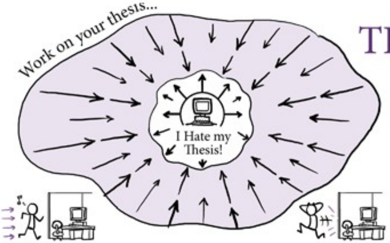
Dynamical Systems: Computability, Verification, Analysis

Amaury Pouly

Journée des nouveaux arrivants, IRIF

15 october 2018

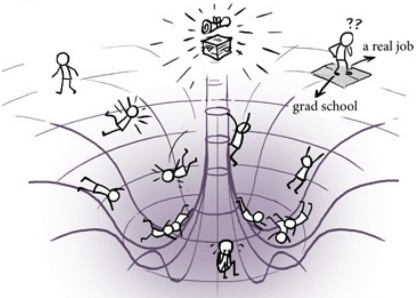
THE THESIS REPULSOR FIELD



The Thesis Repulsor Field (TRF) is a generalized model of the forces experienced by an individual in the final stages of graduate space-time*.

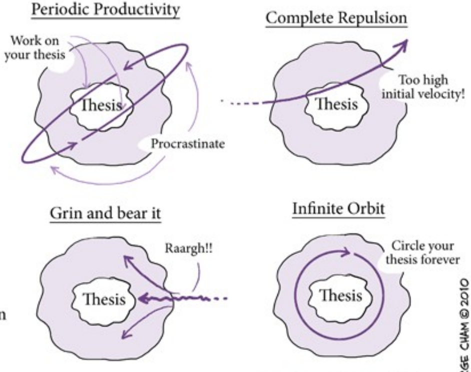
It is characterized by an attractor vector field directed towards completion of the thesis but with an intense repulsive singularity at its origin.

The resulting potential well of wasted potential acts as grad students follow the gradient in the perceived direction of least work:



In reality, TRF is not an actual force, but rather a distortion of the mindspace continuum, in which grad students are simply responding to the curvature of their own neuroses.

Several trajectories are possible due to this vector field:



*Graduate space-time is just like real space-time, but with added imaginary dimensions.

Trajectory

2011 Master : ENS Lyon

2015 PhD : LIX, Polytechnique and University of Algarve, Portugal
Olivier Bournez and Daniel S. Graça

2016 Postdoc : Oxford
Joel Ouaknine and James Worrell

2017 Postdoc : Max Planck Institute for Software Systems
Joel Ouaknine

Trajectory

2011 Master : ENS Lyon

2015 PhD : LIX, Polytechnique and University of Algarve, Portugal
Olivier Bournez and Daniel S. Graça

2016 Postdoc : Oxford
Joel Ouaknine and James Worrell

2017 Postdoc : Max Planck Institute for Software Systems
Joel Ouaknine

2018 Attracted to IRIF : starting 1st January

Research Interests

Verification

$\left\{\begin{array}{l}\text{Decidability}\\ \text{Complexity}\end{array}\right\}$ questions in **Dynamical Systems** with continuous space

Models of Computation

Computable Analysis

Research Interests

Verification

$\left\{\begin{array}{l}\text{Decidability}\\ \text{Complexity}\end{array}\right\}$ questions in **Dynamical Systems** with continuous space

$\boxed{\text{Models of Computation}}$

Computable Analysis

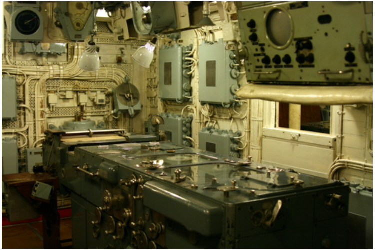
What is a computer ?

What is a computer ?

What is a computer ?

vs

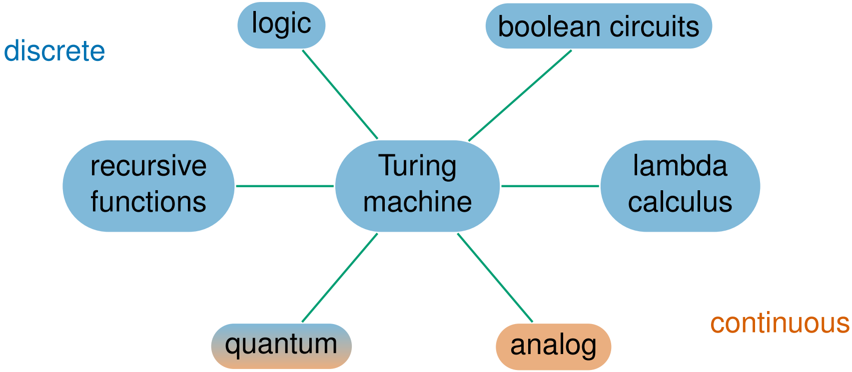
Differential Analyser
"Mathematica of the 1920s"



Admiralty Fire Control Table
British Navy ships (WW2)

Church Thesis

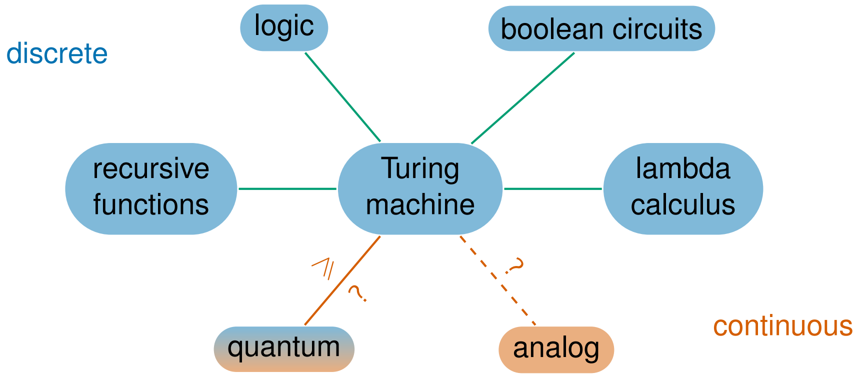
Computability



Church Thesis

All **reasonable** models of computation are equivalent.

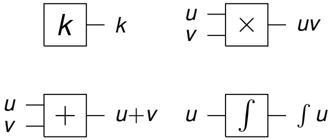
Complexity



Effective Church Thesis

All **reasonable** models of computation are equivalent for complexity.

Polynomial Differential Equations



General Purpose
Analog Computer

Differential Analyzer

Newton mechanics

Reaction networks :

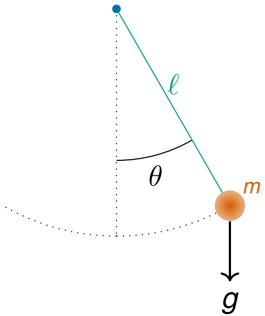
- chemical
- enzymatic

polynomial differential equations :

$$\begin{cases} y(0) = y_0 \\ y'(t) = p(y(t)) \end{cases}$$

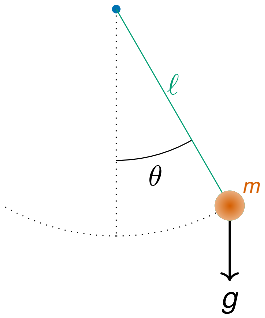
- Rich class
- Stable (+, $\times$, $\circ$, /, ED)
- No closed-form solution

Example of dynamical system



$\ddot{\theta} + \frac{g}{\ell}\sin(\theta) = 0$

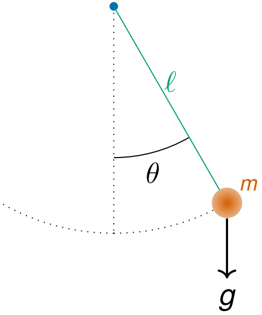
Example of dynamical system

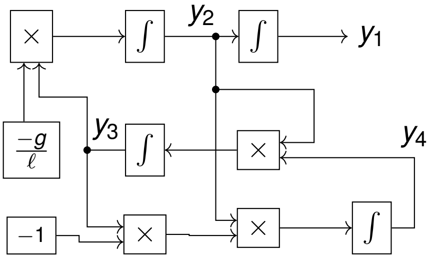


$$\ddot{\theta} + \frac{g}{\ell}\sin(\theta) = 0$$

$$\begin{cases} y_1' = y_2 \\ y_2' = -\frac{g}{l} y_3 \\ y_3' = y_2 y_4 \\ y_4' = -y_2 y_3 \end{cases} \Leftrightarrow \begin{cases} y_1 = \theta \\ y_2 = \dot{\theta} \\ y_3 = \sin(\theta) \\ y_4 = \cos(\theta) \end{cases}$$

