Continuous dynamical systems and computation

Computing with polynomial ODE's

Manuel Campagnolo

DM/ISA, Technical University of Lisbon
and
SQIG/Instituto de Telecomunicações, Lisbon

mlc@math.isa.utl.pt

CUNY Logic Workshop, New York City, March 12, 2010

Outline

Motivation

- How can we link computation and continuous dynamical systems?

Motivation

- How can we link computation and continuous dynamical systems?
- What does this tell us about continuous dynamical systems?

Motivation

- How can we link computation and continuous dynamical systems?
- What does this tell us about continuous dynamical systems?
- Is there a canonical continuous dynamical system with respect to computability?

The differential analyser

Computing in continuous time with an analog computer:

> *The differential analyser, which concept dates to Lord Kelvin and his brother James Thomson in 1876, and was constructed at MIT under the supervision of Vannevar Bush in 1932.*

The differential analyser

Figure: Hartree and Porter Meccano differential analyzer originally built in 1934

Analog circuits

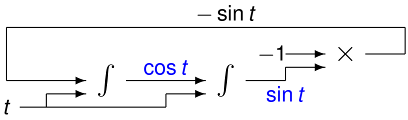


Figure: A circuit that calculates sin and cos. Its initial conditions are $\sin(0) = 0$ and $\cos(0) = 1$. The output w of the integrator unit $\int$ obeys $dw = u\,dv$ where u and v are its upper and lower inputs respectively.

Analog circuits

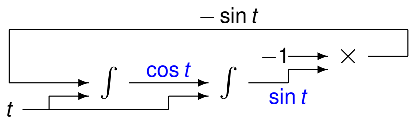


Figure: A circuit that calculates sin and cos. Its initial conditions are $\sin(0) = 0$ and $\cos(0) = 1$. The output w of the integrator unit $\int$ obeys $dw = u\,dv$ where u and v are its upper and lower inputs respectively.

The circuit above is represented by the system of equations

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= -y_1 \end{aligned}$$

which solution is $y_1 = \sin t$ and $y_2 = \cos t$ given the initial conditions $y_1(0) = 0$ and $y_2(0) = 1$.

Shannon's GPAC

In 1941 Claude Shannon proposed a mathematical model for the Differential Analyser.

Shannon proved that, given a sufficient number of integrators, any differentially algebraic function, i.e., a solution of

$$p(t, y, y', \ldots, y^{(n)}) = 0$$

could be generated.

Shannon's GPAC

In 1941 Claude Shannon proposed a mathematical model for the Differential Analyser.

Shannon proved that, given a sufficient number of integrators, any differentially algebraic function, i.e., a solution of

$$p(t, y, y', \ldots, y^{(n)}) = 0$$

could be generated.

Shannon's model was refined by Graça and Costa (2003) who showed that all non-degenerate GPAC functions are precisely PIVP functions, i.e. solutions of polynomial initial value problems

$$\bar{y}' = p(\bar{y}, t) \quad , \quad \bar{y}(0) = \bar{x}.$$

Dynamical systems

A discrete dynamical system $[X, \mathbb{N}, \phi]$ defined on the topological space X over $\mathbb{N}$ is a function $\phi : X \times \mathbb{N} \to X$ with the properties:

1. initial condition: $\phi(x, 0) = x$ for all $x \in X$;
2. semigroup property: $\phi(\phi(x, t_1), t_2) = \phi(x, t_1 + t_2)$ for all $t_1, t_2 \in \mathbb{N}$ (for $\phi(.,.) \in X$).

A continuous dynamical system $[X, \mathbb{R}_0^+, \phi]$ is defined analogously, where $\mathbb{N}$ is replaced by $\mathbb{R}_0^+$.

Dynamical systems

Maps and flows:

- A discrete dynamical system can be written as $y_{t+1} = f(y_t)$, with $y_0 = x$, where $f(.) = \phi(., 1)$ is called the transition function.

Dynamical systems

Maps and flows:

- A discrete dynamical system can be written as $y_{t+1} = f(y_t)$, with $y_0 = x$, where $f(.) = \phi(., 1)$ is called the transition function.

- If ϕ is continuously differentiable with respect to t then, a continuous dynamical system gives rise to an initial value problem $y' = f(t, y)$, $y(0) = x$, where f is called a vector field.

