Solving Analytic Differential Equations in Polynomial Time over Unbounded Domains

Olivier Bournez Daniel S. Graça Amaury Pouly

ENS Lyon

May 24, 2011

Outline

The case of integers

Many models:

- Recursive functions
- Turing machines
- λ-calculus
- circuits
- $\ldots$

The case of integers

Many models:

- Recursive functions
- Turing machines
- λ-calculus
- circuits
- ...

And

Church Thesis

All reasonable discrete models of computation are equivalent.

The case of analog computations

Several models:

- BSS model (Blum Shub Smale)
- Computable analysis
- GPAC (General Purpose Analog Computer)
- ...

The case of analog computations

Several models:

- BSS model (Blum Shub Smale)
- Computable analysis
- GPAC (General Purpose Analog Computer)
- ...

Questions:

- Church Thesis for analog computers ?

The case of analog computations

Several models:

- BSS model (Blum Shub Smale)
- Computable analysis
- GPAC (General Purpose Analog Computer)
- ...

Questions:

- Church Thesis for analog computers ? $\Rightarrow$ No (GPAC $\subsetneq$ BSS)

The case of analog computations

Several models:

- BSS model (Blum Shub Smale)
- Computable analysis
- GPAC (General Purpose Analog Computer)
- ...

Questions:

- Church Thesis for analog computers ? $\Rightarrow$ No (GPAC $\subsetneq$ BSS)
- Comparison with digital models of computation ?

The case of analog computations

Several models:

- BSS model (Blum Shub Smale)
- Computable analysis
- GPAC (General Purpose Analog Computer)
- ...

Questions:

- Church Thesis for analog computers ? $\Rightarrow$ No (GPAC $\subsetneq$ BSS)
- Comparison with digital models of computation ? $\Rightarrow$ How ?

The case of analog computations

Several models:

- BSS model (Blum Shub Smale)
- Computable analysis
- GPAC (General Purpose Analog Computer)
- ...

Questions:

- Church Thesis for analog computers ? $\Rightarrow$ No (GPAC $\subsetneq$ BSS)
- Comparison with digital models of computation ? $\Rightarrow$ How ?
- What is a reasonable model ?

GPAC

General Purpose Analog Computer

- by Claude Shanon (1941)
- idealization of an analog computer: Differential Analyzer
- circuit by from:

GPAC

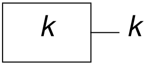
General Purpose Analog Computer

- by Claude Shanon (1941)
- idealization of an analog computer: Differential Analyzer
- circuit by from:

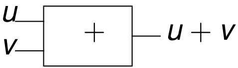
GPAC

General Purpose Analog Computer

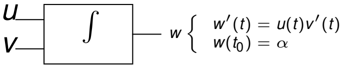
- by Claude Shanon (1941)
- idealization of an analog computer: Differential Analyzer
- circuit by from:



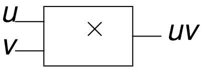
A constant unit



An adder unit



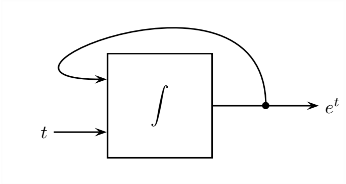
An integrator unit



A multiplier unit

GPAC: examples

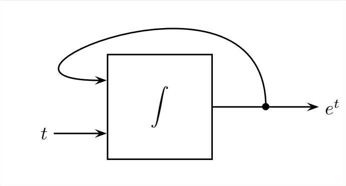
Example (Exponential)



$$\begin{cases} y' = y \\ y(0) = 1 \end{cases}$$

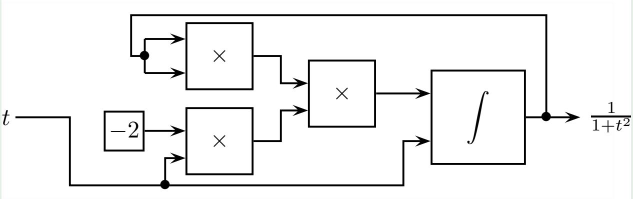
GPAC: examples

Example (Exponential)



