CoqApprox: Bilan et Perspectives

Érik Martin-Dorel

Post-doctorant TaMaDi
Équipe-Projet Marelle
Inria Sophia Antipolis
erik.martin-dorel@ens-lyon.org

Journées du projet ANR TaMaDi
18–19 Octobre 2012
Campus de Jussieu

Outline

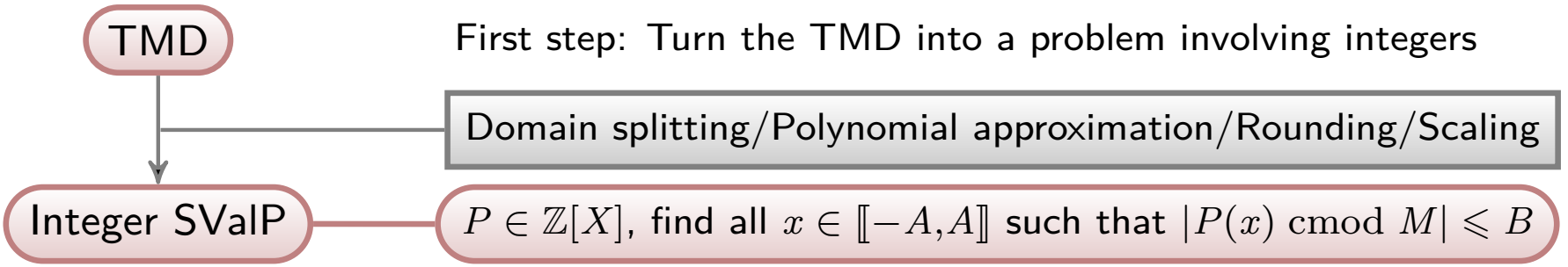
Outline

The SLZ algorithm

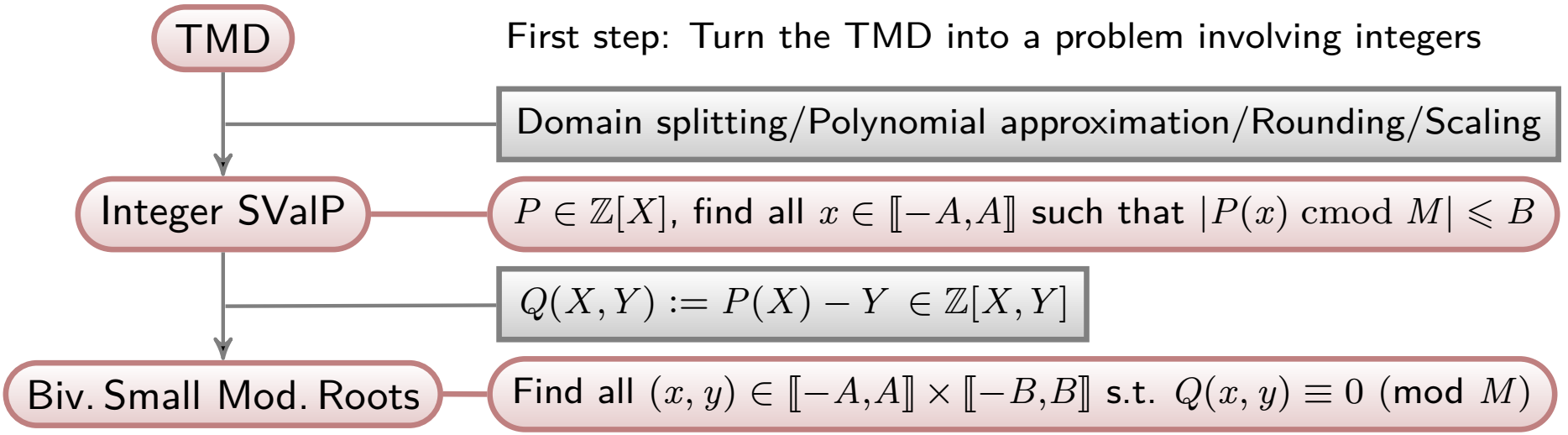
TMD

First step: Turn the TMD into a problem involving integers

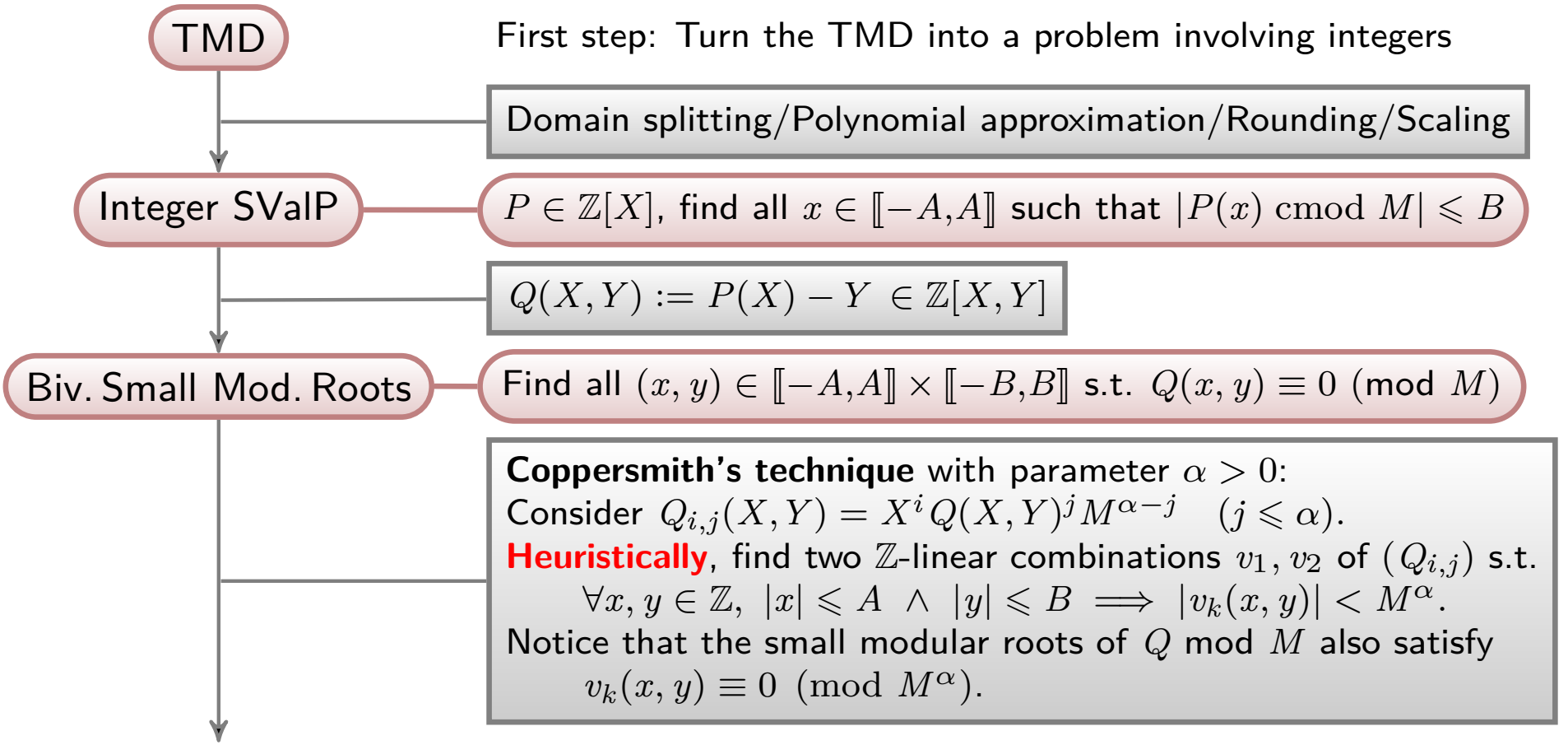
The SLZ algorithm



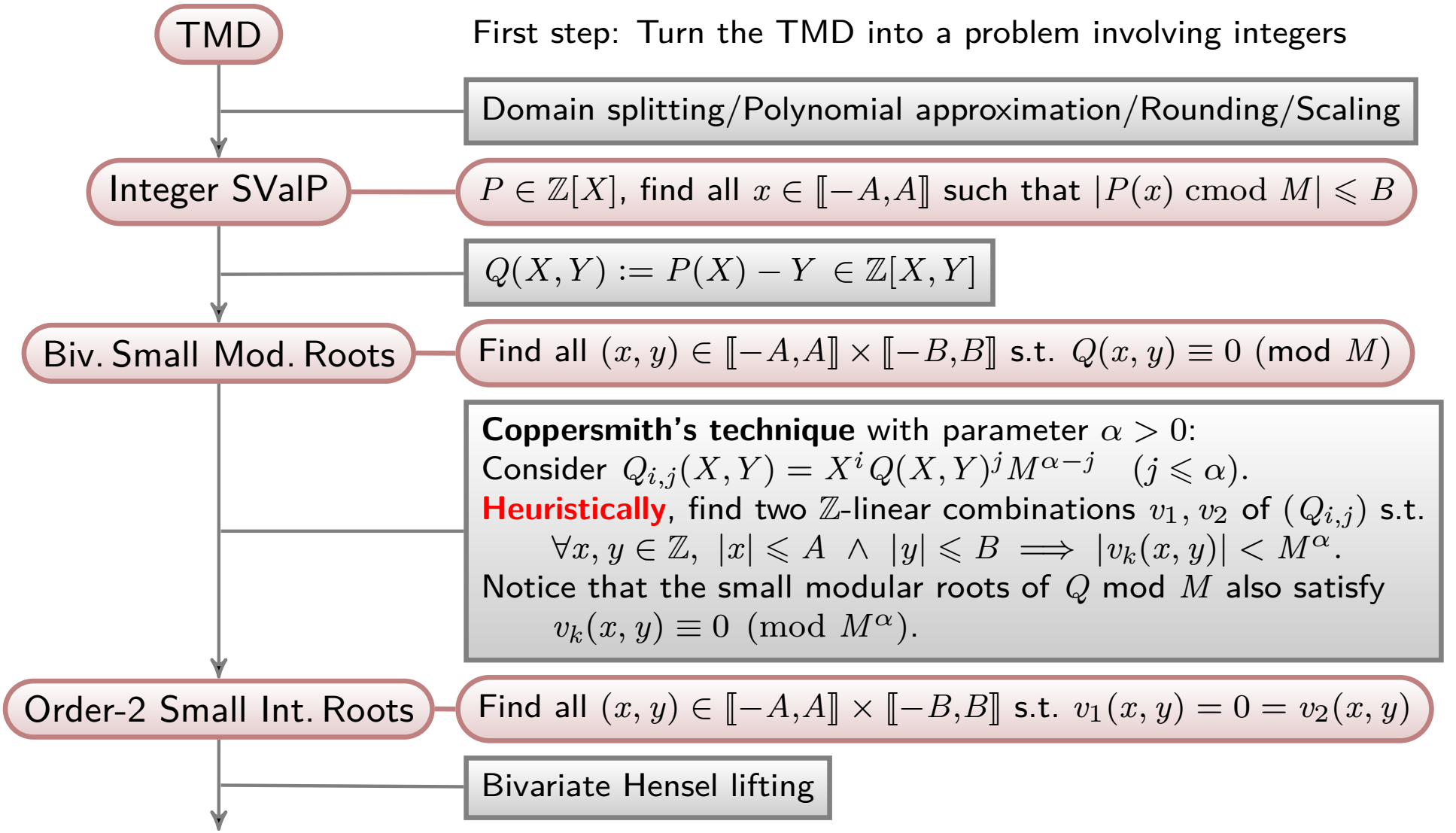
The SLZ algorithm



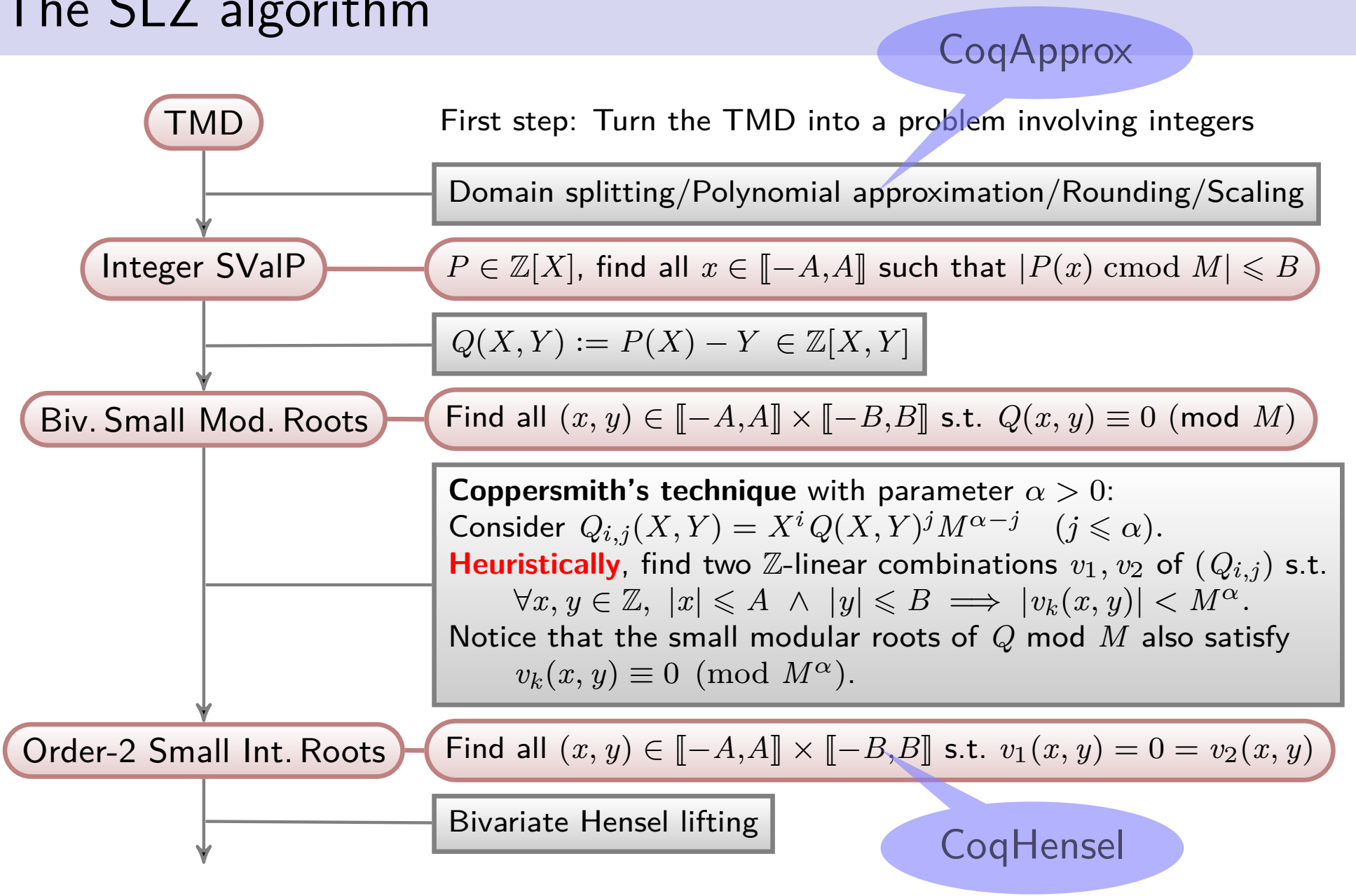
The SLZ algorithm



The SLZ algorithm



The SLZ algorithm



CoqApprox: A big team

Érik Martin-Dorel, Micaela Mayero, Ioana Paşca, Laurence Rideau, Laurent Théry

Nicolas Brisebarre, Mioara Joldeş, Jean-Michel Muller

The TaMaDi project of the French ANR

Goal: certified polynomial approximation for real functions

- Dedicated data structure: Rigorous Polynomial Approximation (RPA)