Example of dynamical system





$$\ddot{\theta} + \frac{g}{\ell}\sin(\theta) = 0$$

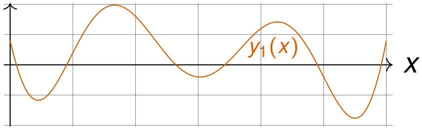
$$\begin{cases} y_1' = y_2 \\ y_2' = -\frac{g}{l} y_3 \\ y_3' = y_2 y_4 \\ y_4' = -y_2 y_3 \end{cases} \Leftrightarrow \begin{cases} y_1 = \theta \\ y_2 = \dot{\theta} \\ y_3 = \sin(\theta) \\ y_4 = \cos(\theta) \end{cases}$$

Computing with differential equations

Generable functions

$$\begin{cases} y(0) = y_0 \\ y'(x) = p(y(x)) \end{cases} \qquad x \in \mathbb{R}$$

$$f(x) = y_1(x)$$



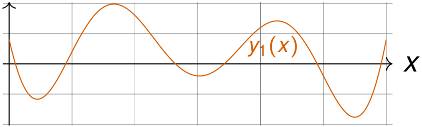
Shannon's notion

Computing with differential equations

Generable functions

$$\begin{cases} y(0) = y_0 \\ y'(x) = p(y(x)) \end{cases} \qquad x \in \mathbb{R}$$

$f(x) = y_1(x)$



Shannon's notion

$\sin, \cos, \exp, \log, \ldots$

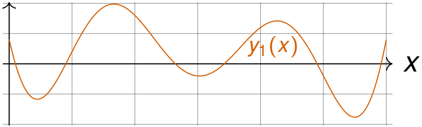
Strictly weaker than Turing machines [Shannon, 1941]

Computing with differential equations

Generable functions

$$\begin{cases} y(0) = y_0 \\ y'(x) = p(y(x)) \end{cases} \qquad x \in \mathbb{R}$$

$$f(x) = y_1(x)$$



Shannon's notion

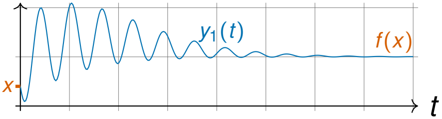
$\sin, \cos, \exp, \log, \ldots$

Strictly weaker than Turing machines [Shannon, 1941]

Computable

$$\begin{cases} y(0) = q(x) & x \in \mathbb{R} \\ y'(t) = p(y(t)) & t \in \mathbb{R}_+ \end{cases}$$

$$f(x) = \lim_{t \to \infty} y_1(t)$$



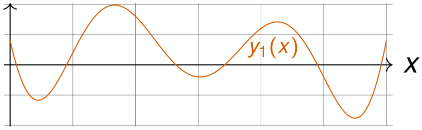
Modern notion

Computing with differential equations

Generable functions

$$\begin{cases} y(0) = y_0 \\ y'(x) = p(y(x)) \end{cases} \qquad x \in \mathbb{R}$$

$$f(x) = y_1(x)$$



Shannon's notion

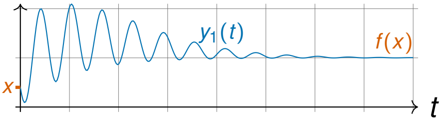
$\sin, \cos, \exp, \log, \ldots$

Strictly weaker than Turing machines [Shannon, 1941]

Computable

$$\begin{cases} y(0) = q(x) & x \in \mathbb{R} \\ y'(t) = p(y(t)) & t \in \mathbb{R}_+ \end{cases}$$

$$f(x) = \lim_{t \to \infty} y_1(t)$$

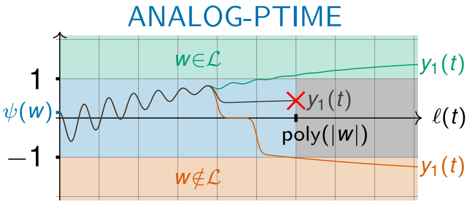


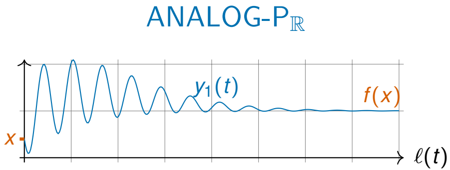
Modern notion

$\sin, \cos, \exp, \log, \Gamma, \zeta, \ldots$

Turing powerful [Bournez et al., 2007]

Highlights of some results





Theorem

- PTIME = ANALOG-PTIME
- $f : [a, b] \to \mathbb{R}$ *computable in polynomial time* $\Leftrightarrow f \in$ ANALOG-$P_{\mathbb{R}}$

- Analog complexity theory based on length
- Time of Turing machine $\Leftrightarrow$ length of the GPAC
- Purely continuous characterization of PTIME

Research Interests

Verification

$\left\{\begin{array}{l}\text{Decidability}\\ \text{Complexity}\end{array}\right\}$ questions in **Dynamical Systems** with continuous space

Models of Computation

Computable Analysis

A word on computability for real functions

Classical computability (Turing machine) : compute on words, integers, rationals, ...

A word on computability for real functions

Classical computability (Turing machine) : compute on words, integers, rationals, ...

Real computability :

A word on computability for real functions

Classical computability (Turing machine) : compute on words, integers, rationals, **...**

Real computability :at least *two different notions*

- BSS (Blum-Shub-Smale) machine : register machine that can store arbitrary real numbers and that can compute rational functions over reals at unit cost**. Comparisons between reals are allowed.**

A word on computability for real functions

Classical computability (Turing machine) : compute on words, integers, rationals, ...

Real computability :at least *two different notions*

- BSS (Blum-Shub-Smale) machine : register machine that can store arbitrary real numbers and that can compute rational functions over reals at unit cost. **Comparisons between reals are allowed.**
- Computable Analysis : reals are represented as converging Cauchy sequences, computations are carried out by rational approximations using Turing machines. **Comparisons between reals is not decidable in general.** Computable implies continuous.

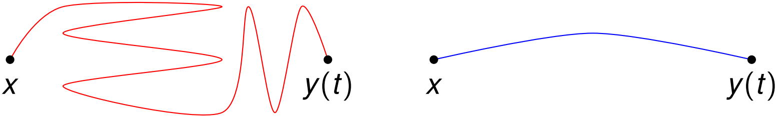
Computable Analysis : differential equations

Let $f : \mathbb{R}^n \to \mathbb{R}^n$ continuous, consider

$$y(0) = x, \qquad y' = f(y) \tag{1}$$

Question

When is y computable ? What about its complexity ?



Computable Analysis : differential equations

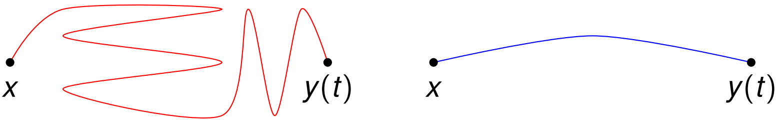
Let $f : \mathbb{R}^n \to \mathbb{R}^n$ continuous, consider

$$y(0) = x, \qquad y' = f(y) \tag{1}$$

It can be very bad :

Theorem (Pour-El and Richards)

There exists a computable (hence continuous) f such that ***none of the solutions*** *to* (1) *is computable.*



Computable Analysis : differential equations

Let $f : \mathbb{R}^n \to \mathbb{R}^n$ continuous, consider

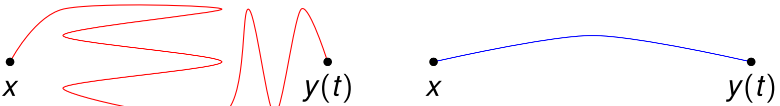
$$y(0) = x, \qquad y' = f(y) \tag{1}$$

Some good news :

Theorem (Ruohonen)

If f is computable and (1) has a unique solution, then it is computable.

But complexity can be unbounded



Computable Analysis : differential equations

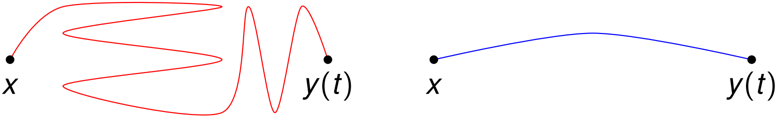
Let $f : \mathbb{R}^n \to \mathbb{R}^n$ continuous, consider

$$y(0) = x, \qquad y' = f(y) \tag{1}$$

Still things are bad :

Theorem (Buescu, Campagnolo and Graça)

Computing the maximum interval of life (or deciding if it is bounded) is undecidable, even if f is a polynomial.