- On $[0,1] \times [0,1]$ there is a piecewise linear function that simulates the transition function of an arbitrary TM (Moore 1991)

- On $[0,1] \times [0,1]$ there is a piecewise linear function that simulates the transition function of an arbitrary TM (Moore 1991)
- On $\mathbb{R}$ there is an analytic closed form function that simulates the transition function of an arbitrary TM (Moore and Koiran 1996)

- On $[0,1] \times [0,1]$ there is a piecewise linear function that simulates the transition function of an arbitrary TM (Moore 1991)
- On $\mathbb{R}$ there is an analytic closed form function that simulates the transition function of an arbitrary TM (Moore and Koiran 1996)
- 3-dimensional piecewise constant differential flows on bounded domains simulate the dynamics of an arbitrary TM (Asarin, Maler and Pnueli 1995)

- On $[0, 1] \times [0, 1]$ there is a piecewise linear function that simulates the transition function of an arbitrary TM (Moore 1991)
- On $\mathbb{R}$ there is an analytic closed form function that simulates the transition function of an arbitrary TM (Moore and Koiran 1996)
- 3-dimensional piecewise constant differential flows on bounded domains simulate the dynamics of an arbitrary TM (Asarin, Maler and Pnueli 1995)
- On $\mathbb{R}^3$ there is a continuous flow $y' = f(t, y)$ that simulates the dynamics of an arbitrary TM (Branicky 1996)

- On $[0,1] \times [0,1]$ there is a piecewise linear function that simulates the transition function of an arbitrary TM (Moore 1991)
- On $\mathbb{R}$ there is an analytic closed form function that simulates the transition function of an arbitrary TM (Moore and Koiran 1996)
- 3-dimensional piecewise constant differential flows on bounded domains simulate the dynamics of an arbitrary TM (Asarin, Maler and Pnueli 1995)
- On $\mathbb{R}^3$ there is a continuous flow $y' = f(t, y)$ that simulates the dynamics of an arbitrary TM (Branicky 1996)
- It is conjectured that no analytic map on a compact, finite-dimensional space can simulate a Turing machine through a reasonable encoding (Moore 1998).

Polynomial initial value problems

- Analytic solutions;

Polynomial initial value problems

- Analytic solutions;
- Common: they define most of the usual mathematical functions, in particular the “elementary functions” of Analysis;

Polynomial initial value problems

- Analytic solutions;
- Common: they define most of the usual mathematical functions, in particular the “elementary functions” of Analysis;
- Widely used: e.g. Lorenz, Lokta-Volterra, or Van der Pol equations;

Polynomial initial value problems

- Analytic solutions;
- Common: they define most of the usual mathematical functions, in particular the "elementary functions" of Analysis;
- Widely used: e.g. Lorenz, Lokta-Volterra, or Van der Pol equations;
- Challenging: many open questions;

Polynomial initial value problems

- Analytic solutions;
- Common: they define most of the usual mathematical functions, in particular the “elementary functions” of Analysis;
- Widely used: e.g. Lorenz, Lokta-Volterra, or Van der Pol equations;
- Challenging: many open questions;
- They satisfy an elimination result: Given $y' = f(y, t)$ and $y(0) = x$, with the components of f being compositions of polynomials and PIVP functions, y is given by the first components of some PIVP function (Graça, Buescu and Campagnolo 2009).

Polynomial initial value problems

Some particular questions for PIVPs:

- Are PIVP functions computable?

Polynomial initial value problems

Some particular questions for PIVPs:

- Are PIVP functions computable?
- Is the domain of the solution computable?

Polynomial initial value problems

Some particular questions for PIVPs:

- Are PIVP functions computable?
- Is the domain of the solution computable?
- Is it even decidable if the domain is bounded?

Polynomial initial value problems

Some particular questions for PIVPs:

- Are PIVP functions computable?
- Is the domain of the solution computable?
- Is it even decidable if the domain is bounded?
- Is the reachability problem for PIVPs decidable?

Overview

$$
\begin{array}{ccccc}
\text{Unbounded domain} & & \text{Suspension in } \mathbb{R}^n & & \\
& y_{t+1} = f(y_t) & \longrightarrow & y' = p(t, y) & \\
\text{Encoding in } \mathbb{N}^n & \nwarrow & & \swarrow & \text{computation of } y \\
& & \text{TM} & &
\end{array}
$$

- TM: Turing machine
- Discrete dynamical system:
  $y_0 = x,\ y_{t+1} = f(y_t),\ x \in \mathbb{N}^n,\ f : \mathbb{N}^n \to \mathbb{N}^n,\ t \in \mathbb{N}$
- Continuous dynamical system:
  $y(0) = x,\ y' = p(t, y),\ x \in \mathbb{R}^n,\ p : \mathbb{R}^{n+1} \to \mathbb{R}^n,\ t \in \mathbb{R}_0^+$

Computability of solutions

Definition (PIVP function with parameters in S)

$\bar{y}$ (or y_i) is a PIVP function with parameters in S if $\bar{y}$ is the solution of a polynomial initial value problem $\bar{y}' = p(\bar{y}, t)$, $\bar{y}(0) = \bar{y}_0$ where the coefficients of p and the components of $\bar{y}_0$ are in S.

Computability of solutions

Definition (PIVP function with parameters in S)

$\bar{y}$ (or y_i) is a PIVP function with parameters in S if $\bar{y}$ is the solution of a polynomial initial value problem $\bar{y}' = p(\bar{y}, t)$, $\bar{y}(0) = \bar{y}_0$ where the coefficients of p and the components of $\bar{y}_0$ are in S.