  - a pair $(P, \mathbf{\Delta})$ combining a polynomial and an interval that surrounds the approximation error
  - so-called Taylor Models (TMs) are a typical instance of RPAs

Goal: certified polynomial approximation for real functions

- Dedicated data structure: Rigorous Polynomial Approximation (RPA)
  - a pair $(P, \mathbf{\Delta})$ combining a polynomial and an interval that surrounds the approximation error
  - so-called Taylor Models (TMs) are a typical instance of RPAs
- Formal verification
  - ensure correctness of the RPA algorithms
  - ensure correct computation of RPAs

    ...
  - by using a proof assistant

Outline

Rigorous Polynomial Approximation

Definition

An order-n Rigorous Polynomial Approximation (RPA) for a function $f : D \subset \mathbb{R} \to \mathbb{R}$ over $\boldsymbol{I}$ is a pair $(P, \boldsymbol{\Delta})$ where P is a degree-n polynomial and $\boldsymbol{\Delta}$ is an interval, such that $\forall x \in \boldsymbol{I},\ f(x) - P(x) \in \boldsymbol{\Delta}$.

Rigorous Polynomial Approximation

Definition

An order-n Rigorous Polynomial Approximation (RPA) for a function $f : D \subset \mathbb{R} \to \mathbb{R}$ over $\boldsymbol{I}$ is a pair $(P, \boldsymbol{\Delta})$ where P is a degree-n polynomial and $\boldsymbol{\Delta}$ is an interval, such that $\forall x \in \boldsymbol{I},\ f(x) - P(x) \in \boldsymbol{\Delta}$. ▸ TM_exp_correct

Various possible instances of RPAs, depending on the polynomial basis and on the algorithms that are used:

Taylor Models: truncated Taylor series, naturally expressed in Taylor basis

Chebyshev Models: Chebyshev interpolants / truncated Chebyshev series

Rigorous Polynomial Approximation

Definition

An order-n Rigorous Polynomial Approximation (RPA) for a function $f : D \subset \mathbb{R} \to \mathbb{R}$ over $\boldsymbol{I}$ is a pair $(P, \boldsymbol{\Delta})$ where P is a degree-n polynomial and $\boldsymbol{\Delta}$ is an interval, such that $\forall x \in \boldsymbol{I},\ f(x) - P(x) \in \boldsymbol{\Delta}$.