Computable Analysis : differential equations

Let $f : \mathbb{R}^n \to \mathbb{R}^n$ continuous, consider

$$y(0) = x, \qquad y' = f(y) \tag{1}$$

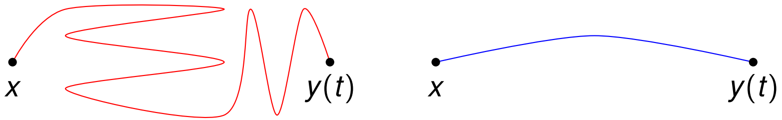
A new hope :

Theorem

If $y(t)$ exists, we can compute $r \in \mathbb{Q}$ such $|r - y(t)| \leqslant 2^{-n}$ in time

$$\operatorname{poly}(\textit{size of } x \textit{ and } p, n, \ell(t))$$

where $\ell(t) \approx$ length of the curve y (between x and $y(t)$)



Research Interests

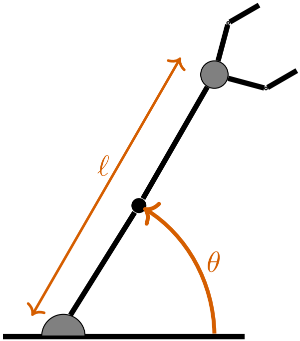
Verification

$\left\{\begin{matrix}\text{Decidability}\\ \text{Complexity}\end{matrix}\right\}$ questions in **Dynamical Systems** with continuous space

Models of Computation

Computable Analysis

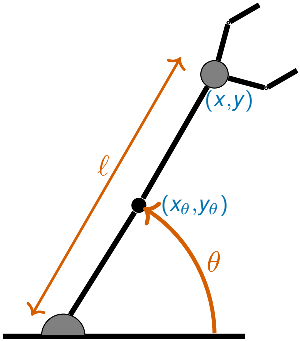
Example : 2D robot



Available actions :

- rotate arm
- change arm length

Example : 2D robot

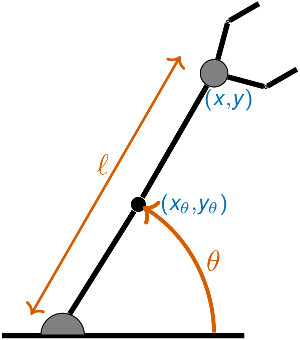


State : $X = (x_\theta, y_\theta, x, y) \in \mathbb{R}^4$

Available actions :

- rotate arm
- change arm length

Example : 2D robot



State : $X = (x_\theta, y_\theta, x, y) \in \mathbb{R}^4$

Rotate arm (increase θ) :

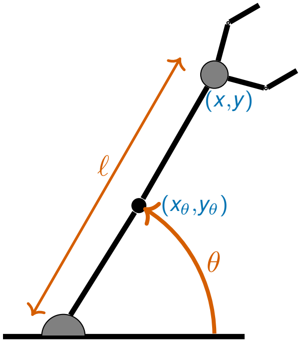
$$\begin{bmatrix} x \\ y \end{bmatrix}' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x_\theta \\ y_\theta \end{bmatrix}' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_\theta \\ y_\theta \end{bmatrix}$$

Available actions :

- rotate arm
- change arm length

Example : 2D robot



Available actions :

- rotate arm
- change arm length

State : $X = (x_\theta, y_\theta, x, y) \in \mathbb{R}^4$

Rotate arm (increase θ) :

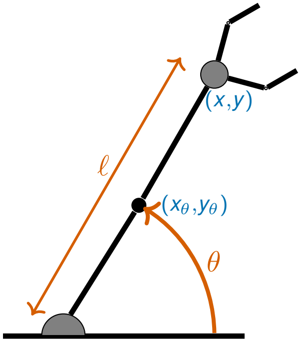
$$\begin{bmatrix} x \\ y \end{bmatrix}' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x_\theta \\ y_\theta \end{bmatrix}' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_\theta \\ y_\theta \end{bmatrix}$$

Change arm length (increase ℓ) :

$$\begin{bmatrix} x \\ y \end{bmatrix}' = \begin{bmatrix} x_\theta \\ y_\theta \end{bmatrix}$$

Example : 2D robot



Available actions :

- rotate arm
- change arm length

$\rightarrow$ Switched linear system :

$$X' = AX$$

where $A \in \{A_{rot}, A_{arm}\}$.

State : $X = (x_\theta, y_\theta, x, y) \in \mathbb{R}^4$

Rotate arm (increase θ) :

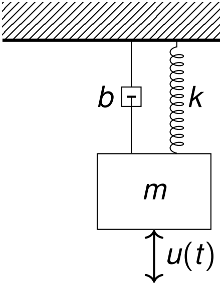
$$\begin{bmatrix} x \\ y \end{bmatrix}' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x_\theta \\ y_\theta \end{bmatrix}' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_\theta \\ y_\theta \end{bmatrix}$$

Change arm length (increase ℓ) :

$$\begin{bmatrix} x \\ y \end{bmatrix}' = \begin{bmatrix} x_\theta \\ y_\theta \end{bmatrix}$$

Example : mass-spring-damper system



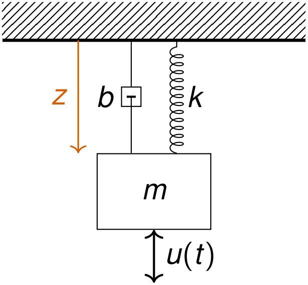
Model with external input $u(t)$

State : $X = z \in \mathbb{R}$

Equation of motion :

$$mz'' = -kz - bz' + mg + u$$

Example : mass-spring-damper system



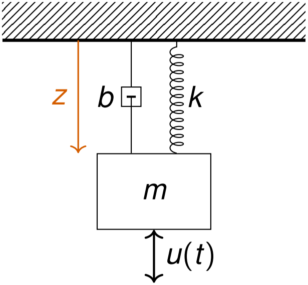
Model with external input $u(t)$

State : $X = z \in \mathbb{R}$

Equation of motion :

$$mz'' = -kz - bz' + mg + u$$

Example : mass-spring-damper system



Model with external input $u(t)$

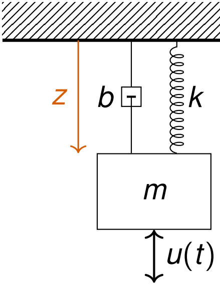
State : $X = z \in \mathbb{R}$

Equation of motion :

$$mz'' = -kz - bz' + mg + u$$

$\rightarrow$ Affine but not first order

Example : mass-spring-damper system



Model with external input $u(t)$

State : $X = z \in \mathbb{R}$

Equation of motion :

$$mz'' = -kz - bz' + mg + u$$

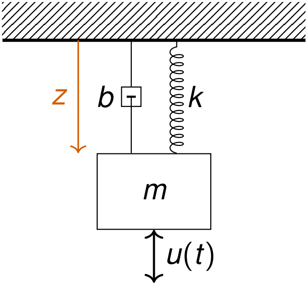
$\rightarrow$ Affine but not first order

State : $X = (z, z', 1) \in \mathbb{R}^3$

Equation of motion :

$$\begin{bmatrix} z \\ z' \\ 1 \end{bmatrix}' = \begin{bmatrix} z' \\ -\frac{k}{m}z - \frac{b}{m}z' + g + \frac{1}{m}u \\ 0 \end{bmatrix}$$

Example : mass-spring-damper system



Model with external input $u(t)$

$\rightsquigarrow$ Linear time invariant system :

$$X' = AX + Bu$$

with some constraints on u.

State : $X = z \in \mathbb{R}$

Equation of motion :

$$mz'' = -kz - bz' + mg + u$$

$\rightarrow$ Affine but not first order

State : $X = (z, z', 1) \in \mathbb{R}^3$

Equation of motion :

$$\begin{bmatrix} z \\ z' \\ 1 \end{bmatrix}' = \begin{bmatrix} z' \\ -\frac{k}{m}z - \frac{b}{m}z' + g + \frac{1}{m}u \\ 0 \end{bmatrix}$$

Linear dynamical systems

Discrete case

$$x(n+1) = Ax(n)$$

- biology,
- software verification,
- probabilistic model checking,
- combinatorics,
- ....

Continuous case

$$x'(t) = Ax(t)$$

- biology,
- physics,
- probabilistic model checking,
- electrical circuits,
- ....

Typical questions

- reachability : does the trajectory reach some states ?
- safety : does it always avoid the bad(unsafe) states ?

Linear dynamical systems

Discrete case

$$x(n+1) = Ax(n) + Bu(n)$$

- biology,
- software verification,
- probabilistic model checking,
- combinatorics,
- ....

Continuous case

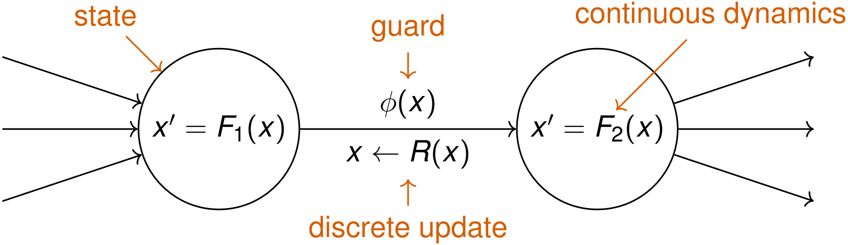
$$x'(t) = Ax(t) + Bu(t)$$

- biology,
- physics,
- probabilistic model checking,
- electrical circuits,
- ....