Theorem (follows from Graça, Zhong and Buescu, 2007)

Let $\bar{y} : (\alpha, \beta) \subset \mathbb{R} \to \mathbb{R}^k$ be a PIVP function with computable parameters. Then, $\bar{y}$ is computable on (α, β).

Encoding TMs

Encoding Turing machine as a discrete dynamical system on $\mathbb{N}^3$:

$$\begin{array}{ccc} (\omega, q, h) & \xrightarrow{\psi} & x \in \mathbb{N}^3 \\ \delta \downarrow & & \downarrow f \\ (\omega', q', h') & \xrightarrow{\psi} & f(x) \in \mathbb{N}^3 \end{array}$$

If the tape contents is

$$\ldots 000 a_{-p} \ldots a_{-1} a_0 a_1 \ldots a_n 000 \ldots$$

$a_i \in \{1, \ldots, 9\}$, $q \in \{1, \ldots, m\}$, then

$$\psi : \begin{cases} x_1 = a_0 + a_1\, 10 + \cdots + a_n\, 10^n \\ x_2 = a_{-1} + a_{-2}\, 10 + \cdots + a_{-p}\, 10^{p-1} \\ x_3 = q \end{cases}$$

Extension

Definition

The map $\Omega : \mathbb{R}^m \to \mathbb{R}^m$ is a robust extension of the map $\omega : \mathbb{N}^m \to \mathbb{N}^m$ if there are $\delta_{in}, \delta_{ev}, \delta_{out} \in (0, \frac{1}{2})$ such that

$$||n_0 - x_0||_\infty \leq \delta_{in}, \; ||\Omega - \overline{\Omega}||_\infty \leq \delta_{ev}$$

implies that

$$||\omega^{[k]}(n_0) - \overline{\Omega}^{[k]}(x_0)||_\infty \leq \delta_{out}$$

for all $k \in \mathbb{N}$

Let $\mathbb{Q}[\pi] := \{b_n\pi^n + \cdots + b_1\pi + b_0 \in \mathbb{R} \mid b_0, \ldots, b_n \in \mathbb{Q}\}$.

Theorem (Graça, Campagnolo and Buescu, 2008)

The transition function $\omega : \mathbb{N}^3 \to \mathbb{N}^3$ of a Turing machine (under the encoding ψ) admits a robust extension $\Omega : \mathbb{R}^3 \to \mathbb{R}^3$.
Ω can be chosen to be a composition of polynomials with coefficients in $\mathbb{Q}[\pi]$ and PIVP functions with parameters in $\mathbb{Q}[\pi]$.

Let $\mathbb{Q}[\pi] := \{b_n\pi^n + \cdots + b_1\pi + b_0 \in \mathbb{R} \mid b_0, \ldots, b_n \in \mathbb{Q}\}$.

Theorem (Graça, Campagnolo and Buescu, 2008)

The transition function $\omega : \mathbb{N}^3 \to \mathbb{N}^3$ of a Turing machine (under the encoding ψ) admits a robust extension $\Omega : \mathbb{R}^3 \to \mathbb{R}^3$.
Ω can be chosen to be a composition of polynomials with coefficients in $\mathbb{Q}[\pi]$ and PIVP functions with parameters in $\mathbb{Q}[\pi]$.

Sketch of the proof.

Use a trigonometric interpolation of $x \mod 10$ to obtain a_0 from x_1. Use polynomial interpolations to represent the next state, the next symbol, the direction of the move of the head and update x_1 and x_2. Use error contracting functions (compositions of polynomials and trigonometric functions) to keep the error within, say, $\frac{1}{4}$ of the exact encoding. $\square$

Suspension

Definition (Robust suspension)

The solution ϕ of $y' = f(t, y)$, $y(0) = x \in \mathbb{N}^m$

is a robust suspension of the map $\omega : \mathbb{N}^m \to \mathbb{N}^m$ if there are $\delta_{in}, \delta_{ev}, \delta_{out}, \delta_{time} \in (0, \frac{1}{2})$ such that

$$||x - y_0||_\infty \leq \delta_{in}, \ ||f - \overline{f}||_\infty \leq \delta_{ev}, \ ||t - k||_\infty \leq \delta_{time}$$

implies that the solution $\overline{\phi}$ of $y' = \overline{f}(t, y)$, $y(0) = y_0$

satisfies

$$||\omega^{[k]}(x) - \overline{\phi}(t)||_\infty \leq \delta_{out} \ \text{ for all } k \in \mathbb{N}$$

.

Construction of the suspension: Branicky's clocks

- The solution of
$$y' = c\,(b - y)^3 \phi(t),$$
where $\phi(t) > 0$, approaches at $t = 1$ the "target" b with arbitrary precision depending on c, irrespectively of the initial condition at $t = 0$.