Various possible instances of RPAs, depending on the polynomial basis and on the algorithms that are used:

Taylor Models: truncated Taylor series, naturally expressed in Taylor basis

Chebyshev Models: Chebyshev interpolants / truncated Chebyshev series

Taylor Models in CoqApprox

As regards $\boldsymbol{\Delta}$: interval remainder with floating-point bounds;
As regards P: small interval coefficients with floating-point bounds
$\Longrightarrow$ rounding errors are directly handled by the interval arithmetic

Taylor-Lagrange Remainder

Theorem (Taylor-Lagrange)

If f is $n+1$ times derivable on $\boldsymbol{I}$, then $\forall x \in \boldsymbol{I}$, $\exists \xi$ between x_0 and x s.t.:

$$f(x) = \underbrace{\left(\sum_{i=0}^{n} \frac{f^{(i)}(x_0)}{i!} (x - x_0)^i \right)}_{\textit{Taylor expansion}} + \underbrace{\frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)^{n+1}}_{\Delta(x,\xi)}.$$

Outline

For P: Compute interval enclosures of $\frac{f^{(i)}(x_0)}{i!}$, $i = 0, \ldots, n$.

For $\boldsymbol{\Delta}$: Compute enclosure of $\Delta(x, \xi)$:

Compute enclosure of $\frac{f^{(n+1)}(\xi)}{(n+1)!}$ and deduce $\boldsymbol{\Delta} := \frac{f^{(n+1)}(\boldsymbol{I})}{(n+1)!}(\boldsymbol{I} - x_0)^{n+1}$

Taylor-Lagrange Remainder

Theorem (Taylor-Lagrange)

If f is $n+1$ times derivable on $\boldsymbol{I}$, then $\forall x \in \boldsymbol{I}$, $\exists \xi$ between x_0 and x s.t.:

$$f(x) = \underbrace{\left(\sum_{i=0}^{n} \frac{f^{(i)}(x_0)}{i!} (x - x_0)^i \right)}_{\textit{Taylor expansion}} + \underbrace{\frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)^{n+1}}_{\Delta(x,\xi)}.$$

Outline

For P: Compute interval enclosures of $\frac{f^{(i)}(x_0)}{i!}$, $i = 0, \ldots, n$.

For $\boldsymbol{\Delta}$: Compute enclosure of $\Delta(x, \xi)$:

Compute enclosure of $\frac{f^{(n+1)}(\xi)}{(n+1)!}$ and deduce $\boldsymbol{\Delta} := \frac{f^{(n+1)}(\boldsymbol{I})}{(n+1)!}(\boldsymbol{I} - x_0)^{n+1}$

Composite functions $\Rightarrow$ enclosure for $\boldsymbol{\Delta}$ can be largely overestimated

Methodology of Taylor Models

Define arithmetic operations on Taylor Models:

- $\mathtt{TM}_{\mathtt{add}}$, $\mathtt{TM}_{\mathtt{mul}}$, $\mathtt{TM}_{\mathtt{comp}}$, and $\mathtt{TM}_{\mathtt{div}}$
- E.g., $\mathtt{TM}_{\mathtt{add}} : \Big((P_1, \mathbf{\Delta_1}),\ (P_2, \mathbf{\Delta_2})\Big) \mapsto (P_1 + P_2, \mathbf{\Delta_1} + \mathbf{\Delta_2})$.

A two-fold approach:

- Apply these operations recursively on the structure of the function
- Use Taylor-Lagrange remainder for atoms (i.e., for base functions)

Methodology of Taylor Models

Define arithmetic operations on Taylor Models:

- $\mathtt{TM}_{\mathtt{add}}$, $\mathtt{TM}_{\mathtt{mul}}$, $\mathtt{TM}_{\mathtt{comp}}$, and $\mathtt{TM}_{\mathtt{div}}$
- E.g., $\mathtt{TM}_{\mathtt{add}} : \Big((P_1, \mathbf{\Delta_1}),\ (P_2, \mathbf{\Delta_2})\Big) \mapsto (P_1 + P_2, \mathbf{\Delta_1} + \mathbf{\Delta_2})$.

A two-fold approach:

- Apply these operations recursively on the structure of the function
- Use Taylor-Lagrange remainder for atoms (i.e., for base functions)

$\Rightarrow$ Need to consider a relevant class for base functions, so that:

- We can easily compute their successive derivatives
- The interval remainder computed for these atoms is thin enough

D-finite functions (a.k.a. holonomic functions)

Definition

A D-finite function is a solution of a homogeneous linear ordinary differential equation with polynomial coefficients:
$a_r(x)y^{(r)}(x) + \cdots + a_1(x)y'(x) + a_0(x)y(x) = 0$, for given $a_k \in \mathbb{K}[X]$.

Property

The Taylor coefficients of these functions satisfy a *linear recurrence with polynomial coefficients*

D-finite functions (a.k.a. holonomic functions)

Definition

A D-finite function is a solution of a homogeneous linear ordinary differential equation with polynomial coefficients:
$a_r(x)y^{(r)}(x) + \cdots + a_1(x)y'(x) + a_0(x)y(x) = 0$, for given $a_k \in \mathbb{K}[X]$.

Property

The Taylor coefficients of these functions satisfy a *linear recurrence with polynomial coefficients* $\rightarrow$ fast numerical computation of the coefficients

Example (the exponential function)

The Taylor coefficients of $\exp$ at x_0 satisfy the recurrence
$\forall n \in \mathbb{N},\ (n+1)u_{n+1} = u_n$, with $u_0 = \exp(x_0)$ as an initial condition.