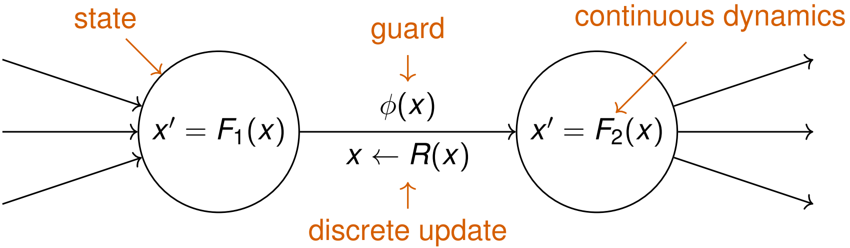
Typical questions

- reachability : does the trajectory reach some states ?
- safety : does it always avoid the bad(unsafe) states ?
- controllability : can we control it to some state ?

Hybrid/Cyber-physical systems

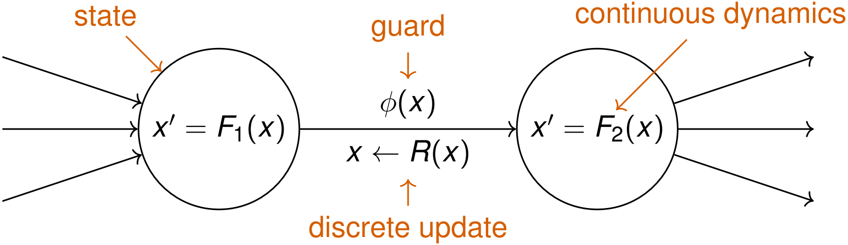


Hybrid/Cyber-physical systems



- $F_i(x) = 1$: timed automata
- $F_i(x) = c_i$: rectangular hybrid automata
- $F_i(x) = A_i x$: linear hybrid automata

Hybrid/Cyber-physical systems



- $F_i(x) = 1$: timed automata
- $F_i(x) = c_i$: rectangular hybrid automata
- $F_i(x) = A_i x$: linear hybrid automata

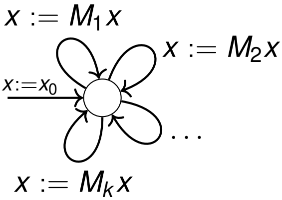
Typical question

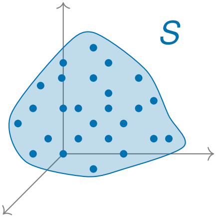
Verify some temporal specification :

$$\mathbb{G}(P_1 \Rightarrow (P_2 \mathbb{U} P_3))$$

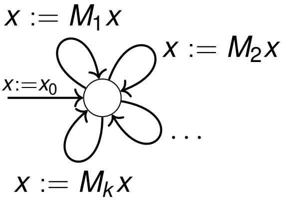
"When the trajectory enters P_1, it must remain within P_2 until it reaches P_3"

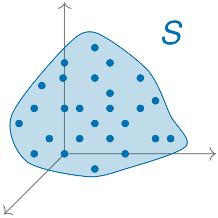
Exact verification is unfeasible





Exact verification is unfeasible



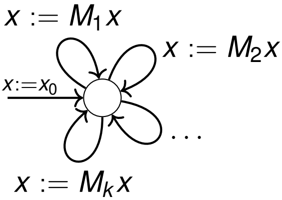


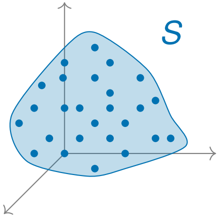
Theorem (Markov 1947 [1])

There is a *fixed set* *of* 6×6 *integer matrices* $M_1, \dots, M_k$ *such that the reachability problem "*y *is reachable from* x_0 *?" is* *undecidable*.

1. Original theorems about semigroups, reformulated with hybrid systems.

Exact verification is unfeasible





Theorem (Markov 1947 [1])

There is a fixed set of 6×6 *integer matrices* $M_1, \ldots, M_k$ *such that the reachability problem "y is reachable from x_0 ?" is undecidable.*

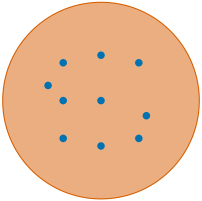
Theorem (Paterson 1970 [1])

The mortality problem "0 is reachable from x_0 with $M_1, \ldots, M_k$?" is undecidable for 3×3 *matrices.*

1. Original theorems about semigroups, reformulated with hybrid systems.

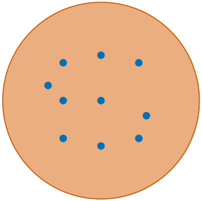
Invariants

invariant = overapproximation of the reachable states

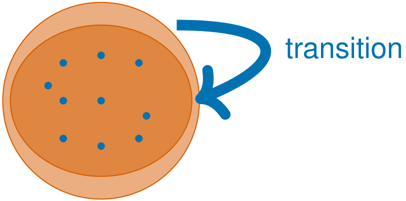


Invariants

invariant = overapproximation of the reachable states

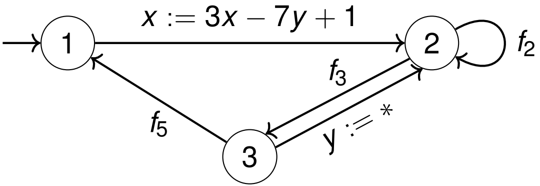


inductive invariant = invariant preserved by the transition relation



Invariants : example result

affine program :
nondeterministic branching, no guards, affine assignments



Theorem

There is an algorithm which computes, for any given affine program over $\mathbb{Q}$, its strongest polynomial inductive invariant.

Verification

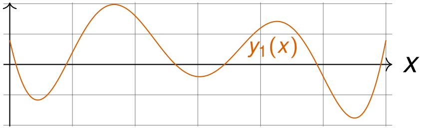
$\left\{\begin{array}{l}\text{Decidability}\\ \text{Complexity}\end{array}\right\}$ questions in **Dynamical Systems** with continuous space

Models of Computation

Computable Analysis

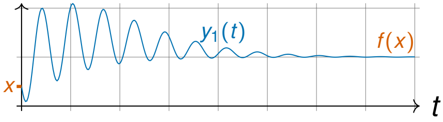
Universal differential equations

Generable functions



subclass of analytic functions

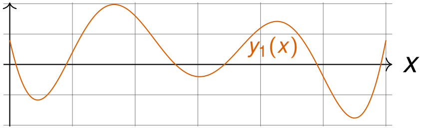
Computable functions



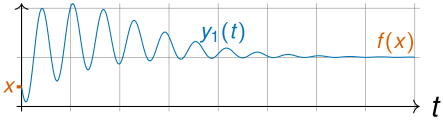
any computable function

Universal differential equations

Generable functions

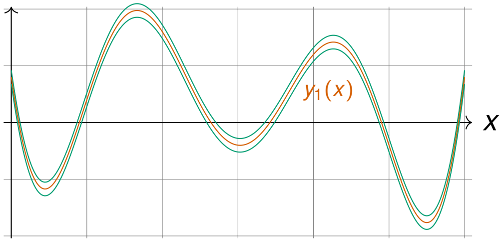


Computable functions

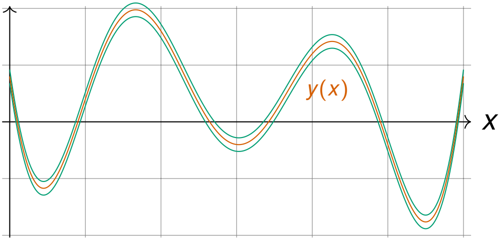


subclass of analytic functions

any computable function



Universal differential algebraic equation (DAE)



Theorem (Rubel, 1981)

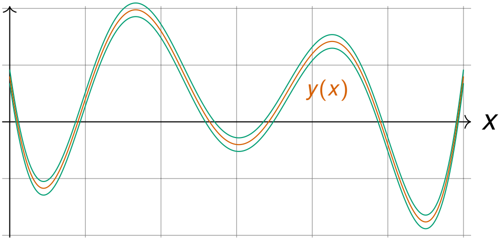
For any continuous functions f and ε, there exists $y:\mathbb{R}\to\mathbb{R}$ solution to

$$\begin{aligned} 3y'^4y''y''''^2 & -4y'^4y'''^2y'''' + 6y'^3y''^2y'''y'''' + 24y'^2y''^4y'''' \\ & -12y'^3y''y'''^3 - 29y'^2y''^3y'''^2 + 12y''^7 && = 0 \end{aligned}$$

such that $\forall t\in\mathbb{R}$,

$$|y(t)-f(t)|\leqslant\varepsilon(t).$$

Universal differential algebraic equation (DAE)



Theorem (Rubel, 1981)

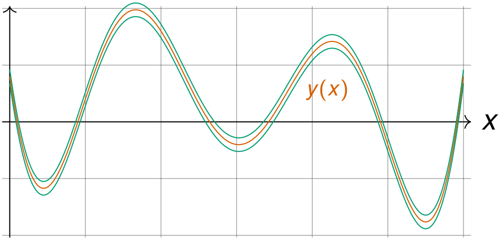
There exists a ***fixed*** *polynomial p and $k \in \mathbb{N}$ such that for any continuous functions f and ε, there exists a solution $y : \mathbb{R} \to \mathbb{R}$ to*

$$p(y, y', \dots, y^{(k)}) = 0$$

such that $\forall t \in \mathbb{R}$,

$$|y(t) - f(t)| \leqslant \varepsilon(t).$$

Universal differential algebraic equation (DAE)



Theorem (Rubel, 1981)

*There exists a **fixed** polynomial p and $k \in \mathbb{N}$ such that for any continuous functions f and ε, there exists a solution $y : \mathbb{R} \to \mathbb{R}$ to*

$$p(y, y', \dots, y^{(k)}) = 0$$

such that $\forall t \in \mathbb{R}$,

$$|y(t) - f(t)| \leqslant \varepsilon(t).$$

Problem : this is «weak» result.

