A universal ordinary differential equation

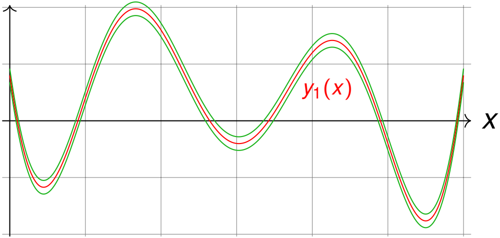
Olivier Bournez[1], Amaury Pouly[2]

[1]LIX, École Polytechnique, France

[2]Max Planck Institute for Software Systems, Germany

12 july 2017

Universal differential algebraic equation (Rubel)



Theorem (Rubel, 1981)

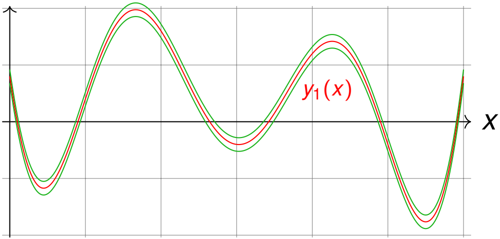
There exists a **fixed** polynomial p and $k \in \mathbb{N}$ such that for any continuous functions f and ε, there exists a solution y to

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

such that

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

Universal differential algebraic equation (Rubel)



Theorem (Rubel, 1981)

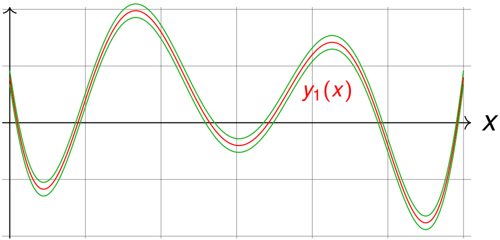
There exists a **fixed** polynomial p and $k \in \mathbb{N}$ such that for any continuous functions f and ε, there exists a solution y to

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

such that

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

Universal differential algebraic equation (Rubel)



Open Problem

This is a DAE. Is there a universal ODE ?

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 to

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

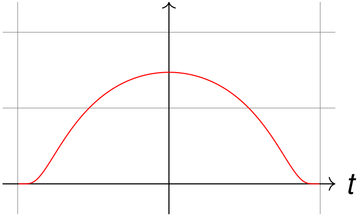
such that

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

Rubel's (disappointing) proof in one slide

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

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

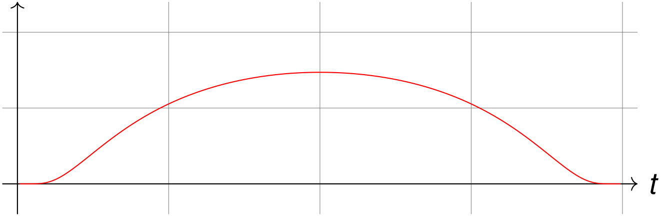


Rubel's (disappointing) proof in one slide

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

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

- Can do the same with $cf(at+b)$ (translation+scaling)

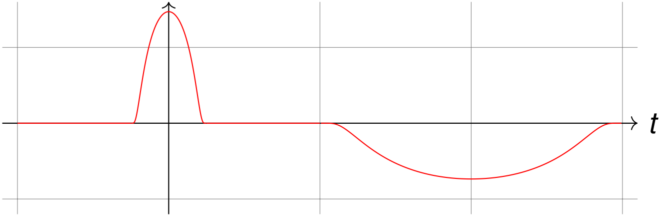


Rubel's (disappointing) proof in one slide

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

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

- Can do the same with $cf(at+b)$ (translation+scaling)
- Can glue together arbitrary many such pieces

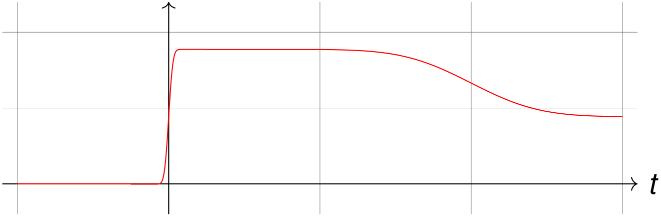


Rubel's (disappointing) proof in one slide

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

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

- Can do the same with $cf(at+b)$ (translation+scaling)
- Can glue together arbitrary many such pieces
- Can arrange so that $\int f$ is solution : piecewise pseudo-linear