$\ln$, $\sin$, $\arcsin$, $\sinh$, $\operatorname{arcsinh}$, $\arctan$, $\operatorname{arctanh}$... are D-finite; $\tan$ is not

The Coq proof assistant

We use Coq for

- programming
  - strongly typed functional language
  - computation
- proving
  - use higher order logic
  - build proofs interactively
  - program automatic tactics
  - check proofs

Computing within the Coq proof assistant

Coq comes with a primitive notion of computation, called reduction.

Three main reduction tactics are available:

1984: `compute`: reduction machine (inside the kernel)

2004: `vm_compute`: virtual machine (byte-code)

2011: `native_compute`: compilation (native-code)

Several levels of trust:

method	trust	speed
`compute`	+++	+
`vm_compute`	++	++
`native_compute`	+	+++

Formally verified computation: CoqInterval

- Abstract interface for intervals
- Instantiation to intervals with floating-point bounds
- Formal verification with respect to the Reals library

for $x, y : \mathbb{R}$

and $\boldsymbol{X}, \boldsymbol{Y} : \mathbb{IR}$

$$x \in \boldsymbol{X} \ \wedge \ y \in \boldsymbol{Y} \implies x + y \in \boldsymbol{X} + \boldsymbol{Y}$$

$$x \in \boldsymbol{X} \implies \exp(x) \in \mathbf{exp}(\boldsymbol{X})$$

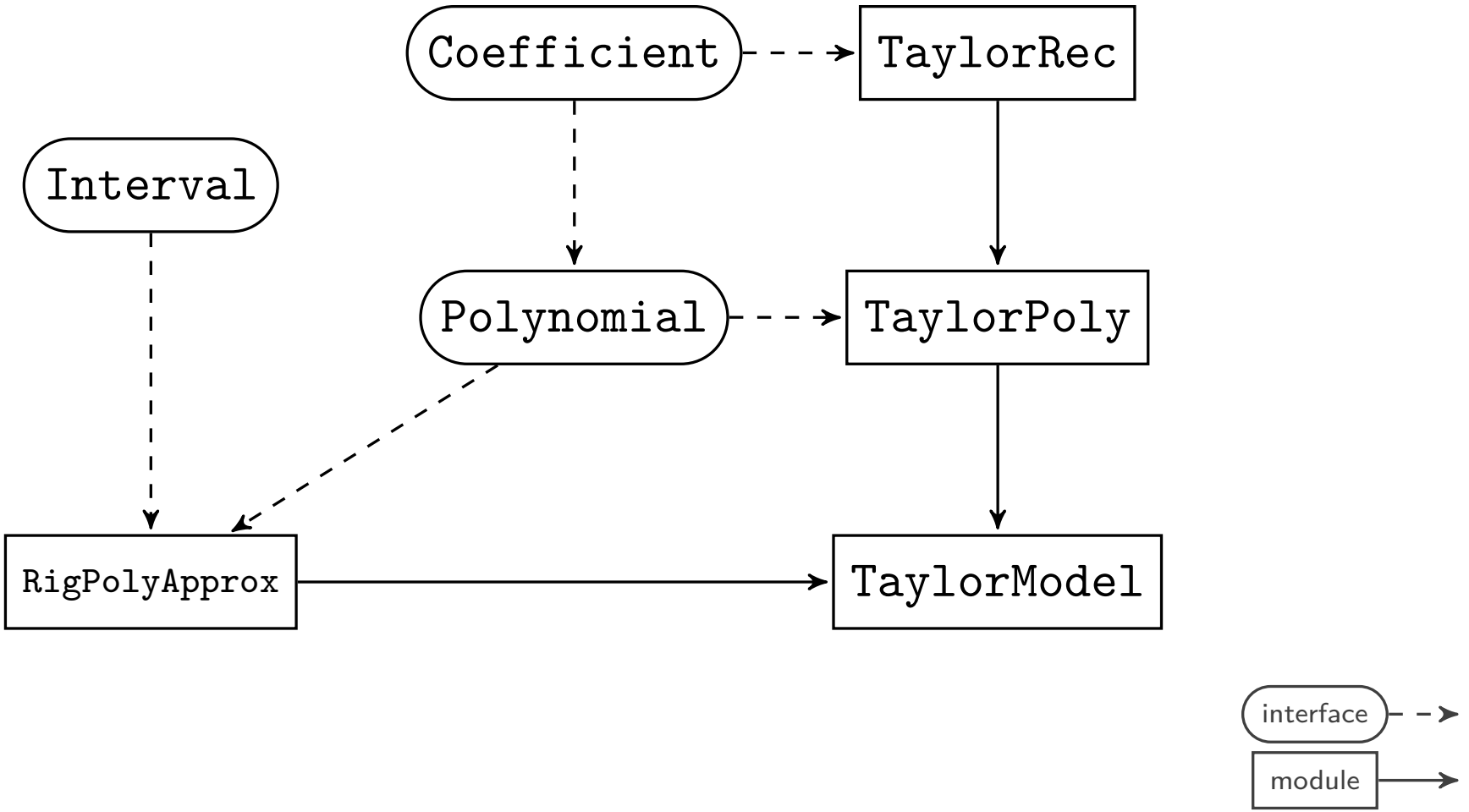
Implementation of Taylor Models in Coq

Focus on being generic:

- a Taylor Model is an instance of a Rigorous Polynomial Approximation, i.e., a pair $(P, \mathbf{\Delta})$
- generic with respect to
  - the type of coefficients of polynomial P,
  - the type of P and the implementation of related operations
  - the type of interval $\mathbf{\Delta}$

Prove correctness with respect to the standard Reals library

A generic implementation of TMs: modular hierarchy



Comparison with a dedicated tool implemented in C

Sollya [S.Chevillard, M.Joldeș, C.Lauter]

- written in C
- based on the MPFI library
- contains an implementation of univariate Taylor Models
- in an imperative-programming framework
- polynomials as arrays of coefficients

CoqApprox

- formalized in Coq
- based on the CoqInterval library
- implements Taylor Models using a similar algorithm
- in a functional-programming framework
- polynomials as lists of coefficients (linear access time)

Coq is around 10 times slower than Sollya! It's very good!