Construction of the suspension: Branicky's clocks

- The solution of

$$y' = c\,(b - y)^3 \phi(t),$$

  where $\phi(t) > 0$, approaches at $t = 1$ the "target" b with arbitrary precision depending on c, irrespectively of the initial condition at $t = 0$.

- The following system of polynomial ODEs allows to iterate the map Ω

$$\begin{cases} z_1' = c_1(\Omega(r(z_2)) - z_1)^3\,\theta(\sin 2\pi t) & z_1(0) = x_0 \\ z_2' = c_2(r(z_1) - z_2)^3\,\theta(-\sin 2\pi t) & z_2(0) = x_0 \end{cases}$$

  where r is a rounding function and $\theta = H(x) * x^k$.

Construction of the suspension: example

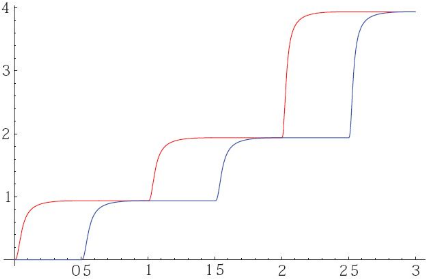


Figure: Suspension of the iteration of $f(n) = 2^n$ with ODEs

Construction of the suspension: removing the non-PIVP functions

Consider the perturbed version of the ODE $y' = c\,(b - y)^3 \phi(t)$:

$$\bar{y}' = c\,(\bar{b}(t) - \bar{y})^3 \phi(t) + e(t)$$

where $|\bar{b}(t) - b| \leq \rho$ and $|e(t)| \leq \delta$.

Construction of the suspension: removing the non-PIVP functions

Consider the perturbed version of the ODE $y' = c\,(b - y)^3 \phi(t)$:

$$\bar{y}' = c\,(\bar{b}(t) - \bar{y})^3 \phi(t) + e(t)$$

where $|\bar{b}(t) - b| \leq \rho$ and $|e(t)| \leq \delta$.

Lemma

If $|y(1) - b| \leq \gamma$ then $|\bar{y}(1) - b| \leq \gamma + \rho + \delta/2$.

Construction of the suspension: removing the non-PIVP functions

Consider the perturbed version of the ODE $y' = c\,(b-y)^3\phi(t)$:

$$\bar{y}' = c\,(\bar{b}(t) - \bar{y})^3\phi(t) + e(t)$$

where $|\bar{b}(t) - b| \leq \rho$ and $|e(t)| \leq \delta$.

Lemma

If $|y(1) - b| \leq \gamma$ then $|\bar{y}(1) - b| \leq \gamma + \rho + \delta/2$.

Again, this allows to replace the rounding function r and the control functions $\theta(\sin 2\pi t)$ by appropriate PIVP functions and re-write the suspension of Ω as a PIVP.

Suspension for discrete dynamical systems with PIVP functions

Theorem (Graça, Buescu and Campagnolo, 2009)

If the map $\omega : \mathbb{N}^m \to \mathbb{N}^m$ *admits a robust extension* $\Omega : \mathbb{R}^m \to \mathbb{R}^m$ *whose components are compositions of polynomials and PIVP functions with parameters in* $\mathbb{Q}[\pi]$*, then* ω *admits a robust suspension* ϕ *which a PIVP function with parameters in* $\mathbb{Q}[\pi]$*.*

Suspension for discrete dynamical systems with PIVP functions

Theorem (Graça, Buescu and Campagnolo, 2009)

If the map $\omega : \mathbb{N}^m \to \mathbb{N}^m$ admits a robust extension $\Omega : \mathbb{R}^m \to \mathbb{R}^m$ whose components are compositions of polynomials and PIVP functions with parameters in $\mathbb{Q}[\pi]$, then ω admits a robust suspension ϕ which a PIVP function with parameters in $\mathbb{Q}[\pi]$.

Corollary

The transition function $\omega : \mathbb{N}^3 \to \mathbb{N}^3$ of a Turing machine (under the encoding ψ) admits a robust suspension ϕ. Moreover ϕ is a PIVP function with parameters in $\mathbb{Q}[\pi]$.

Reachability

Corollary

The following problem is undecidable:

> *Given a vector of polynomial $p : \mathbb{R}^{n+1} \to \mathbb{R}^n$ with coefficients in $\mathbb{Q}[\pi]$, $y_0 \in \mathbb{Q} \times \mathbb{Q}^n$, and an open set A in $\mathbb{R}^n$ decide if the solution of*
>
> $$y' = p(t, y), \; y(0) = y_0$$
>
> *crosses A.*

Boundedness of the maximal interval of existence

Note: the maximal interval of the PIVP

$$y' = \alpha(y^2 - 1)t, \; y(0) = 3$$

is bounded for $\alpha > 0$ and unbounded for $\alpha \leq 0$.