$$\begin{cases} y' = y \\ y(0) = 1 \end{cases}$$

Example (Nonlinear)



$$\begin{cases} y' = -2ty^2 \\ y(0) = 1 \end{cases}$$

GPAC: beyond the circuit approach

Theorem

y is generated by a GPAC iff it is a component of the solution $y = (y_1, \ldots, y_d)$ of the ordinary differential equation (ODE):

$$\begin{cases} \dot{y} &= p(y) \\ y(t_0) &= y_0 \end{cases}$$

where p is a vector or polynomials.

Example (Counter-example)

$$\begin{cases} \dot{y} &= \frac{1}{y} \\ y(0) &= 1 \end{cases}$$

GPAC: beyond the circuit approach

Theorem

y is generated by a GPAC iff it is a component of the solution $y = (y_1, \ldots, y_d)$ of the ordinary differential equation (ODE):

$$\begin{cases} \dot{y} &= p(y) \\ y(t_0) &= y_0 \end{cases}$$

where p is a vector or polynomials.

Example (Counter-example)

$$\begin{cases} \dot{y} &= \frac{1}{y} \\ y(0) &= 1 \end{cases}$$

Computable real

Definition

A real $r \in \mathbb{R}$ is computable is one can compute an arbitrary close approximation for a given precision:

$$\text{Given } p \in \mathbb{N}, \text{ compute } r_p \text{ s.t. } |r - r_p| \leqslant 2^{-p}$$

Computable real

Definition

A real $r \in \mathbb{R}$ is computable is one can compute an arbitrary close approximation for a given precision:

$$\text{Given } p \in \mathbb{N}, \text{ compute } r_p \text{ s.t. } |r - r_p| \leqslant 2^{-p}$$

Computable real

Definition

A real $r \in \mathbb{R}$ is computable is one can compute an arbitrary close approximation for a given precision:

$$\text{Given } p \in \mathbb{N}, \text{ compute } r_p \text{ s.t. } |r - r_p| \leqslant 2^{-p}$$

Example

Rationals, π, e, ...

Computable real

Definition

A real $r \in \mathbb{R}$ is computable is one can compute an arbitrary close approximation for a given precision:

$$\text{Given } p \in \mathbb{N}, \text{ compute } r_p \text{ s.t. } |r - r_p| \leqslant 2^{-p}$$

Example

Rationals, π, e, ...

Counter-Example

$$r = \sum_{n=0}^{\infty} d_n 2^{-n}$$

where

$$d_n = 1 \Leftrightarrow \text{the } n^{th} \text{ Turing Machine halts on input } n$$

Computable function

Definition

A function $f : \mathbb{R} \to \mathbb{R}$ is computable if there exist a Turing Machine M s.t. for any $x \in \mathbb{R}$ and oracle $\mathcal{O}$ computing x, $M^{\mathcal{O}}$ computes $f(x)$.

Definition (Simplified)

A function $f : \mathbb{R} \to \mathbb{R}$ is computable if f is continuous and for a any rational r one can compute $f(r)$.

Computable function

Definition

A function $f : \mathbb{R} \to \mathbb{R}$ is computable if there exist a Turing Machine M s.t. for any $x \in \mathbb{R}$ and oracle $\mathcal{O}$ computing x, $M^{\mathcal{O}}$ computes $f(x)$.

Definition (Simplified)

A function $f : \mathbb{R} \to \mathbb{R}$ is computable if f is continuous and for a any rational r one can compute $f(r)$.

Example

Polynomials, trigonometric functions, $e^{\cdot}$, $\sqrt{\cdot}$, ...

Computable function

Definition

A function $f : \mathbb{R} \to \mathbb{R}$ is computable if there exist a Turing Machine M s.t. for any $x \in \mathbb{R}$ and oracle $\mathcal{O}$ computing x, $M^{\mathcal{O}}$ computes $f(x)$.

