Analytic Functions and Ordinary Differential Equations in Exact Real Arithmetic

Holger Thies [1]

The University of Tokyo

May 12, 2016

[1] Joint work with Akitoshi Kawamura and Florian Steinberg

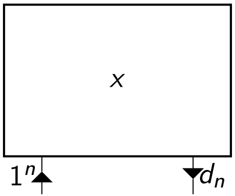
Computable Real Numbers

Computable Real Numbers

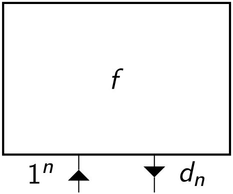
A real number is called computable if it can be approximated up to any desired precision.

Computable Real Numbers

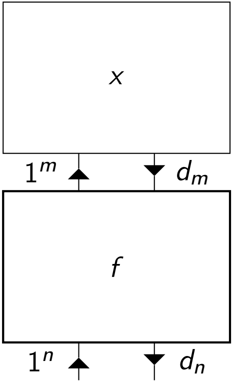
A real number is called computable if it can be approximated up to any desired precision.



Computable Real Functions



Computable Real Functions

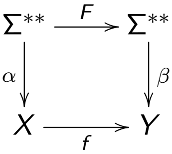


Representations

A representation for a set X is a partial surjective function $\rho : \Sigma^{**} \to X$.

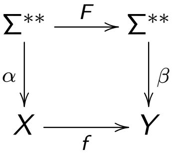
Representations

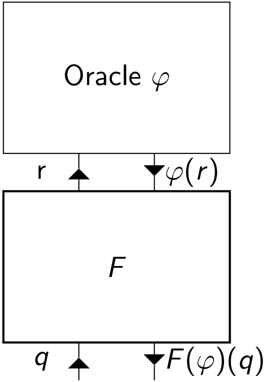
A representation for a set X is a partial surjective function $\rho : \Sigma^{**} \to X$.



Representations

A representation for a set X is a partial surjective function $\rho : \Sigma^{**} \to X$.





The iRRAM Framework

- The computational model can be implemented

The iRRAM Framework

- The computational model can be implemented
- `iRRAM` is a `C++` framework for exact real arithmetic written by Norbert Müller (University of Trier)

The iRRAM Framework

- The computational model can be implemented
- `iRRAM` is a `C++` framework for exact real arithmetic written by Norbert Müller (University of Trier)
- `REAL` as data type that can perform arithmetic operations on `REAL` without introducing rounding errors

The iRRAM Framework

- The computational model can be implemented
- iRRAM is a C++ framework for exact real arithmetic written by Norbert Müller (University of Trier)
- REAL as data type that can perform arithmetic operations on REAL without introducing rounding errors
- In the end, the user can output an approximation to the result of the computation with any desired precision

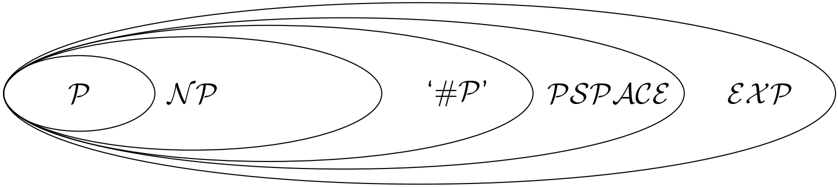
The iRRAM Framework

- The computational model can be implemented
- `iRRAM` is a `C++` framework for exact real arithmetic written by Norbert Müller (University of Trier)
- `REAL` as data type that can perform arithmetic operations on `REAL` without introducing rounding errors
- In the end, the user can output an approximation to the result of the computation with any desired precision
- Many predefined functions already exist (sine,...)

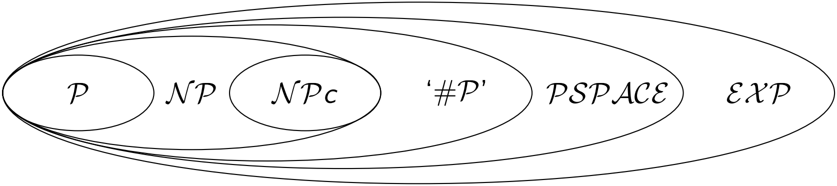
The iRRAM Framework

- The computational model can be implemented
- `iRRAM` is a `C++` framework for exact real arithmetic written by Norbert Müller (University of Trier)
- `REAL` as data type that can perform arithmetic operations on `REAL` without introducing rounding errors
- In the end, the user can output an approximation to the result of the computation with any desired precision
- Many predefined functions already exist (sine,...)
- New real numbers and functions can be defined (limit operator)

Complexity Theory



Complexity Theory



Data-types for functions

We want to implement a data-type for functions and perform operations on this type.