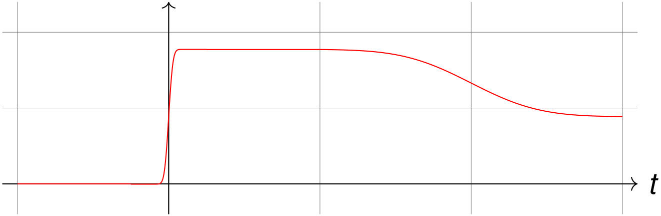


Rubel's (disappointing) proof in one slide

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

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

- Can do the same with $cf(at+b)$ (translation+scaling)
- 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

The problem with Rubel's DAE

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

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

The problem with Rubel's DAE

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

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

- ...even with a countable number of extra conditions :

$$p(y, y', \dots, y^{(k)}) = 0, y^{(d_i)}(a_i) = b_i, i \in \mathbb{N}$$

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** :

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

- ...even with a countable number of extra conditions :

$$p(y, y', \dots, y^{(k)}) = 0, y^{(d_i)}(a_i) = b_i, i \in \mathbb{N}$$

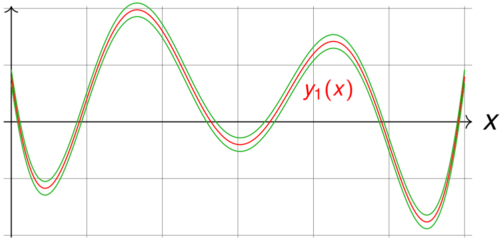
In fact, this is fundamental for Rubel's proof to work !

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

Open Problem (Rubel, 1981)

This is a DAE. Is there a universal ODE $y' = p(y)$?
Note : ODE $\Rightarrow$ unique solution

Universal ordinary differential equation (ODE)



Main result

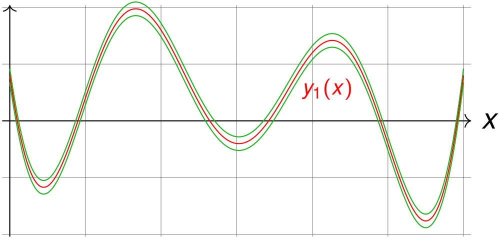
There exists a **fixed** polynomial p and $d \in \mathbb{N}$ such that for any continuous functions f and ε, there exists $\alpha \in \mathbb{R}^d$ such that

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

has a **unique solution** and this solution satisfies

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

Universal ordinary differential equation (ODE)



Main result

There exists a **fixed** polynomial p and $d \in \mathbb{N}$ 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** and this solution satisfies

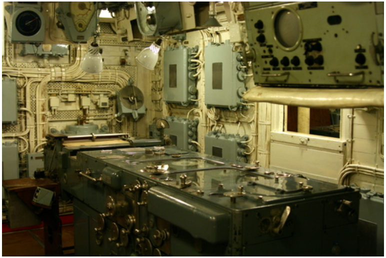
$$|y(t) - f(t)| \leqslant \varepsilon(t).$$

Unfortunately, we need $d \approx 300$.

Wait, is this a CS talk ?

Polynomial ODEs correspond to analog computers :

Differential Analyser

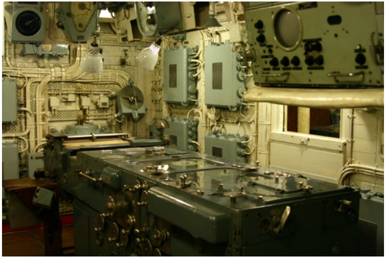


British Navy mecanical computer

Wait, is this a CS talk ?