Some benchmarks for base functions

	Timing		Approximation error		
	Coq	Sollya	Coq	Sollya	Mathematical
$f = \exp$ prec=1000, deg=70 $\boldsymbol{I}=[127/128, 1]$	0.716s	0.093s	1.80×2^{-906}	1.79×2^{-906}	1.79×2^{-906}
$f = \sin$ prec=1000, deg=70 $\boldsymbol{I}=[127/128, 1]$	2.636s	0.088s	1.45×2^{-908}	1.44×2^{-908}	1.44×2^{-908}
$f = \arctan$ prec=1000, deg=118 $\boldsymbol{I}=[127/128, 1]$	2.969s	0.420s	1.71×2^{-913}	1.30×2^{-967}	1.07×2^{-1001}

- with Coq v8.3pl4 using `vm_compute`,
- and Sollya v3.0 using `taylorform()`, along with `supnorm()` for last column.

Some benchmarks for composite functions

	Timing		Approximation error		
	Coq	Sollya	Coq	Sollya	Mathematical
$f = \exp \times \sin$ prec=400, deg=20 $\boldsymbol{I}$=[127/128, 1]	0.812s	0.013s	1.36×2^{-222}	1.36×2^{-222}	1.36×2^{-222}
$f = \exp \times \sin$ prec=400, deg=40 $\boldsymbol{I}$=[127/128, 1]	1.736s	0.040s	1.01×2^{-397}	1.53×2^{-397}	1.06×2^{-402}
$f = \exp \circ \sin$ prec=400, deg=20 $\boldsymbol{I}$=[127/128, 1]	7.165s	0.011s	1.56×2^{-192}	1.83×2^{-192}	1.56×2^{-192}
$f = \exp \circ \sin$ prec=400, deg=40 $\boldsymbol{I}$=[127/128, 1]	52.687s	0.065s	1.88×2^{-385}	1.38×2^{-384}	1.88×2^{-385}

- with Coq v8.3pl4 using `vm_compute`,
- and Sollya v3.0 using `taylorform()`, along with `supnorm()` for last column.

Overview of the correctness lemma for exp

```
Lemma TM_exp_correct :
  forall (X0 X : I.type) (n : nat),
  I.subset_ (I.convert X0) (I.convert X) ->
  ( exists t : ExtendedR, contains (I.convert X0) t ) ->
  validTM (I.convert X0) (I.convert X) (TM_exp X0 X n) Xexp.
```

◂ informal definition of validTM ▸ complete definition of validTM

Idea of the proof of TMs for the exponential

$\mathtt{TM}_{\mathtt{exp}}(\boldsymbol{x_0}, \boldsymbol{I}, n) := (\boldsymbol{a_0} :: \ldots :: \boldsymbol{a_n}, \boldsymbol{\Delta})$ with

$$\boldsymbol{x_0} \subset \boldsymbol{I}, \quad \boldsymbol{a_0} = \mathbf{exp}(\boldsymbol{x_0}), \quad \boldsymbol{a_{n+1}} = \frac{\boldsymbol{a_n}}{n+1}, \quad \boldsymbol{\Delta} = \frac{\mathbf{exp}(\boldsymbol{I})}{(n+1)!} \times (\boldsymbol{I} - \boldsymbol{x_0})^{n+1}$$

Idea of the proof of TMs for the exponential

$\mathtt{TM}_{\mathtt{exp}}(\boldsymbol{x_0}, \boldsymbol{I}, n) := (\boldsymbol{a_0} :: \ldots :: \boldsymbol{a_n}, \boldsymbol{\Delta})$ with

$$\boldsymbol{x_0} \subset \boldsymbol{I}, \quad \boldsymbol{a_0} = \mathbf{exp}(\boldsymbol{x_0}), \quad \boldsymbol{a_{n+1}} = \frac{\boldsymbol{a_n}}{n+1}, \quad \boldsymbol{\Delta} = \frac{\mathbf{exp}(\boldsymbol{I})}{(n+1)!} \times (\boldsymbol{I} - \boldsymbol{x_0})^{n+1}$$

We want to show that $\mathtt{TM}_{\mathtt{exp}}(\boldsymbol{x_0}, \boldsymbol{I}, n)$ is a valid TM for $\exp$.

- $\boldsymbol{x_0} \subset \boldsymbol{I}$
- $0 \in \boldsymbol{\Delta}$
- $\forall \xi_0 \in \boldsymbol{x_0}, \exists \alpha_0 \in \boldsymbol{a_0}, \ldots, \alpha_n \in \boldsymbol{a_n},$

  $$\forall x \in \boldsymbol{I},\ \exp(x) - \sum_{i=0}^{n} \alpha_i \left(x - \xi_0\right)^i \in \boldsymbol{\Delta}$$

Idea of the proof of TMs for the exponential

$\mathtt{TM}_{\mathtt{exp}}(\boldsymbol{x_0}, \boldsymbol{I}, n) := (\boldsymbol{a_0} :: \ldots :: \boldsymbol{a_n}, \boldsymbol{\Delta})$ with

$$\boldsymbol{x_0} \subset \boldsymbol{I}, \quad \boldsymbol{a_0} = \mathbf{exp}(\boldsymbol{x_0}), \quad \boldsymbol{a_{n+1}} = \frac{\boldsymbol{a_n}}{n+1}, \quad \boldsymbol{\Delta} = \frac{\mathbf{exp}(\boldsymbol{I})}{(n+1)!} \times (\boldsymbol{I} - \boldsymbol{x_0})^{n+1}$$

We want to show that $\mathtt{TM}_{\mathtt{exp}}(\boldsymbol{x_0}, \boldsymbol{I}, n)$ is a valid TM for $\exp$.