Unless one can decide $\alpha \neq 0$ this gives rise to trivial undecidability results.

Boundedness of the maximal interval of existence

Note: the maximal interval of the PIVP

$$y' = \alpha(y^2 - 1)t, \; y(0) = 3$$

is bounded for $\alpha > 0$ and unbounded for $\alpha \leq 0$.

Unless one can decide $\alpha \neq 0$ this gives rise to trivial undecidability results.

Let's restrict the parameters of the PIVP to $\mathbb{Q}[\pi]$, which is a comparable set, i.e., given $\alpha, \beta \in \mathbb{Q}[\pi]$ we can decide if $\alpha = \beta$ and $\alpha < \beta$.

Theorem (Graça, Buescu and Campagnolo, 2009)

The following problem is undecidable:

> *Given a vector of polynomial $p : \mathbb{R}^{n+1} \to \mathbb{R}^n$ with coefficients in $\mathbb{Q}[\pi]$ and $y_0 \in \mathbb{Q}^n$, decide if the maximal interval of existence of the solution of $y' = p(t, y),\ y(0) = y_0$ is bounded.*

Theorem (Graça, Buescu and Campagnolo, 2009)

The following problem is undecidable:

> *Given a vector of polynomial $p : \mathbb{R}^{n+1} \to \mathbb{R}^n$ with coefficients in $\mathbb{Q}[\pi]$ and $y_0 \in \mathbb{Q}^n$, decide if the maximal interval of existence of the solution of $y' = p(t, y),\ y(0) = y_0$ is bounded.*

Sketch of the proof.

Let x_q be the component of the PIVP that encodes the state in the suspension of a TM M. We set M s.t. it halts iff x_q reaches m. Consider the system (equivalent to a PIVP)

$$z_1' = x_q - (m - \frac{1}{2}), \quad z_2 = \frac{1}{z_1}, \quad z_1(0) = z_2(0) = -1.$$

If M halts then z_1' becomes eventually larger than, say, 1/8 and z_2 blows up. Otherwise, z_1 always decreases and the solution is defined everywhere. □

Summary and questions (part 1)

- There are PIVPs able to simulate arbitrary Turing machines (on unbounded domains);

Summary and questions (part 1)

- There are PIVPs able to simulate arbitrary Turing machines (on unbounded domains);
- Although the solutions of PIVPs with computable parameters are computable, several properties of those dynamical systems are undecidable, even if the parameters of the system are in $\mathbb{Q}[\pi]$.

Summary and questions (part 1)

- There are PIVPs able to simulate arbitrary Turing machines (on unbounded domains);
- Although the solutions of PIVPs with computable parameters are computable, several properties of those dynamical systems are undecidable, even if the parameters of the system are in $\mathbb{Q}[\pi]$.
- PIVPs, which are a well known model of physical phenomena, are also a robust yet powerful model of continuous time computation.

Summary and questions (part 1)

- There are PIVPs able to simulate arbitrary Turing machines (on unbounded domains);
- Although the solutions of PIVPs with computable parameters are computable, several properties of those dynamical systems are undecidable, even if the parameters of the system are in $\mathbb{Q}[\pi]$.
- PIVPs, which are a well known model of physical phenomena, are also a robust yet powerful model of continuous time computation.
- Can we get rid of π and obtain the result about suspensions for PIVPs with parameters in $\mathbb{Q}$?

Summary and questions (part 1)

- There are PIVPs able to simulate arbitrary Turing machines (on unbounded domains);
- Although the solutions of PIVPs with computable parameters are computable, several properties of those dynamical systems are undecidable, even if the parameters of the system are in $\mathbb{Q}[\pi]$.
- PIVPs, which are a well known model of physical phenomena, are also a robust yet powerful model of continuous time computation.
- Can we get rid of π and obtain the result about suspensions for PIVPs with parameters in $\mathbb{Q}$?
- Can we do a more genuine suspension of discrete dynamical systems, which doesn't rely on "clocks"?

Motivation

Is $f :\subset \mathbb{R} \to \mathbb{R}$ computable?

Several notions of computability for real functions:

- Turing machine approach: Computable Analysis
- GPAC functions
- BSS machines
- ...

Computable analysis

Since real numbers and many other objects studied in analysis are "infinite" objects containing an "infinite amount of information", one has to approximate them by "finite" objects containing only a "finite amount of information" and to perform the actual computations on these finite objects.
(Brattka et al., A Tutorial on Computable Analysis, 2008)

Computable analysis

> *Since real numbers and many other objects studied in analysis are "infinite" objects containing an "infinite amount of information", one has to approximate them by "finite" objects containing only a "finite amount of information" and to perform the actual computations on these finite objects.*
> *(Brattka et al., A Tutorial on Computable Analysis, 2008)*

For instance, a real number is computable if there is a Turing machine with no input that outputs a binary expansion of that number (note: the output tape is one-way).