Definition (Simplified)

A function $f : \mathbb{R} \to \mathbb{R}$ is computable if f is continuous and for a any rational r one can compute $f(r)$.

Example

Polynomials, trigonometric functions, $e^{\cdot}$, $\sqrt{\cdot}$, ...

Counter-Example

$$f(x) = \lceil x \rceil$$

Church Thesis

We have:

- TM: $\mathbb{N} \to \mathbb{N}$
- GPAC: $\mathbb{R} \to \mathbb{R}$, analytic ($\Rightarrow C^\infty$)
- CA: $\mathbb{R} \to \mathbb{R}$, continuous

Church Thesis ?

Church Thesis

We have:

- TM: $\mathbb{N} \to \mathbb{N}$
- GPAC: $\mathbb{R} \to \mathbb{R}$, analytic ($\Rightarrow C^{\infty}$)
- CA: $\mathbb{R} \to \mathbb{R}$, continuous

Church Thesis ?

- TM $=$ CA: definition

Church Thesis

We have:

- TM: $\mathbb{N} \to \mathbb{N}$
- GPAC: $\mathbb{R} \to \mathbb{R}$, analytic ($\Rightarrow C^\infty$)
- CA: $\mathbb{R} \to \mathbb{R}$, continuous

Church Thesis ?

- TM $=$ CA: definition
- TM $\subseteq$ GPAC: simulating a TM with an ODE

Church Thesis

We have:

- TM: $\mathbb{N} \to \mathbb{N}$
- GPAC: $\mathbb{R} \to \mathbb{R}$, analytic ($\Rightarrow C^\infty$)
- CA: $\mathbb{R} \to \mathbb{R}$, continuous

Church Thesis ?

- TM $=$ CA: definition
- TM $\subseteq$ GPAC: simulating a TM with an ODE
- GPAC $\subseteq$ CA: computing the solution of an ODE

Church Thesis

We have:

- TM: $\mathbb{N} \to \mathbb{N}$
- GPAC: $\mathbb{R} \to \mathbb{R}$, analytic ($\Rightarrow C^\infty$)
- CA: $\mathbb{R} \to \mathbb{R}$, continuous

Church Thesis ?

- TM $=$ CA: definition
- TM $\subseteq$ GPAC: simulating a TM with an ODE
- GPAC $\subseteq$ CA: computing the solution of an ODE
- CA $\subseteq$ GPAC: approximating a computable function with a ODE

Church Thesis

We have:

- TM: $\mathbb{N} \to \mathbb{N}$
- GPAC: $\mathbb{R} \to \mathbb{R}$, analytic ($\Rightarrow C^\infty$)
- CA: $\mathbb{R} \to \mathbb{R}$, continuous

Church Thesis ?

- TM $=$ CA: definition
- TM $\subseteq$ GPAC: simulating a TM with an ODE
- GPAC $\subseteq$ CA: computing the solution of an ODE
- CA $\subseteq$ GPAC: approximating a computable function with a ODE

Church Thesis

Turing Machines, GPAC and Computable analysis are equivalent models of computations.

Really ?

▷ GPAC $\subseteq$ CA: computing the solution of an ODE:
"Of course, the resulting algorithms are highly inefficient in practice"

Really ?

▷ GPAC $\subseteq$ CA: computing the solution of an ODE:
"Of course, the resulting algorithms are highly inefficient in practice"

Effective Church Thesis ?

Are all (sufficiently powerful) "reasonable" models of computations with "reasonable" measure of time polynomially equivalent ?