Data-types for functions

We want to implement a data-type for functions and perform operations on this type.

Fact

For general polynomial time computable functions, many important operators have been shown to be computationally hard. For example

- *Polynomial time computable functions may have noncomputable derivatives. (Ko 1983)*
- *Parametric maximization is NP-hard. (Ko/Friedman (1982))*
- *Integration is #P-hard. (Friedman (1984))*

Analytic Function

An analytic function is a function locally given by a complex power series.

Definition (Analytic Function)

$f : D \to \mathbb{C}$, $D \subseteq \mathbb{C}$ is analytic if for any $x_0 \in D$ the Taylor-series

$$T(x) := \sum_{n=0}^{\infty} a_n (x - x_0)^n$$

converges to $f(x)$ for x in a neighborhood of x_0.

Some non-uniform results

$$a_m = \frac{f^{(m)}(x_0)}{m!}, \quad f(x) = \sum_{m=0}^{\infty} a_m (x - x_0)^k \quad \text{for } x \in B(x_0, R)$$

Theorem (Pour-El, Richards, Ko, Friedman, Müller (1987/1989))

f is (polytime) computable iff $(a_m)_{m \in \mathbb{N}}$ is.

Some non-uniform results

$$a_m = \frac{f^{(m)}(x_0)}{m!}, \ f(x) = \sum_{m=0}^{\infty} a_m (x - x_0)^k \quad \text{for } x \in B(x_0, R)$$

Theorem (Pour-El, Richards, Ko, Friedman, Müller (1987/1989))

f is (polytime) computable iff $(a_m)_{m \in \mathbb{N}}$ is.

From that polynomial time computability of the derivative and the anti-derivative of a function follows immediately.

Some non-uniform results

$$a_m = \frac{f^{(m)}(0)}{m!}, \; f(x) = \sum_{m=0}^{\infty} a_m x^k \quad \text{for } x \in B(0,R)$$

Theorem (Müller (1995))

- *The operator* $f \to (a_m)_{m\in\mathbb{N}}$ *is not computable.*
- *The evaluation operator* $((a_m)_{m\in\mathbb{N}}, x) \to f(x)$ *is not computable.*

A practical representation for power series

Lemma

Let $f : \overline{B}(0,1) \to \mathbb{R}$ *be analytic and* $(a_n)_{n\in\mathbb{N}}$ *its power series around* 0.
Then there exists an $k, A \in \mathbb{N}$ *such that*

1. $\sqrt[k]{2}$ *is a lower bound on the radius of convergence*
2. $|a_n| \leq A \cdot 2^{-\frac{n}{k}}$

A practical representation for power series

Lemma

Let $f : \overline{B}(0,1) \to \mathbb{R}$ be analytic and $(a_n)_{n\in\mathbb{N}}$ its power series around 0.
Then there exists an $k, A \in \mathbb{N}$ such that

1. $\sqrt[k]{2}$ *is a lower bound on the radius of convergence*
2. $|a_n| \leq A \cdot 2^{-\frac{n}{k}}$

A and k can be used to make a tail estimate on

$$\left| \sum_{n \geq N} a_n z^n \right|$$

Analytic Functions and Computational Complexity

Theorem (Kawamura, Rösnick, Müller, Ziegler (2013))

The following operators are computable in time polynomial in $n + k + \log(A)$, *where* 2^{-n} *is the output precision.*

1. *evaluation*
2. *addition and multiplication*
3. *differentiation and anti-differentiation*
4. *parametric maximization*

Ordinary Differential Equations

We are further interested in the following problem:
Consider Systems of Ordinary Differential Equations of the form

$$\begin{aligned} \dot{y}_v(t) &= F_v(t, y_1(t), \ldots, y_t(t)) \\ y_v(w_0) &= y_0 \end{aligned}$$

for $v = 1, \ldots, d$ and $F_v : \mathbb{R}^{d+1} \to \mathbb{R}$.