Computable reals

Definition (Cauchy representation)

A sequence $\{r_n\}$ of rationals is a ρ-name of a real number x if there exists three functions a, b, c from $\mathbb{N}$ to $\mathbb{N}$ such that for all $n \in \mathbb{N}$

$$r_n = (-1)^{a(n)} \frac{b(n)}{c(n)+1} \quad \text{and} \quad |r_n - x| \leq 2^{-n}.$$

Computable reals

Definition (Cauchy representation)

A sequence $\{r_n\}$ of rationals is a ρ-name of a real number x if there exists three functions a, b, c from $\mathbb{N}$ to $\mathbb{N}$ such that for all $n \in \mathbb{N}$

$$r_n = (-1)^{a(n)} \frac{b(n)}{c(n)+1} \quad \text{and} \quad |r_n - x| \leq 2^{-n}.$$

Definition (Computable real number)

$x \in \mathbb{R}$ is computable if it has a computable ρ-name, i.e., if a, b and c are computable.

Computable real functions

M is an oracle Turing machine if, at any step of the computation of M using oracle $\phi : \mathbb{N} \to \mathbb{N}^k$, M is allowed to query the value of $\phi(n)$ for any n. (Below, ϕ is a ρ-name for x.)

Definition (Computable function)

A function $f : D \subset \mathbb{R}^m \to \mathbb{R}^p$ is computable if there is an oracle Turing machine such that for any accuracy n and any ρ-name for $x \in D$ given as oracle, computes a rational vector r satisfying $||r - f(x)|| \leq 2^{-n}$.

Computable real functions

M is an oracle Turing machine if, at any step of the computation of M using oracle $\phi : \mathbb{N} \to \mathbb{N}^k$, M is allowed to query the value of $\phi(n)$ for any n. (Below, ϕ is a ρ-name for x.)

Definition (Computable function)

A function $f : D \subset \mathbb{R}^m \to \mathbb{R}^p$ is computable if there is an oracle Turing machine such that for any accuracy n and any ρ-name for $x \in D$ given as oracle, computes a rational vector r satisfying $||r - f(x)|| \leq 2^{-n}$.

In other words, the machine produces a rapidly converging rational sequence with limit $f(x)$ from a rapidly converging rational sequence with limit x.

Computable real functions

M is an oracle Turing machine if, at any step of the computation of M using oracle $\phi : \mathbb{N} \to \mathbb{N}^k$, M is allowed to query the value of $\phi(n)$ for any n. (Below, ϕ is a ρ-name for x.)

Definition (Computable function)

A function $f : D \subset \mathbb{R}^m \to \mathbb{R}^p$ is computable if there is an oracle Turing machine such that for any accuracy n and any ρ-name for $x \in D$ given as oracle, computes a rational vector r satisfying $||r - f(x)|| \leq 2^{-n}$.

In other words, the machine produces a rapidly converging rational sequence with limit $f(x)$ from a rapidly converging rational sequence with limit x.

Definition

$\mathbf{C}(\mathbb{R})$ denotes the set of computable functions.

Equivalent formulation of CA: modulus of continuity

Theorem (see Ko (1991), Corollary 2.14)

$f : [0,1] \to \mathbb{R}$ is in $\mathbf{C}(\mathbb{R})$ iff there exist three computable functions $m : \mathbb{N} \to \mathbb{N}$, $\mathrm{sgn}, \mathrm{abs} : \mathbb{N}^3 \to \mathbb{N}$ such that:

1. *m is a modulus of continuity for f, i.e. for all $n \in \mathbb{N}$ and all $x, y \in [0,1]$,*

$$|x - y| \leq 2^{-m(n)} \quad \Longrightarrow \quad |f(x) - f(y)| \leq 2^{-n}$$

2. *For all $(j,k) \in \mathbb{N}^2$ such that $\frac{j}{2^k} \in [0,1]$, and all $n \in \mathbb{N}$,*

$$\left| (-1)^{\mathrm{sgn}(j,k,n)} \frac{\mathrm{abs}(j,k,n)}{2^n} - f\left(\frac{j}{2^k}\right) \right| \leq 2^{-n}.$$

$$\mathbf{C}(\mathbb{R}) \stackrel{?}{=} \textit{PIVP functions with parameters in } X \ (X \subset \mathbb{R})$$

$\mathbf{C}(\mathbb{R}) \stackrel{?}{=}$ *PIVP functions with parameters in X ($X \subset \mathbb{R}$)*

- $\supseteq$ TRUE, if X is a subset of the computable reals (follows from Graça, Zhong and Buescu, 2007).

$\mathbf{C}(\mathbb{R}) \stackrel{?}{=}$ *PIVP functions with parameters in* X *(*$X \subset \mathbb{R}$*)*

$\supseteq$ TRUE, if X is a subset of the computable reals (follows from Graça, Zhong and Buescu, 2007).