Computational complexity

- TM: well known and understood
- CA:
  - A few technical differences
  - Relatively clear
- GPAC: unclear

Computational complexity

- TM: well known and understood
- CA:
  - A few technical differences
  - Relatively clear
- GPAC: unclear

Computational complexity

- TM: well known and understood
- CA:
  - A few technical differences
  - Relatively clear
- GPAC: unclear

Computational complexity

- TM: well known and understood
- CA:
  - A few technical differences
  - Relatively clear
- GPAC: unclear

Current situation: TM *vs* GPAC

$$\text{TM} \subseteq \text{GPAC}$$

- Simulating a TM with an ODE
- Preserves time: state at step $n \Leftrightarrow$ function value at time n
- Satisfying

Current situation: TM *vs* GPAC

$$\text{TM} \subseteq \text{GPAC}$$

- Simulating a TM with an ODE
- Preserves time: state at step $n \Leftrightarrow$ function value at time n
- Satisfying

Current situation: TM *vs* GPAC

$$\text{TM} \subseteq \text{GPAC}$$

- Simulating a TM with an ODE
- Preserves time: state at step $n \Leftrightarrow$ function value at time n
- Satisfying

Current situation: TM *vs* GPAC

$$\text{TM} \subseteq \text{GPAC}$$

- Simulating a TM with an ODE
- Preserves time: state at step $n \Leftrightarrow$ function value at time n
- Satisfying

$$\text{GPAC} \subseteq \text{CA}$$

- Computing the solution of an ODE quickly
- Lots of algorithms…
- Few theoretical results
- Not satisfying

Current situation: TM *vs* GPAC

$$\text{TM} \subseteq \text{GPAC}$$

- Simulating a TM with an ODE
- Preserves time: state at step $n \Leftrightarrow$ function value at time n
- Satisfying

$$\text{GPAC} \subseteq \text{CA}$$

- Computing the solution of an ODE quickly
- Lots of algorithms...
- Few theoretical results
- Not satisfying

Current situation: TM *vs* GPAC

$$\text{TM} \subseteq \text{GPAC}$$

- Simulating a TM with an ODE
- Preserves time: state at step $n \Leftrightarrow$ function value at time n
- Satisfying

$$\text{GPAC} \subseteq \text{CA}$$

- Computing the solution of an ODE quickly
- Lots of algorithms...
- Few theoretical results
- Not satisfying

Current situation: TM *vs* GPAC

$$\text{TM} \subseteq \text{GPAC}$$

- Simulating a TM with an ODE
- Preserves time: state at step $n \Leftrightarrow$ function value at time n
- Satisfying

$$\text{GPAC} \subseteq \text{CA}$$

- Computing the solution of an ODE quickly
- Lots of algorithms…
- Few theoretical results
- Not satisfying

Problem statement & facts

We want to solve:

$$\begin{cases} \dot{y} & = p(y) \\ y(t_0) & = y_0 \end{cases}$$

Solve ?

▷Compute $y_i(t)$ with arbitrary precision

Properties & hypothesis:

- Assume y defined over $\mathbb{R}$: no loss of generality
- y is analytical over $\mathbb{R}$:

$$y(t+\varepsilon) = \sum_{n=0}^{\infty} a_n \varepsilon^n$$

- Problem: local Taylor series, difficult to use
- Idea: stronger assumption: y analytical over $\mathbb{C}$
- Consequence: Taylor series valid over $\mathbb{C}$

Problem statement & facts

We want to solve:

$$\begin{cases} \dot{y} &= p(y) \\ y(t_0) &= y_0 \end{cases}$$

Solve ?

▷Compute $y_i(t)$ with arbitrary precision

Properties & hypothesis:

- Assume y defined over $\mathbb{R}$: no loss of generality
- y is analytical over $\mathbb{R}$:

$$y(t+\varepsilon) = \sum_{n=0}^{\infty} a_n \varepsilon^n$$

- Problem: local Taylor series, difficult to use
- Idea: stronger assumption: y analytical over $\mathbb{C}$
- Consequence: Taylor series valid over $\mathbb{C}$