Complexity of Ordinary Differential Equations

Theorem (Kawamura, 2010)

Consider the IVP

$$y'(t) = f(t, y(t)) \quad ; \quad y(0) = 0.$$

There exists functions $f : [0,1] \times [-1,1] \to \mathbb{R}$ *and* $y : [0,1] \to [-1,1]$ *as above such that*

Complexity of Ordinary Differential Equations

Theorem (Kawamura, 2010)

Consider the IVP

$$y'(t) = f(t, y(t)) \quad ; \quad y(0) = 0.$$

There exists functions $f : [0,1] \times [-1,1] \to \mathbb{R}$ *and* $y : [0,1] \to [-1,1]$ *as above such that*

1. f *is Lipschitz-continuous and polynomial time computable*
2. y *is PSPACE-hard.*

Ordinary Differential Equations

We are further interested in the following problem:
Consider Systems of Ordinary Differential Equations of the form

$$\begin{aligned} \dot{y}_v(t) &= F_v(t, y_1(t), \dots, y_t(t)) \\ y_v(0) &= 0 \end{aligned}$$

for $v = 1, \dots, d$ and $F_v : \mathbb{R}^{d+1} \to \mathbb{R}$.
This is hard in the general case.

Ordinary Differential Equations

We are further interested in the following problem:
Consider Systems of Ordinary Differential Equations of the form

$$\begin{aligned} \dot{y}_v(t) &= F_v(t, y_1(t), \ldots, y_t(t)) \\ y_v(0) &= 0 \end{aligned}$$

for $v = 1, \ldots, d$ and $F_v : \mathbb{R}^{d+1} \to \mathbb{R}$.
This is hard in the general case.
If all F_v are analytic

- the solutions y_v will again be analytic
- the coefficients of the power series can be computed in polynomial time if the F_v are polynomial time computable.

Multi-Dimensional Functions

Consider functions $f : \mathbb{R}^d \to \mathbb{R}$ analytic in a neighborhood of $0 \in \mathbb{C}^d$.

Multi-Dimensional Functions

Consider functions $f : \mathbb{R}^d \to \mathbb{R}$ analytic in a neighborhood of $0 \in \mathbb{C}^d$. Their power series is given by

$$\sum_{n_1,\ldots,n_d \in \mathbb{N}} a_{n_1,\ldots,n_d} x_1^{n_1} \ldots x_d^{n_d}$$

Multi-Dimensional Functions

Consider functions $f : \mathbb{R}^d \to \mathbb{R}$ analytic in a neighborhood of $0 \in \mathbb{C}^d$. Their power series is given by

$$\sum_{n_1,\dots,n_d \in \mathbb{N}} a_{n_1,\dots,n_d} x_1^{n_1} \dots x_d^{n_d}$$

The beforementioned representation can be generalized.
The main idea is let

$$f(x_1,\dots,x_d) = \sum_{i \in \mathbb{N}} b_i \cdot x_1^i$$

with

$$b_i := \sum_{i_2,\dots,i_d \in \mathbb{N}^{d-1}} a_{i,\dots,i_d} \cdot x_2^{i_2} \dots x_d^{i_d}$$

Multi-Dimensional Functions

Consider functions $f : \mathbb{R}^d \to \mathbb{R}$ analytic in a neighborhood of $0 \in \mathbb{C}^d$. Their power series is given by

$$\sum_{n_1,\dots,n_d \in \mathbb{N}} a_{n_1,\dots,n_d} x_1^{n_1} \dots x_d^{n_d}$$

The beforementioned representation can be generalized.
The main idea is let

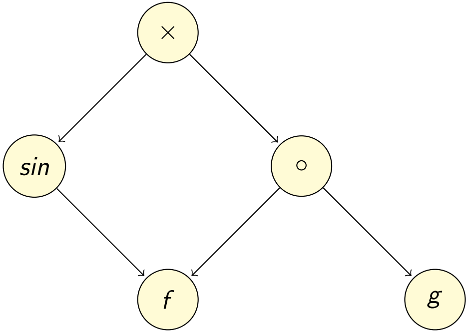
$$f(x_1,\dots,x_d) = \sum_{i \in \mathbb{N}} b_i \cdot x_1^i$$

with

$$b_i := \sum_{i_2,\dots,i_d \in \mathbb{N}^{d-1}} a_{i,\dots,i_d} \cdot x_2^{i_2} \dots x_d^{i_d}$$

b_i can be evaluated by evaluating a $d-1$ dimensional analytic function.