$\subseteq$ FALSE, even if $X = \mathbb{R}$. This follows from the fact that, for instance, Euler's gamma function

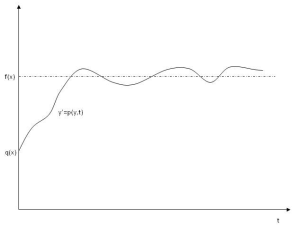
$$\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} dt$$

is not differentially algebraic (Hölder, 1887).

Approximation of computable real functions with PIVPs

Definition

$y(t; x_1, \ldots, x_k)$ is a PIVP function with parameters in S if there are polynomials $p : \mathbb{R}^n \to \mathbb{R}^n$ with coefficients in S and PIVP functions $q_1, \ldots, q_n$ with parameters in S such that y is the solution of $y' = p(y, t)$ and $y(0) = (q_1(x), \ldots, q_n(x))$, where x is some $x_1, \ldots, x_k \in \mathbb{R}$.



(A) approximating C($\mathbb{R}$): two initial conditions x and $\eta \approx n$

Lemma (Bournez, Campagnolo, Graça and Hainry, 2007)

Let $f : [0,1] \to \mathbb{R}$ be a computable function. Then there exists a PIVP function $y(t; x, \eta)$ with parameters in $\mathbb{Q}[\pi]$, where $x \in [0,1]$ and $|\eta - n| \leq \varepsilon < 1/2$ ($n \in \mathbb{N}$), and some $T \geq 0$ s.t.

$$|y_1(t; x, \eta) - f(x)| \leq 2^{-n} \text{ for all } t \geq T.$$

(A) approximating C($\mathbb{R}$): two initial conditions x and $\eta \approx n$

Lemma (Bournez, Campagnolo, Graça and Hainry, 2007)

Let $f : [0,1] \to \mathbb{R}$ be a computable function. Then there exists a PIVP function $y(t; x, \eta)$ with parameters in $\mathbb{Q}[\pi]$, where $x \in [0,1]$ and $|\eta - n| \leq \varepsilon < 1/2$ ($n \in \mathbb{N}$), and some $T \geq 0$ s.t.

$$|y_1(t; x, \eta) - f(x)| \leq 2^{-n} \text{ for all } t \geq T.$$

Sketch of the proof (i): a "cascade" of ODEs

ODE	initial conditions depend on	value after T_i
$y' = p_1(y,t)$	x, η	$y_1(t; x, \eta) \approx 2^n$ $y_2(t; x, \eta) \approx x\, 2^{m(n)}$
$y' = p_2(y,t)$	$x_1 \approx 2^n$, $x_2 \approx x\, 2^{m(n)}$	$y_3(t; x_1, x_2) \approx$ $\mathrm{abs}(x\, 2^{m(n)}, m(n), n)$
$y' = p_3(y,t)$	$x_3 \approx \mathrm{abs}(x\, 2^{m(n)}, m(n), n)$ $x_4 \approx 2^n$	$y_4 \approx f(x)$

Switching dynamics

Sketch of the proof (ii): define $y' = f(y,t)$ where:

$$f(y,t) = \begin{cases} p_1(y,t) & , \quad t < T_1 \\ p_2(y,t) & , \quad T_1 + \delta < t < T_1 + T_2 \\ p_3(y,t) & , \quad T_1 + T_2 + \delta < t < T_1 + T_2 + T_3. \end{cases}$$

Switching dynamics

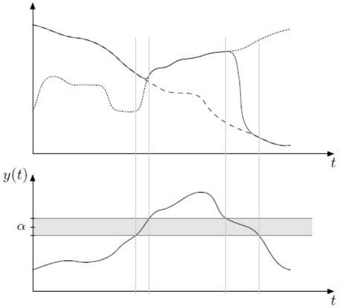
Sketch of the proof (ii): define $y' = f(y,t)$ where:

$$f(y,t) = \begin{cases} p_1(y,t) & , \quad t < T_1 \\ p_2(y,t) & , \quad T_1 + \delta < t < T_1 + T_2 \\ p_3(y,t) & , \quad T_1 + T_2 + \delta < t < T_1 + T_2 + T_3. \end{cases}$$

Given f_1 and f_2 (dotted lines) define some $f = \phi_{\mathrm{y}}^1 f_1 + \phi_{\mathrm{y}}^1 f_2$ (solid line) such that

$$|f(t) - f_1(t)| \leq \varepsilon \text{ if } \mathrm{y}(t) \leq \alpha - 1/4$$
$$|f(t) - f_2(t)| \leq \varepsilon \text{ if } \mathrm{y}(t) \geq \alpha + 1/4.$$

$\mathrm{y}(t)$ is the control function:



(B) approximating C($\mathbb{R}$): one initial condition x.

