decomposition via Reinforcement Learning.

Proc. NIPS 2023. <https://openreview.net/forum?id=vNsdFwjPtL>

All the ML approaches to the problem to date suffer a drop-off in performance when models are applied to alternative datasets.

- Training on real world data is limited to large datasets (even with augmentation). The large datasets available are not representative of all problem domains.
- Training on random data does not generalise well to non-random data: the real world is not “*random*”.

Is ML optimisation of CAD limited to domain-specific use?

Inspiration

(Slide 44/50)



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 that it seemed to use some simple, but novel, strategies. Suggests that ML can reveal new mathematical truths about the algorithm, and direct future non-ML algorithm development.

How to automate such analysis?

Outline

- 1 Introduction
 - AI for Mathematics
 - Computer Algebra
 - Machine Learning and Computer Algebra
- 2 Learning Integration Algorithm Selection
 - Problem and Data Generation
 - ML Experiments
 - Can XAI Reveal Anything?
- 3 Learning a Good Variable Ordering
 - Cylindrical Algebraic Decomposition
 - Machine Learning the Variable Ordering
 - Two XAI Attempts

Post-hoc XAI for CAD Variable Ordering

(Slide 45/50)

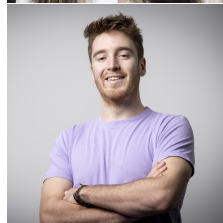
We applied the SHAP XAI tool to analyse the features used by ML classifiers to choose the CAD variable ordering.



L. Pickering, T. del Rio Almajano, M. England and K. Cohen.

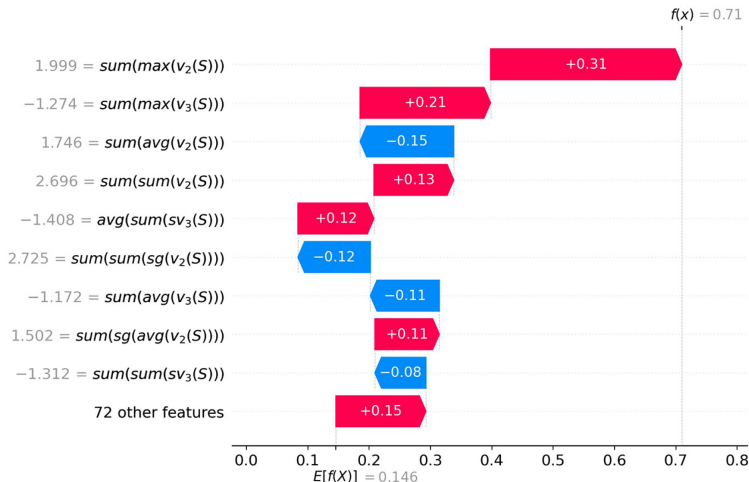
Explainable AI Insights for Symbolic Computation: A case study on selecting the variable ordering for cylindrical algebraic decomposition.

Journal of Symbolic Computation, **123**,
Article Number 102276, 2024.



Example SHAP Waterfall Plot

(Slide 45/50)



Using the SHAP output

(Slide 46/50)

- We aggregated the feature importance scores across the dataset for each of our ML models (KNN, MLP, SVM, RF).

This allowed us to see which models used which features — they did not fully agree on feature ranking.

- We had the models “vote” (used the Dowdall System) on features to create an overall feature ranking.

Some features identified as impactful had been known before; others were new.

- Used the top 3 to form a simple heuristic equivalent in complexity to that currently commonly used in CASs for making this choice.

It performs *almost* as well as the full ML models on our dataset.

Ante-hoc XAI for CAD Variable Ordering

(Slide 47/50)

The SHAP analysis was post-hoc XAI. Can we use something inherently interpretable?

We tried to design an interpretable model by forming a small neural network equivalent to the current human-designed heuristic, training, and then reinterpreting into human readable code.

We trained on polynomials whose exponents and coefficients were created randomly from distributions with the same mean and standard deviation as the SMT-LIB, while testing on the actual SMT-LIB data.



D. Florescu and M. England.

Constrained Neural Networks for Interpretable Heuristic Creation to Optimise Computer Algebra Systems.

Proc. ICMS 2024 (LNCS 14749), pp. 186-195,

https://doi.org/10.1007/978-3-031-64529-7_19

Generalisation over datasets!

(Slide 48/50)

Both experiments use XAI techniques to produce “*simple*” heuristics for CAD variable ordering and both have been shown to produce generalisability over datasets:

- In [FE24] we actually trained on random data and tested on SMT-LIB data.
- In [PdREC24] we trained and tested on the SMT-LIB. But subsequent experiments below on very large amounts of random data showed generalisability.



C. Chen, R.-J. Jing, C. Qian, Y. Yuan, and Y. Zhao.

A dataset for suggesting variable orderings for cylindrical algebraic decompositions.

In Proc. CASC 2024, (LNCS 14938), 100–119. Springer, 2024. https://doi.org/10.1007/978-3-031-69070-9_7

Summary of our Two XAI Experiments

(Slide 49/50)

Both experiments took a *human-designed* heuristic as a starting point and then used ML processes to produce another heuristic. The latter were not human-designed but are *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.

They could thus be employed by any CAD developer without having to do any ML. Further, they perform almost as well as the ML models themselves and generalise better.

We suggest that these could exemplify a new general methodology for computer algebra heuristic design. We have a new PhD student, Uzma Shafiq, exploring this.

Thanks for Listening

(Slide 50/50)



`Matthew.England@coventry.ac.uk`
`https://matthewengland.coventry.domains`