Problem statement & facts

We want to solve:

$$\begin{cases} \dot{y} &= p(y) \\ y(t_0) &= y_0 \end{cases}$$

Solve ?
▷Compute $y_i(t)$ with arbitrary precision

Properties & hypothesis:

- Assume y defined over $\mathbb{R}$: no loss of generality
- y is analytical over $\mathbb{R}$:

$$y(t+\varepsilon) = \sum_{n=0}^{\infty} a_n \varepsilon^n$$

- Problem: local Taylor series, difficult to use
- Idea: stronger assumption: y analytical over $\mathbb{C}$
- Consequence: Taylor series valid over $\mathbb{C}$

Problem statement & facts

We want to solve:

$$\begin{cases} \dot{y} &= p(y) \\ y(t_0) &= y_0 \end{cases}$$

Solve ?
▷Compute $y_i(t)$ with arbitrary precision

Properties & hypothesis:

- Assume y defined over $\mathbb{R}$: no loss of generality
- y is analytical over $\mathbb{R}$:

$$y(t+\varepsilon) = \sum_{n=0}^{\infty} a_n \varepsilon^n$$

- Problem: local Taylor series, difficult to use
- Idea: stronger assumption: y analytical over $\mathbb{C}$
- Consequence: Taylor series valid over $\mathbb{C}$

Problem statement & facts

We want to solve:

$$\begin{cases} \dot{y} &= p(y) \\ y(t_0) &= y_0 \end{cases}$$

Solve ?

▷Compute $y_i(t)$ with arbitrary precision

Properties & hypothesis:

- Assume y defined over $\mathbb{R}$: no loss of generality
- y is analytical over $\mathbb{R}$:

$$y(t+\varepsilon) = \sum_{n=0}^{\infty} a_n \varepsilon^n$$

- Problem: local Taylor series, difficult to use
- Idea: stronger assumption: y analytical over $\mathbb{C}$
- Consequence: Taylor series valid over $\mathbb{C}$

Problem statement & facts

We want to solve:

$$\begin{cases} \dot{y} &= p(y) \\ y(t_0) &= y_0 \end{cases}$$

Solve ?
▷Compute $y_i(t)$ with arbitrary precision

Properties & hypothesis:

- Assume y defined over $\mathbb{R}$: no loss of generality
- y is analytical over $\mathbb{R}$:

$$y(t+\varepsilon) = \sum_{n=0}^{\infty} a_n \varepsilon^n$$

- Problem: local Taylor series, difficult to use
- Idea: stronger assumption: y analytical over $\mathbb{C}$
- Consequence: Taylor series valid over $\mathbb{C}$

Problem statement & facts

We want to solve:

$$\begin{cases} \dot{y} &= p(y) \\ y(t_0) &= y_0 \end{cases}$$

Solve ?

▷Compute $y_i(t)$ with arbitrary precision

Properties & hypothesis:

- Assume y defined over $\mathbb{R}$: no loss of generality
- y is analytical over $\mathbb{R}$:

$$y(t+\varepsilon) = \sum_{n=0}^{\infty} a_n \varepsilon^n$$

- Problem: local Taylor series, difficult to use
- Idea: stronger assumption: y analytical over $\mathbb{C}$
- Consequence: Taylor series valid over $\mathbb{C}$

New problems

Two problems:

- Compute Taylor series from function and vice-versa
- Compute Taylor series of the solution to an ODE

Poly-boundedness

Definition

$f : \mathbb{R} \to \mathbb{R}$ poly-bounded if $|f(t)| \leqslant e^{p(\log t)}$.

Theorem

$f : \mathbb{R} \to \mathbb{R}$ computable in polynomial time $\Rightarrow$ f poly-bounded

Theorem

If $f : \mathbb{C} \to \mathbb{C}$ analytical over $\mathbb{C}$: $f(t) = \sum_{n=0}^{\infty} a_n t^n$ and poly-bounded then:

$$f \text{ polynomial time computable}$$

$$\Updownarrow$$

$$\{a_n\} \text{ polynomial time computable}$$

Poly-boundedness

Definition

$f : \mathbb{R} \to \mathbb{R}$ poly-bounded if $|f(t)| \leqslant e^{p(\log t)}$.