Theorem (Bournez, Campagnolo, Graça and Hainry, 2007)

If $f : [0, 1] \to \mathbb{R}$ is computable, then there is a PIVP function $y(t; x)$ with parameters in $\mathbb{Q}[\pi]$ such that:

1. $\lim_{t\to\infty} y_2(t; x) = 0$;
2. *for $x \in [0, 1]$, and $t \in [0, +\infty)$, $|y_1(t; x) - f(x)| \leq y_2(t; x)$.*

(B) approximating $C(\mathbb{R})$: one initial condition x.

Theorem (Bournez, Campagnolo, Graça and Hainry, 2007)

If $f : [0,1] \to \mathbb{R}$ is computable, then there is a PIVP function $y(t;x)$ with parameters in $\mathbb{Q}[\pi]$ such that:

1. $\lim_{t\to\infty} y_2(t;x) = 0$;
2. *for $x \in [0,1]$, and $t \in [0,+\infty)$, $|y_1(t;x) - f(x)| \leq y_2(t;x)$.*

Sketch of the proof.

Use the PIVP of the previous lemma with initial condition $x \in [0,1]$ and replace initial condition η by a component y_3 with:

- $y_3 \approx 1$ for $0 < t < T_1$, $y_3 \approx 2$ for $T_1 + \delta < t < T_2$, ..., depending on y_4;
- $y_4 \approx$ the state of a TM that indicates that $|y_1(t;x) - f(x)| \leq 2^{-n}$.

$\square$

Characterization of C($\mathbb{R}$)

Definition (LIM^*)

Let C be a class of functions over $\mathbb{R}$. LIM^* is an operation which takes $f_1, f_2 \in C$, such that $\lim_{t\to\infty} f_2(t,x) = 0$, and returns $f(x) = \lim_{t\to+\infty} f_1(t,x)$ if $|f_1(t,x) - f(x)| \leq f_2(t,x)$ for positive t. $C(\mathrm{LIM}^*)$ is the closure of C under LIM^*.

Characterization of $\mathbf{C}(\mathbb{R})$

Definition (LIM^*)

Let C be a class of functions over $\mathbb{R}$. LIM^* is an operation which takes $f_1, f_2 \in C$, such that $\lim_{t\to\infty} f_2(t,x) = 0$, and returns $f(x) = \lim_{t\to+\infty} f_1(t,x)$ if $|f_1(t,x) - f(x)| \leq f_2(t,x)$ for positive t. $C(\mathrm{LIM}^*)$ is the closure of C under LIM^*.

Lemma (Campagnolo and Ojakian 2007)

$\mathbf{C}(\mathbb{R}) = \mathbf{C}(\mathbb{R})(\mathrm{LIM}^*)$.

Theorem (Reformulation of the main result of Bournez, Campagnolo, Graça and Hainry 2007)

On $[0,1]$, $\mathbf{C}(\mathbb{R}) = \mathrm{PIVP}_{Q[\pi]}(\mathrm{LIM}^*)$.

Summary and questions (part 2)

- Real computable functions can be approximated by PIVP functions on compact intervals, which is a further evidence of their computational completeness.

Summary and questions (part 2)

- Real computable functions can be approximated by PIVP functions on compact intervals, which is a further evidence of their computational completeness.
- Is the result true for unbounded intervals?

Summary and questions (part 2)

- Real computable functions can be approximated by PIVP functions on compact intervals, which is a further evidence of their computational completeness.
- Is the result true for unbounded intervals?
- Can we replace LIM^* by a more natural limit with the approximation requirement $|f_1(t,x) - f(x)| \leq \frac{1}{t}$?

Summary and questions (part 2)

- Real computable functions can be approximated by PIVP functions on compact intervals, which is a further evidence of their computational completeness.
- Is the result true for unbounded intervals?
- Can we replace LIM^* by a more natural limit with the approximation requirement $|f_1(t,x) - f(x)| \leq \frac{1}{t}$?
- Can we replace the set of parameters $\mathbb{Q}[\pi]$ by $\mathbb{Q}$?

Some references

- D.S. Graça, J. Buescu, and M.L. Campagnolo. Computational bounds on polynomial differential equations, Applied Mathematics and Computation, 215 (2009) 1375:1385.
- D.S. Graça, M.L. Campagnolo, and J. Buescu. Computability with polynomial differential equations, Advances in Applied Mathematics, 40, (2008) 330:349.
- O. Bournez, M. L. Campagnolo, D. S. Graça, and E. Hainry. Polynomial differential equations compute all real computable functions on computable compact intervals. Journal of Complexity, 23:317–335, 2007.
- M. L. Campagnolo and K. Ojakian. Using Approximation to Relate Computational Classes over the Reals. In J. Durand-Lose and M. Margenstern (Eds.): MCU 2007, Lecture Notes in Computer Science, 4664, (2007) 39:61.