- $\boldsymbol{x_0} \subset \boldsymbol{I}$
- $0 \in \boldsymbol{\Delta}$
- $\forall \xi_0 \in \boldsymbol{x_0}, \exists \alpha_0 \in \boldsymbol{a_0}, \ldots, \alpha_n \in \boldsymbol{a_n},$

  $$\forall x \in \boldsymbol{I},\ \exp(x) - \sum_{i=0}^{n} \alpha_i \left(x - \xi_0\right)^i \in \boldsymbol{\Delta}$$

$\exists \alpha_i = \dfrac{\exp(\xi_0)}{i!} \in \boldsymbol{a_i}$ such that for all $x \in \boldsymbol{I}$,

$$\exp(x) - \sum_{i=0}^{n} \frac{\exp(\xi_0)}{i!} \left(x - \xi_0\right)^i = \frac{\exp(\xi)}{(n+1)!} \times \left(x - \xi_0\right)^{n+1} \text{ for some } \xi \in \boldsymbol{I}.$$

Generalization to an arbitrary D-finite function f

Difficulties:

- Find minimal assumptions on the function f
  - the derivative is compatible with the recurrence relation
  - we have a compatible interval evaluator for f
- Provide the Taylor-Lagrange theorem for standard Reals

$\rightsquigarrow$ Generic proof for first-order and second-order recurrences.

Proofs for composite functions

Proof of the algorithm for each algebraic rule

- $\texttt{TM}_{\texttt{add}}$: straightforward
- $\texttt{TM}_{\texttt{mul}}$: rely on truncated multiplication of polynomials
- $\texttt{TM}_{\texttt{comp}}$: rely on $\texttt{TM}_{\texttt{mul}}$, $\texttt{TM}_{\texttt{add}}$ and TMs for constant functions
- $\texttt{TM}_{\texttt{div}}$: it's a TM for $f \times \left(\left(x \mapsto \frac{1}{x}\right) \circ g\right)$

Proof status

Fun/Op	Reals	CoqInterval	Implemented in CoqApprox	Proved in CoqApprox
cst	☒	☒	☒	☒
id	☒	☒	☒	☒
inv	☒	☒	☒	☒
sqrt	☒	☒	☒	☒
$\frac{1}{\sqrt{\cdot}}$	☒	☒	☒	☒
exp	☒	☒	☒	☒
sin	☒	☒	☒	☒
cos	☒	☒	☒	☒
arctan	☒	☒	☒	☐
ln	☒	☐	☒	☐
arcsin	☐	☐	☒	☐
arccos	☐	☐	☒	☐
$\mathtt{TM}_{\mathtt{add}}$			☒	☒
$\mathtt{TM}_{\mathtt{mul}}$			☒	☒
$\mathtt{TM}_{\mathtt{comp}}$			☒	☒
$\mathtt{TM}_{\mathtt{div}}$			☒	☒

Brief reminder of the Taylor-Lagrange theorem

Theorem (Taylor-Lagrange)

If f is $n+1$ times derivable on $\boldsymbol{I}$, then $\forall x \in \boldsymbol{I}$, $\exists \xi$ between x_0 and x s.t.:

$$f(x) = \underbrace{\left(\sum_{i=0}^{n} \frac{f^{(i)}(x_0)}{i!} (x - x_0)^i \right)}_{\textit{Taylor expansion}} + \underbrace{\frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)^{n+1}}_{\Delta(x,\xi)}.$$

- Strategy: Compute an interval enclosure $\boldsymbol{\Delta}$ of $\Delta(x, \xi)$ over $\boldsymbol{I}$

Brief reminder of the Taylor-Lagrange theorem

Theorem (Taylor-Lagrange)

If f is $n+1$ times derivable on $\boldsymbol{I}$, then $\forall x \in \boldsymbol{I}$, $\exists \xi$ between x_0 and x s.t.:

$$f(x) = \underbrace{\left(\sum_{i=0}^{n} \frac{f^{(i)}(x_0)}{i!} (x - x_0)^i \right)}_{\textit{Taylor expansion}} + \underbrace{\frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)^{n+1}}_{\Delta(x,\xi)}.$$

- Strategy: Compute an interval enclosure $\boldsymbol{\Delta}$ of $\Delta(x, \xi)$ over $\boldsymbol{I}$
- For a few functions, the $\boldsymbol{\Delta}$ so obtained can be largely overestimated
- E.g., for $f(x) = \frac{1}{x}$, we have $\boldsymbol{\Delta} = \frac{(-1)^{n+1}}{\boldsymbol{I}^{n+2}} (\boldsymbol{I} - \boldsymbol{x_0})^{n+1}$

Taylor Model for the inverse function: Proved algorithm

Algorithm 1: `TM_inv`

Input: Two intervals $\boldsymbol{x_0} \subset \boldsymbol{I}$, and $n \in \mathbb{N}$
Output: An order-n Taylor Model for $x \mapsto \frac{1}{x}$ around $\boldsymbol{x_0}$ over $\boldsymbol{I}$
$T :=$ `trec1 (inv_rec X0) (tinv X0) n` `// D-finite recurrence`
$L :=$ `trec1 (inv_rec I) (tinv I) n.+1` `// D-finite recurrence`
$\boldsymbol{\Gamma} :=$ `tnth L n.+1`
$\boldsymbol{\Delta} := \boldsymbol{\Gamma} \times (\boldsymbol{I} - \boldsymbol{x_0})^{n+1}$
return $(T, \boldsymbol{\Delta})$

Taylor Model for the inverse function: Optimized algorithm

Algorithm 2: `TM_inv` with tighter error bounds

Input: Two intervals $\boldsymbol{x_0} \subset \boldsymbol{I}$, and $n \in \mathbb{N}$
Output: An order-n Taylor Model for $x \mapsto \frac{1}{x}$ around $\boldsymbol{x_0}$ over $\boldsymbol{I}$