The problem with Rubel's DAE

The solution y is not unique, **even with added initial conditions** :

$$p(y, y', \ldots, y^{(k)}) = 0, \quad y(0) = \alpha_0, y'(0) = \alpha_1, \ldots, y^{(k)}(0) = \alpha_k$$

In fact, this is fundamental for Rubel's proof to work !

The problem with Rubel's DAE

The solution y is not unique, **even with added initial conditions** :

$$p(y, y', \ldots, y^{(k)}) = 0, \quad y(0) = \alpha_0, y'(0) = \alpha_1, \ldots, y^{(k)}(0) = \alpha_k$$

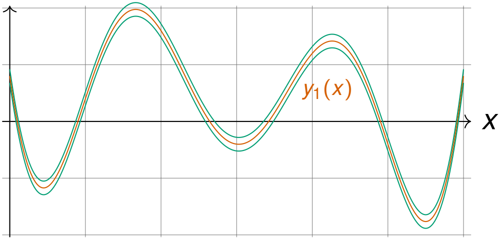
In fact, this is fundamental for Rubel's proof to work !

- Rubel's statement : this DAE is universal
- More realistic interpretation : this DAE allows almost anything

Open Problem (Rubel, 1981)

Is there a universal ODE $y' = p(y)$?
Note : explicit polynomial ODE $\Rightarrow$ unique solution

Universal initial value problem (IVP)



Theorem

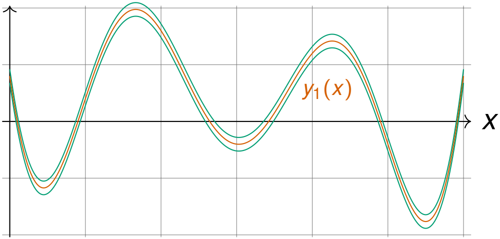
*There exists a **fixed** (vector of) polynomial p such that for any continuous functions f and ε, there exists $\alpha \in \mathbb{R}^d$ such that*

$$y(0) = \alpha, \qquad y'(t) = p(y(t))$$

*has a **unique solution** $y : \mathbb{R} \to \mathbb{R}^d$ and $\forall t \in \mathbb{R}$,*

$$|y_1(t) - f(t)| \leqslant \varepsilon(t).$$

Universal initial value problem (IVP)



Notes :

- **system** of ODEs,
- y is analytic,
- we need $d \approx 300$.

Theorem

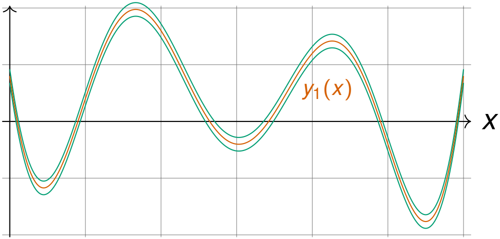
*There exists a **fixed** (vector of) polynomial p such that for any continuous functions f and ε, there exists $\alpha \in \mathbb{R}^d$ such that*

$$y(0) = \alpha, \qquad y'(t) = p(y(t))$$

*has a **unique solution** $y : \mathbb{R} \to \mathbb{R}^d$ and $\forall t \in \mathbb{R}$,*

$$|y_1(t) - f(t)| \leqslant \varepsilon(t).$$

Universal initial value problem (IVP)



Notes :

- **system** of ODEs,
- y is analytic,
- we need $d \approx 300$.

Theorem

*There exists a **fixed** (vector of) polynomial p such that for any continuous functions f and ε, there exists $\alpha \in \mathbb{R}^d$ such that*

$$y(0) = \alpha, \qquad y'(t) = p(y(t))$$

*has a **unique solution** $y : \mathbb{R} \to \mathbb{R}^d$ and $\forall t \in \mathbb{R}$,*

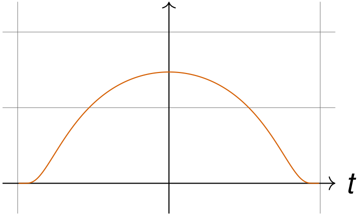
$$|y_1(t) - f(t)| \leqslant \varepsilon(t).$$

Remark : α is usually transcendental, but computable from f and ε

Rubel's proof in one slide

- Take $f(t) = e^{\frac{-1}{1-t^2}}$ for $-1 < t < 1$ and $f(t) = 0$ otherwise.

It satisfies $\quad (1-t^2)^2 f''(t) + 2tf'(t) = 0.$



Rubel's proof in one slide

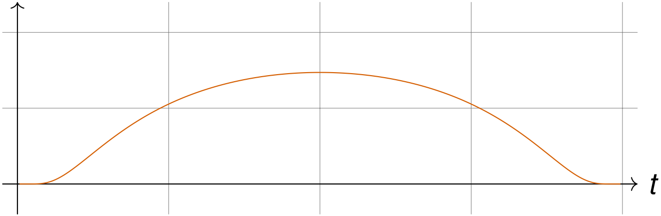
- Take $f(t) = e^{\frac{-1}{1-t^2}}$ for $-1 < t < 1$ and $f(t) = 0$ otherwise.

$$\text{It satisfies} \quad (1-t^2)^2 f''(t) + 2tf'(t) = 0.$$

- For any $a, b, c \in \mathbb{R}$, $y(t) = cf(at+b)$ satisfies

$$\begin{aligned} 3y'^4 y'' y''''^2 \quad & -4y'^4 y''^2 y'''' + 6y'^3 y''^2 y''' y'''' + 24y'^2 y''^4 y'''' \\ & -12y'^3 y'' y'''^3 - 29y'^2 y''^3 y'''^2 + 12y''^7 = 0 \end{aligned}$$

Translation and rescaling :



Rubel's proof in one slide

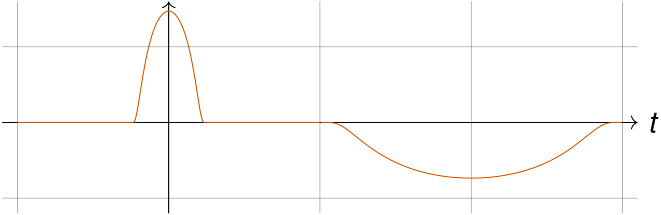
- Take $f(t) = e^{\frac{-1}{1-t^2}}$ for $-1 < t < 1$ and $f(t) = 0$ otherwise.

  $$\text{It satisfies} \quad (1-t^2)^2 f''(t) + 2tf'(t) = 0.$$

- For any $a, b, c \in \mathbb{R}$, $y(t) = cf(at+b)$ satisfies

  $$3y'^4y''y''''^2 - 4y'^4y''^2y'''' + 6y'^3y''^2y'''y'''' + 24y'^2y''^4y'''' - 12y'^3y''y'''^3 - 29y'^2y''^3y'''^2 + 12y''^7 = 0$$

- Can glue together arbitrary many such pieces

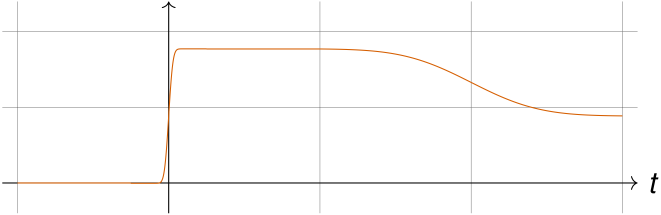


Rubel's proof in one slide

- Take $f(t) = e^{\frac{-1}{1-t^2}}$ for $-1 < t < 1$ and $f(t) = 0$ otherwise.