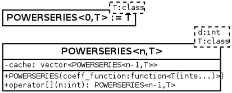
DAG approach



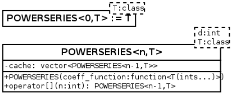
Such a DAG can be evaluated or transformed to an analytic function.

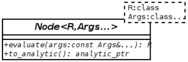
Implementation

Implementation

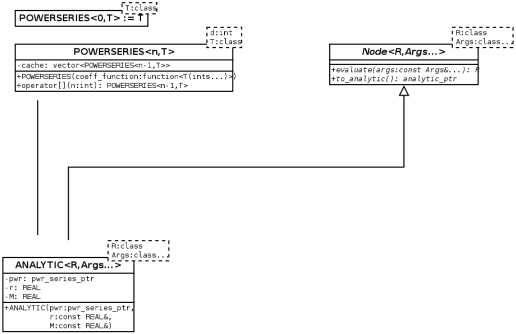


Implementation

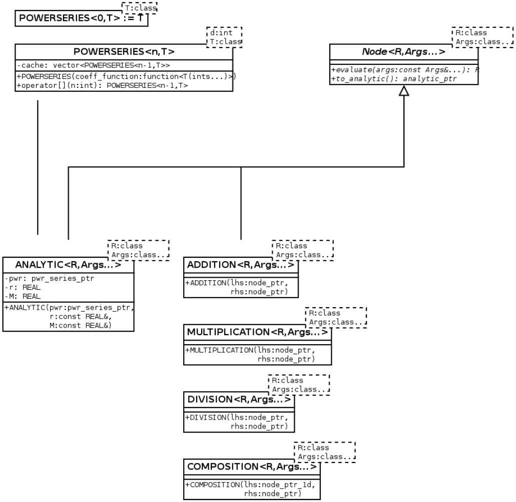




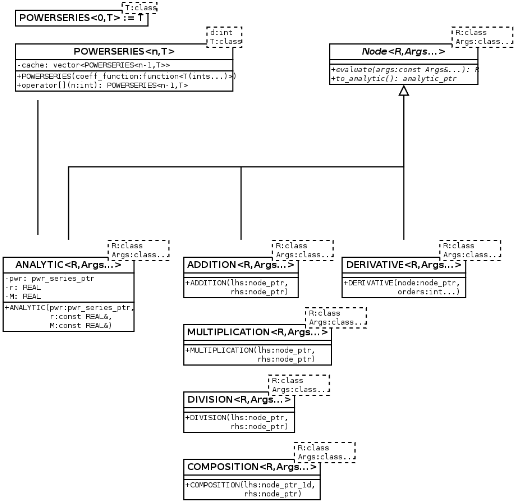
Implementation



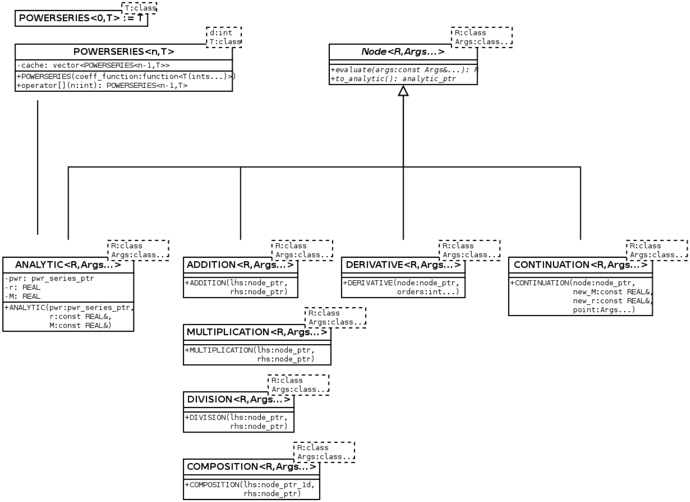
Implementation



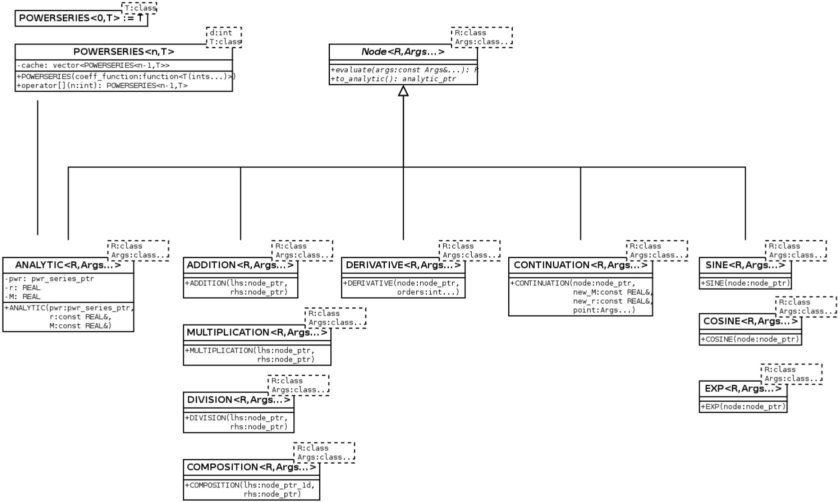
Implementation



Implementation



Implementation



ODE solving

Additionally there is an operator `solve_ivp` for initial value problems of the form

$$\begin{aligned} \dot{y}_v(t) &= F_v(t, y_1(t), \ldots, y_t(t)) \\ y_v(0) &= 0 \text{ for } v = 1, \ldots, d \end{aligned}$$

For initial values different from 0 first compute the power series around another point.

ODE solving

Additionally there is an operator `solve_ivp` for initial value problems of the form

$$\begin{aligned} \dot{y}_v(t) &= F_v(t, y_1(t), \ldots, y_t(t)) \\ y_v(0) &= 0 \text{ for } v = 1, \ldots, d \end{aligned}$$

For initial values different from 0 first compute the power series around another point.
The solution has parameters $r' = \min(r, \frac{r}{M})$ $M' = r$

Increasing the radius

$$r' = \min(r, \frac{r}{M})$$

- Usually radius of the solution function will get very small

Increasing the radius

$$r' = \min(r, \frac{r}{M})$$

- Usually radius of the solution function will get very small
- One can iteratively compute new initial values

Increasing the radius

$$r' = \min(r, \frac{r}{M})$$

- Usually radius of the solution function will get very small
- One can iteratively compute new initial values
- Compute $y_1 = y(t')$ for some small enough t'

Increasing the radius

$$r' = \min(r, \frac{r}{M})$$

- Usually radius of the solution function will get very small
- One can iteratively compute new initial values
- Compute $y_1 = y(t')$ for some small enough t'
- Solve the ODE $\dot{y}(t) = F(t, y(t))$; $y(t') = y_1$