$T :=$ `trec1 (inv_rec X0) (tinv X0) n` // D-finite recurrence
$L :=$ `trec1 (inv_rec I) (tinv I) n.+1` // D-finite recurrence
$\boldsymbol{\Gamma} :=$ `tnth L n.+1`
if $\sup(\boldsymbol{\Gamma}) \leqslant 0$ **or** $\inf(\boldsymbol{\Gamma}) \geqslant 0$ **then**
  $\boldsymbol{a} := [\inf(\boldsymbol{I})]$; $\boldsymbol{b} := [\sup(\boldsymbol{I})]$
  $\boldsymbol{\Delta_a} := 1/\boldsymbol{a} - T(\boldsymbol{a} - \boldsymbol{x_0})$
  $\boldsymbol{\Delta_b} := 1/\boldsymbol{b} - T(\boldsymbol{b} - \boldsymbol{x_0})$
  $\boldsymbol{\Delta_{x_0}} := 1/\boldsymbol{x_0} - T(\boldsymbol{x_0} - \boldsymbol{x_0})$
  $\boldsymbol{\Delta} := \boldsymbol{\Delta_a} \vee \boldsymbol{\Delta_b} \vee \boldsymbol{\Delta_{x_0}}$
else
  $\boldsymbol{\Delta} := \boldsymbol{\Gamma} \times (\boldsymbol{I} - \boldsymbol{x_0})^{n+1}$
return $(T, \boldsymbol{\Delta})$

Implementation in Coq

- Generic implementation located in the "`Trem`" function.
- Impact on the error bounds for base functions as well as for division

	Error bounds		
	Taylor-Lagrange	Optimized algorithm	Sollya
$x \mapsto 1/x$ over $\boldsymbol{I} = [1, 4]$ with $n = 20$	1.218×2^{12}	1.438×2^{-16}	1.438×2^{-16}
$x \mapsto \sin x$ over $\boldsymbol{I} = [0, 1]$ with $n = 20$	1.445×2^{-87}	1.283×2^{-87}	1.283×2^{-87}
$x \mapsto \exp(1/\cos x)$ over $\boldsymbol{I} = [0, 1]$ with $n = 14$	1.016×2^{-4}	1.342×2^{-9}	1.431×2^{-9}

All these TMs are computed around $\boldsymbol{x_0} = [\mathrm{mid}(\boldsymbol{I})]$, in precision-200 IA.

Short-term future work

- Formally prove Algorithm 2 in a generic way
- First, prove the following result (Proposition 2.2.1 in Mioara Joldeș's thesis, adapted from Lemma 5.12 in Roland Zumkeller's thesis):

Proposition

Let f be a function defined over an interval $\boldsymbol{I} = [a, b]$; let $\xi_0 \in \boldsymbol{I}$ and let $n \geqslant 0$ be an integer. If the sign of $f^{(n+1)}$ is constant over $\boldsymbol{I}$, then the remainder between f and its Taylor expansion of degree n around ξ_0 is ***monotonic*** *on $[a, \xi_0]$ and on $[\xi_0, b]$.*

Functions missing from support libraries

Functions missing from the Reals library

- cannot provide a proof for the Taylor Model
- adding them is so far done in a case-by-case manner

→ find a generic way of adding a new function to Reals

→ e.g. by using a differential equation or a recurrence relation as definition

Functions missing from CoqInterval

- cannot provide an initial value for the Taylor Model

→ just implement the missing functions in CoqInterval

→ may use some extra techniques, such as fixed point theorems

Outline

Brief conclusion

CoqApprox: a modular formalization of rigorous polynomial approximation using Taylor Models in the Coq proof assistant

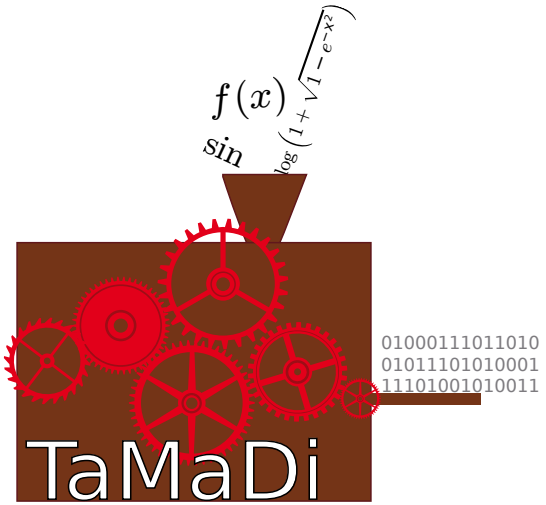
- with a generic approach involving D-finite functions
- taking advantage of the CoqInterval library for interval arithmetic

$\rightarrow$ ability to efficiently compute some formally verified TMs in Coq

Perspectives for CoqApprox

- Add more functions ($\cosh$, $\tan$, etc.) and prove them
- Optimize the multiplication algorithm for Taylor Models
- Implement Chebyshev Models $\rightsquigarrow$ tighter remainders
- Combine TMs with some polynomial global optimization technique
  - Sums of Squares?
  - Bernstein polynomials?
- Consider the possible generalization to the multivariate case
- Investigate ways to ease the definition of RPAs from the ODE
- Investigate the possible use of an "affine arithmetic" module
- Investigate RPAs based on other families of orthogonal polynomials
- Consider alternative techniques for verifying error bounds
  - fixed-point theorems?
  - majorant series?

End of the Talk



Thank you for your attention!

The TaMaDi project homepage:
http://tamadi.gforge.inria.fr/

Proving Taylor Models in Coq

◂ Return

Definition

Let $f : \boldsymbol{I} \to \mathbb{R}$ be a function, $\boldsymbol{x_0}$ be a small interval around an expansion point x_0. Let T be a polynomial with interval coefficients $\boldsymbol{a_0}, \ldots, \boldsymbol{a_n}$ and $\boldsymbol{\Delta}$ an interval. We say that $(T, \boldsymbol{\Delta})$ is a Taylor Model of f at $\boldsymbol{x_0}$ on $\boldsymbol{I}$ when

$$\begin{cases} \boldsymbol{x_0} \subseteq \boldsymbol{I}, \\ 0 \in \boldsymbol{\Delta}, \\ \forall \xi_0 \in \boldsymbol{x_0}, \exists \alpha_0 \in \boldsymbol{a_0}, \ldots, \alpha_n \in \boldsymbol{a_n}, \forall x \in \boldsymbol{I}, \quad f(x) - \sum_{i=0}^{n} \alpha_i \left(x - \xi_0\right)^i \in \boldsymbol{\Delta}. \end{cases}$$