  It satisfies $\quad (1-t^2)^2 f''(t) + 2tf'(t) = 0.$
- For any $a, b, c \in \mathbb{R}$, $y(t) = cf(at+b)$ satisfies

  $$3y'^4y''y''''^2 - 4y'^4y''^2y'''' + 6y'^3y''^2y'''y'''' + 24y'^2y''^4y'''' - 12y'^3y''y'''^3 - 29y'^2y''^3y'''^2 + 12y''^7 = 0$$
- Can glue together arbitrary many such pieces
- Can arrange so that $\int f$ is solution : piecewise pseudo-linear



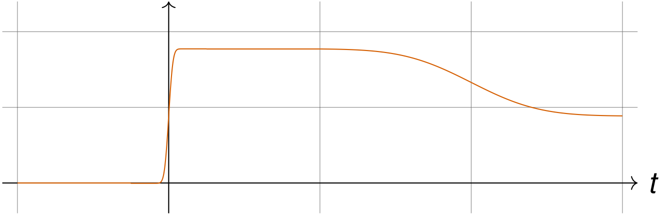
Rubel's proof in one slide

- Take $f(t) = e^{\frac{-1}{1-t^2}}$ for $-1 < t < 1$ and $f(t) = 0$ otherwise.

  It satisfies $\quad (1-t^2)^2 f''(t) + 2tf'(t) = 0.$
- For any $a, b, c \in \mathbb{R}$, $y(t) = cf(at+b)$ satisfies

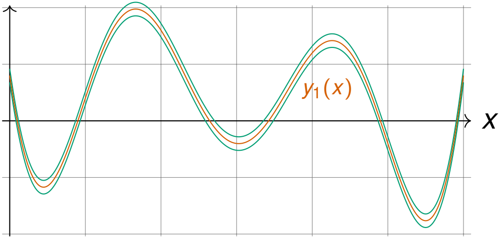
$$3y'^4y''y''''^2 - 4y'^4y''^2y'''' + 6y'^3y''^2y'''y'''' + 24y'^2y''^4y'''' - 12y'^3y''y'''^3 - 29y'^2y''^3y'''^2 + 12y''^7 = 0$$

- Can glue together arbitrary many such pieces
- Can arrange so that $\int f$ is solution : piecewise pseudo-linear



Conclusion : Rubel's equation allows any piecewise pseudo-linear functions, and those are **dense in** C^0

Universal DAE revisited



Theorem

*There exists a **fixed** polynomial p and $k \in \mathbb{N}$ such that for any continuous functions f and ε, there exists $\alpha_0, \dots, \alpha_k \in \mathbb{R}$ such that*

$$p(y, y', \dots, y^{(k)}) = 0, \quad y(0) = \alpha_0, y'(0) = \alpha_1, \dots, y^{(k)}(0) = \alpha_k$$

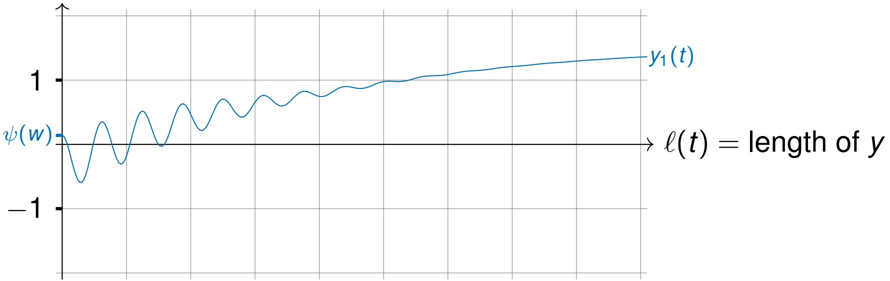
*has a **unique analytic solution** and this solution satisfies such that*

$$|y(t) - f(t)| \leqslant \varepsilon(t).$$

Characterization of polynomial time

Definition : $\mathcal{L} \in$ ANALOG-PTIME $\Leftrightarrow \exists p$ polynomial, $\forall$ word w

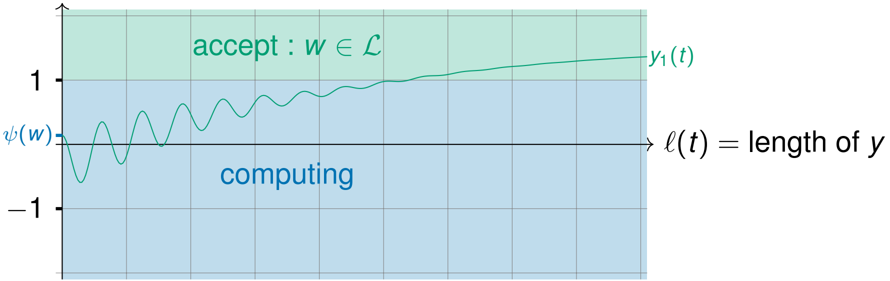
$$y(0) = (\psi(w), |w|, 0, \ldots, 0) \qquad y' = p(y) \qquad \psi(w) = \sum_{i=1}^{|w|} w_i 2^{-i}$$



Characterization of polynomial time

Definition : $\mathcal{L} \in$ ANALOG-PTIME $\Leftrightarrow \exists p$ polynomial, $\forall$ word w

$$y(0) = (\psi(w), |w|, 0, \ldots, 0) \qquad y' = p(y) \qquad \psi(w) = \sum_{i=1}^{|w|} w_i 2^{-i}$$



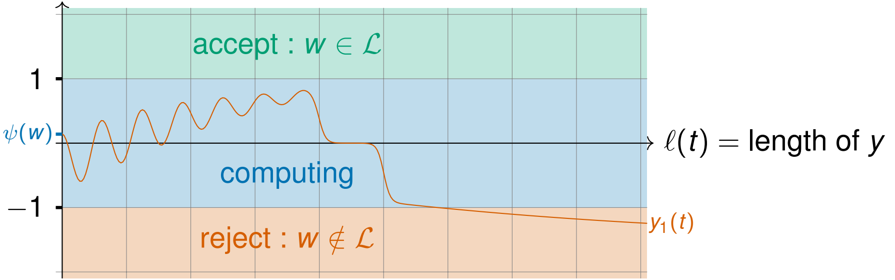
satisfies

1. if $y_1(t) \geqslant 1$ then $w \in \mathcal{L}$

Characterization of polynomial time

Definition : $\mathcal{L} \in$ ANALOG-PTIME $\Leftrightarrow \exists p$ polynomial, $\forall$ word w

$$y(0) = (\psi(w), |w|, 0, \ldots, 0) \qquad y' = p(y) \qquad \psi(w) = \sum_{i=1}^{|w|} w_i 2^{-i}$$



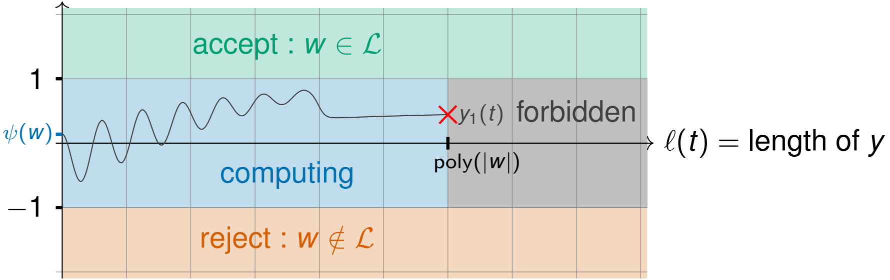
satisfies

2. if $y_1(t) \leqslant -1$ then $w \notin \mathcal{L}$

Characterization of polynomial time

Definition : $\mathcal{L} \in$ ANALOG-PTIME $\Leftrightarrow \exists p$ polynomial, $\forall$ word w

$$y(0) = (\psi(w), |w|, 0, \ldots, 0) \qquad y' = p(y) \qquad \psi(w) = \sum_{i=1}^{|w|} w_i 2^{-i}$$



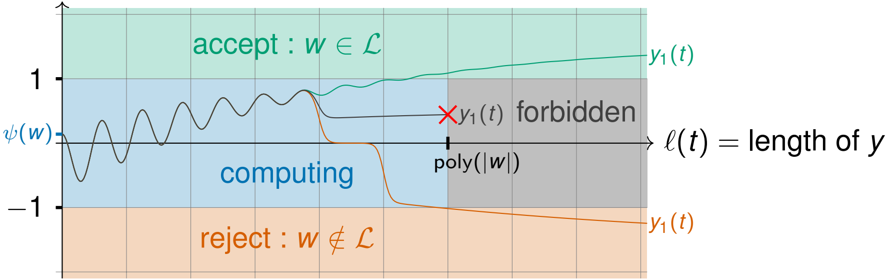
satisfies

3. if $\ell(t) \geqslant \text{poly}(|w|)$ then $|y_1(t)| \geqslant 1$

Characterization of polynomial time

Definition : $\mathcal{L} \in$ ANALOG-PTIME $\Leftrightarrow \exists p$ polynomial, $\forall$ word w

$$y(0) = (\psi(w), |w|, 0, \ldots, 0) \qquad y' = p(y) \qquad \psi(w) = \sum_{i=1}^{|w|} w_i 2^{-i}$$