Theorem

$f : \mathbb{R} \to \mathbb{R}$ computable in polynomial time $\Rightarrow$ f poly-bounded

Theorem

If $f : \mathbb{C} \to \mathbb{C}$ analytical over $\mathbb{C}$: $f(t) = \sum_{n=0}^{\infty} a_n t^n$ and poly-bounded then:

$$f \text{ polynomial time computable}$$

$$\Updownarrow$$

$$\{a_n\} \text{ polynomial time computable}$$

Poly-boundedness

Definition

$f : \mathbb{R} \to \mathbb{R}$ poly-bounded if $|f(t)| \leqslant e^{p(\log t)}$.

Theorem

$f : \mathbb{R} \to \mathbb{R}$ computable in polynomial time $\Rightarrow$ f poly-bounded

Theorem

If $f : \mathbb{C} \to \mathbb{C}$ analytical over $\mathbb{C}$: $f(t) = \sum_{n=0}^{\infty} a_n t^n$ and poly-bounded then:

$$f \text{ polynomial time computable}$$

$$\Updownarrow$$

$$\{a_n\} \text{ polynomial time computable}$$

Result

Theorem

If y is poly-bounded, analytical over $\mathbb{C}$ and satisfies

$$\begin{cases} \dot{y} &= p(y) \\ y(t_0) &= y_0 \end{cases}$$

Then y is polynomial time computable.

- Poly-bounded: optimal, easy to check
- Analytical over $\mathbb{C}$: too strong, hard to check

Result

Theorem

If y is poly-bounded, analytical over $\mathbb{C}$ and satisfies

$$\begin{cases} \dot{y} &= p(y) \\ y(t_0) &= y_0 \end{cases}$$

Then y is polynomial time computable.

- Poly-bounded: optimal, easy to check
- Analytical over $\mathbb{C}$: too strong, hard to check

Result

Theorem

If y is poly-bounded, analytical over $\mathbb{C}$ and satisfies

$$\begin{cases} \dot{y} & = p(y) \\ y(t_0) & = y_0 \end{cases}$$

Then y is polynomial time computable.

- Poly-bounded: optimal, easy to check
- Analytical over $\mathbb{C}$: too strong, hard to check

Limitations and workaround

Consider:

$$\begin{cases} \dot{y} &= -2ty^2 \\ y(0) &= 1 \end{cases} \Rightarrow \quad y(t) = \frac{1}{1+t^2}$$

Problem:

- two poles over $\mathbb{C}$: i and $-i$
- Workaround possible !
- But there is a real limitation here

Limitations and workaround

Consider:

$$\begin{cases} \dot{y} &= -2ty^2 \\ y(0) &= 1 \end{cases} \quad \Rightarrow \quad y(t) = \frac{1}{1+t^2}$$

Problem:

- two poles over $\mathbb{C}$: i and $-i$
- Workaround possible !
- But there is a real limitation here

Limitations and workaround

Consider:

$$\begin{cases} \dot{y} &= -2ty^2 \\ y(0) &= 1 \end{cases} \qquad \Rightarrow \qquad y(t) = \frac{1}{1+t^2}$$

Problem:

- two poles over $\mathbb{C}$: i and $-i$
- Workaround possible !
- But there is a real limitation here

Limitations and workaround

Consider:

$$\begin{cases} \dot{y} &= -2ty^2 \\ y(0) &= 1 \end{cases} \quad \Rightarrow \quad y(t) = \frac{1}{1+t^2}$$

Problem:

- two poles over $\mathbb{C}$: i and $-i$
- Workaround possible !
- But there is a real limitation here

Future work

- Develop method specific to $\mathbb{R}$
- Understand what complexity means for the GPAC

Questions ?

- Do you have any questions ?