Conclusion

- Presented a data-type for multi dimensional analytic functions

Conclusion

- Presented a data-type for multi dimensional analytic functions
- Several operators implemented

Conclusion

- Presented a data-type for multi dimensional analytic functions
- Several operators implemented
- Experiments show that from dimension 4 operations on the power series become quite slow

Conclusion

- Presented a data-type for multi dimensional analytic functions
- Several operators implemented
- Experiments show that from dimension 4 operations on the power series become quite slow
- Can also solve systems of ordinary differential equations with the right hand side functions given by the type

Conclusion

- Presented a data-type for multi dimensional analytic functions
- Several operators implemented
- Experiments show that from dimension 4 operations on the power series become quite slow
- Can also solve systems of ordinary differential equations with the right hand side functions given by the type
- Radius of solution will usually get small so that many steps are needed

Conclusion

- Presented a data-type for multi dimensional analytic functions
- Several operators implemented
- Experiments show that from dimension 4 operations on the power series become quite slow
- Can also solve systems of ordinary differential equations with the right hand side functions given by the type
- Radius of solution will usually get small so that many steps are needed
- Specialized functions for different types of ODEs?

Conclusion

- Presented a data-type for multi dimensional analytic functions
- Several operators implemented
- Experiments show that from dimension 4 operations on the power series become quite slow
- Can also solve systems of ordinary differential equations with the right hand side functions given by the type
- Radius of solution will usually get small so that many steps are needed
- Specialized functions for different types of ODEs?
- Parallelization