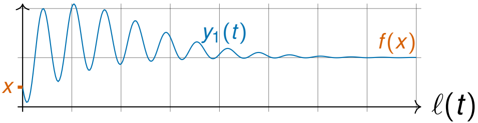
Theorem

PTIME = ANALOG-PTIME

Characterization of real polynomial time

Definition : $f : [a, b] \to \mathbb{R}$ in ANALOG-P$_\mathbb{R}$ $\Leftrightarrow \exists p$ polynomial, $\forall x \in [a, b]$

$$y(0) = (x, 0, \ldots, 0) \qquad y' = p(y)$$



Characterization of real polynomial time

Definition : $f : [a, b] \to \mathbb{R}$ in ANALOG-P$_{\mathbb{R}}$ $\Leftrightarrow \exists p$ polynomial, $\forall x \in [a, b]$

$$y(0) = (x, 0, \ldots, 0) \qquad y' = p(y)$$

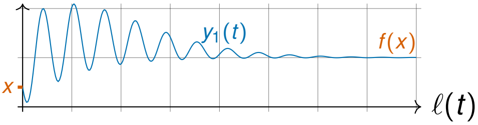
satisfies :

1. $|y_1(t) - f(x)| \leqslant 2^{-\ell(t)}$

   «greater length $\Rightarrow$ greater precision»

2. $\ell(t) \geqslant t$

   «length increases with time»



Characterization of real polynomial time

Definition : $f : [a, b] \to \mathbb{R}$ in ANALOG-P$_{\mathbb{R}}$ $\Leftrightarrow \exists p$ polynomial, $\forall x \in [a, b]$

$$y(0) = (x, 0, \ldots, 0) \qquad y' = p(y)$$

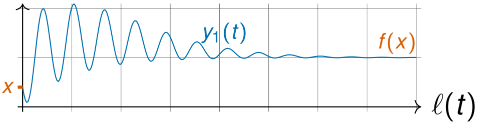
satisfies :

1. $|y_1(t) - f(x)| \leqslant 2^{-\ell(t)}$

   «greater length $\Rightarrow$ greater precision»

2. $\ell(t) \geqslant t$

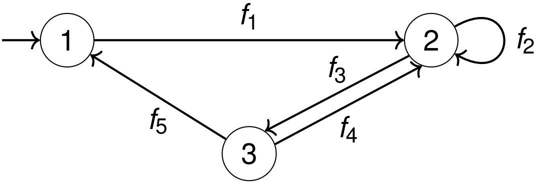
   «length increases with time»



Theorem

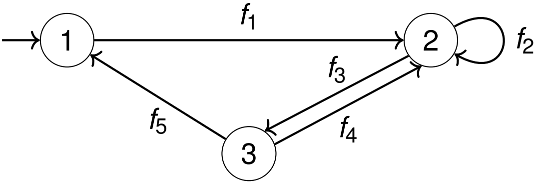
$f : [a, b] \to \mathbb{R}$ computable in polynomial time $\Leftrightarrow f \in$ ANALOG-P$_{\mathbb{R}}$.

Inductive invariants : example



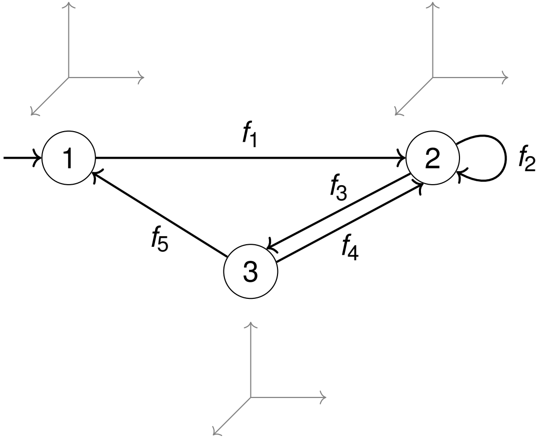
Inductive invariants : example

x, y, z range over $\mathbb{Q}$ $\quad\quad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



Inductive invariants : example

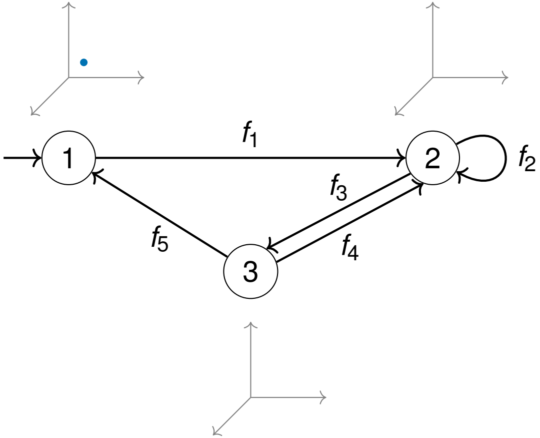
x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$



Inductive invariants : example

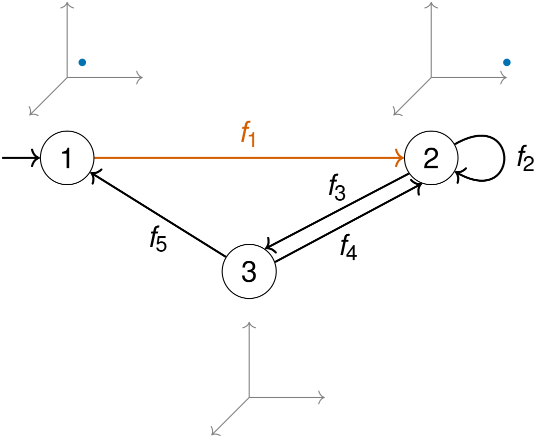
x, y, z range over $\mathbb{Q}$

$f_i : \mathbb{R}^3 \to \mathbb{R}^3$



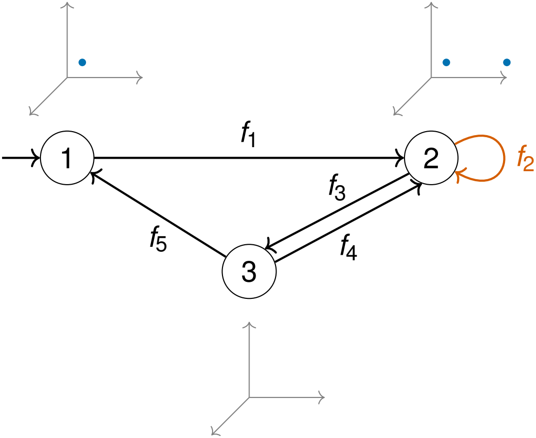
Inductive invariants : example

x, y, z range over $\mathbb{Q}$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



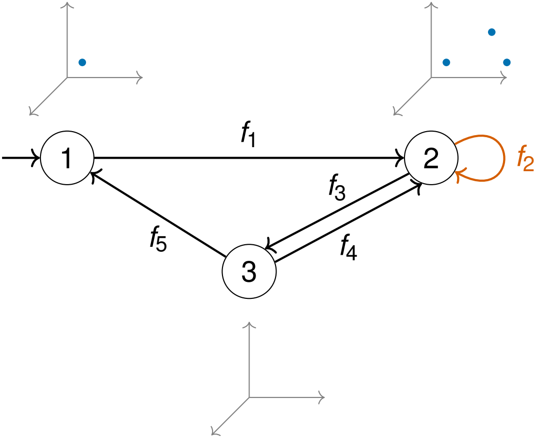
Inductive invariants : example

x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$



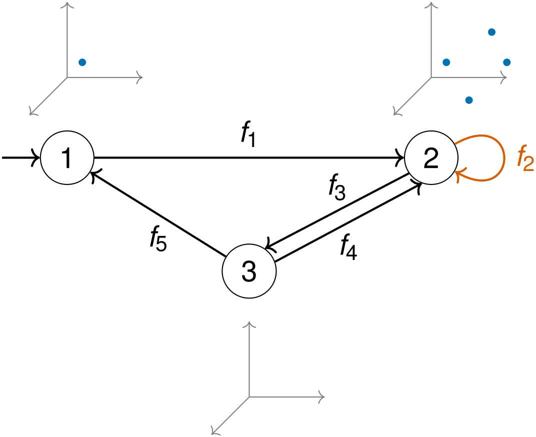
Inductive invariants : example

x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$



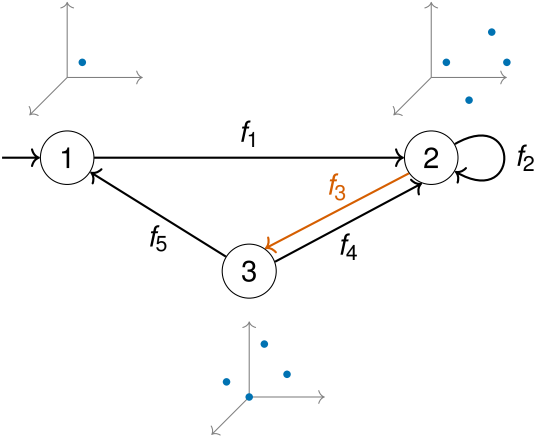
Inductive invariants : example

x, y, z range over $\mathbb{Q}$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



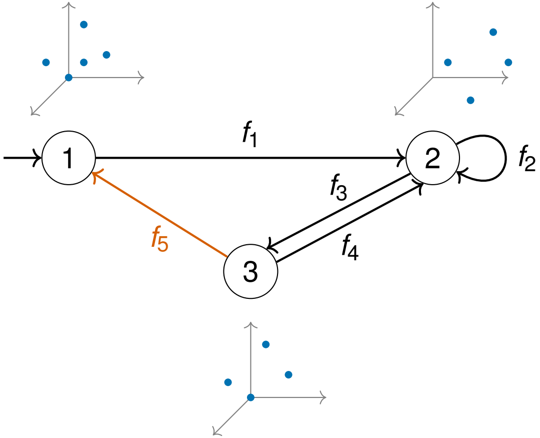
Inductive invariants : example

x, y, z range over $\mathbb{Q}$ $\quad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



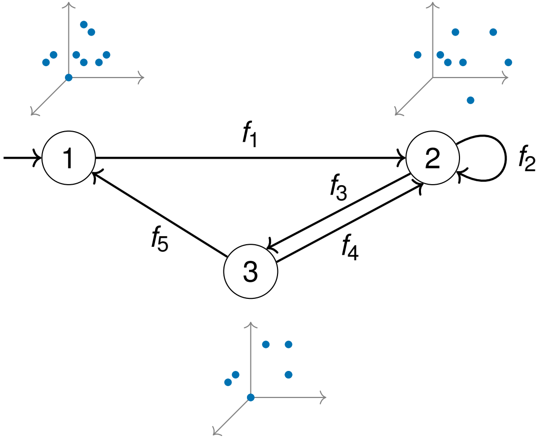
Inductive invariants : example

x, y, z range over $\mathbb{Q}$ $\qquad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



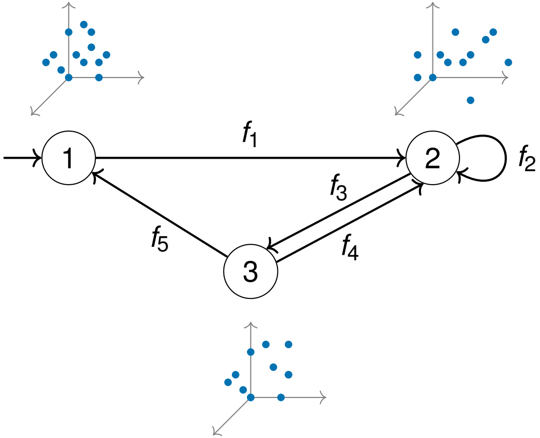
Inductive invariants : example

x, y, z range over $\mathbb{Q}$ $\qquad\qquad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



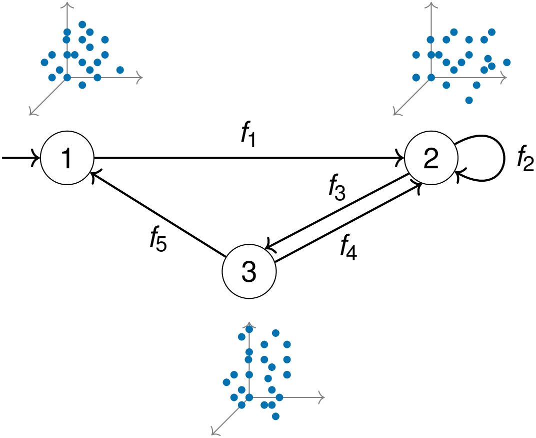
Inductive invariants : example

x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$



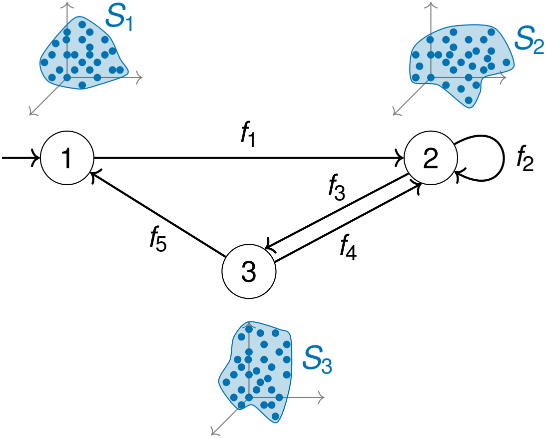
Inductive invariants : example

x, y, z range over $\mathbb{Q}$ $\qquad$ $f_i : \mathbb{R}^3 \to \mathbb{R}^3$



Inductive invariants : example

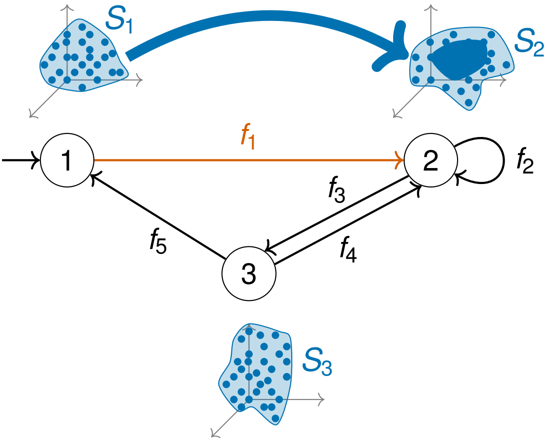
x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$



S_1, S_2, S_3 is an invariant

Inductive invariants : example

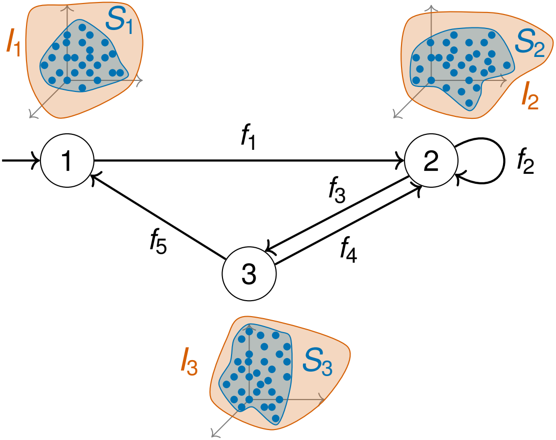
x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$



S_1,S_2,S_3 is an inductive invariant

Inductive invariants : example

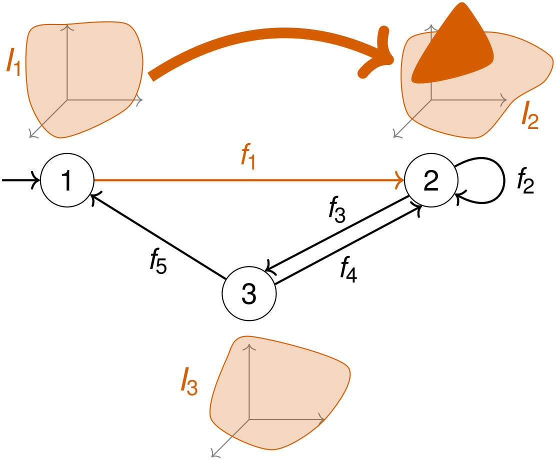
x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$



I_1, I_2, I_3 is an invariant

Inductive invariants : example

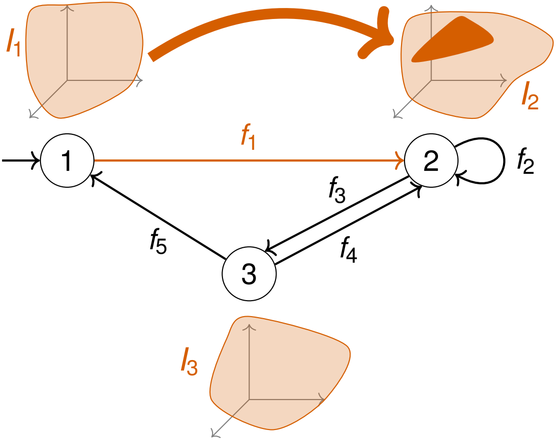
x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$



I_1, I_2, I_3 is **NOT** an inductive invariant

Inductive invariants : example

x, y, z range over $\mathbb{Q}$ $\qquad f_i : \mathbb{R}^3 \to \mathbb{R}^3$



I_1, I_2, I_3 is an inductive invariant