Polynomial ODEs correspond to analog computers :

Differential Analyser

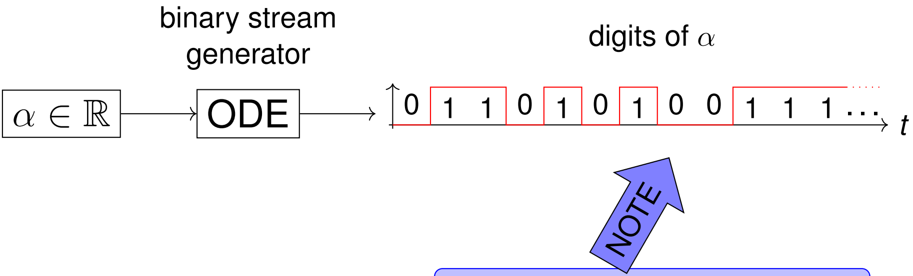


British Navy mecanical computer

- They are **equivalent** to Turing machines !
- One can **characterize P** with pODEs (ICALP 2016)

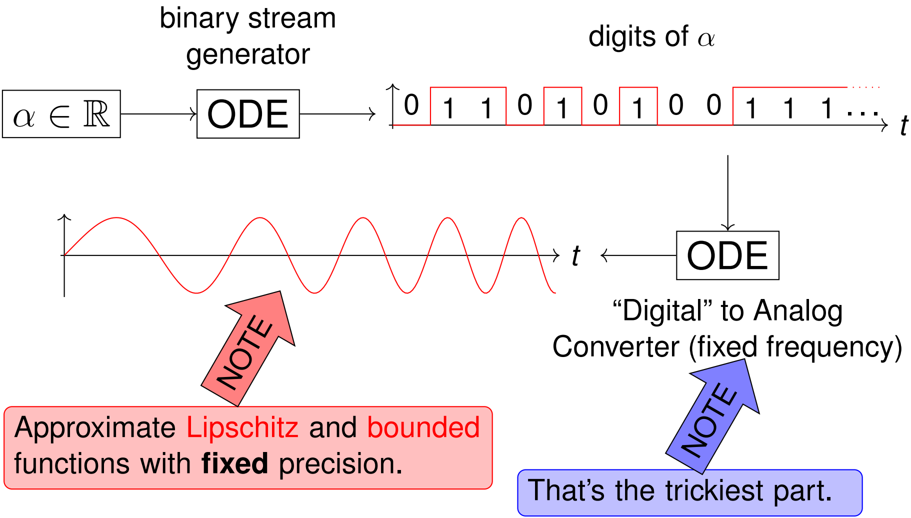
Take away : polynomial ODEs is a natural programming language.

A first idea

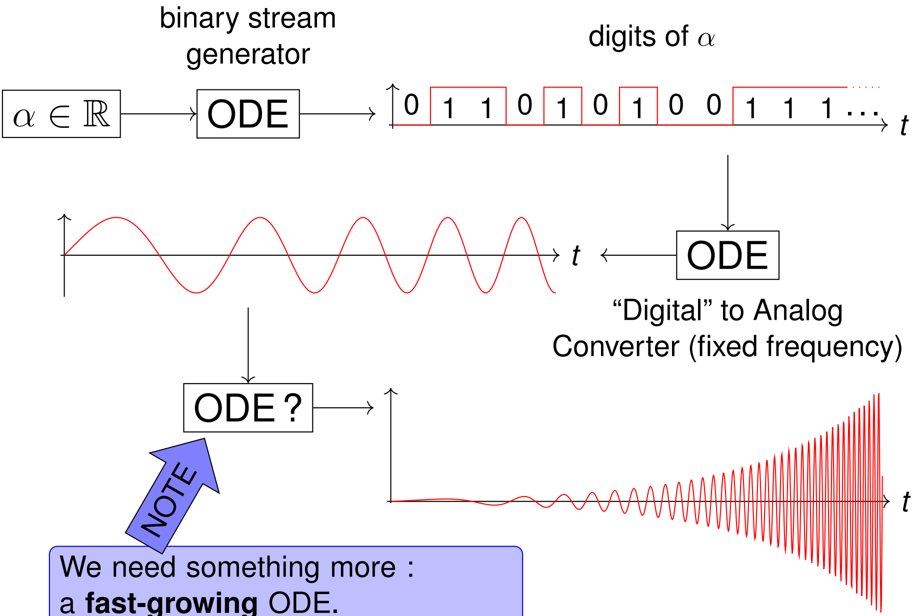


This is the ideal curve, the real one is an approximation of it.

A first idea

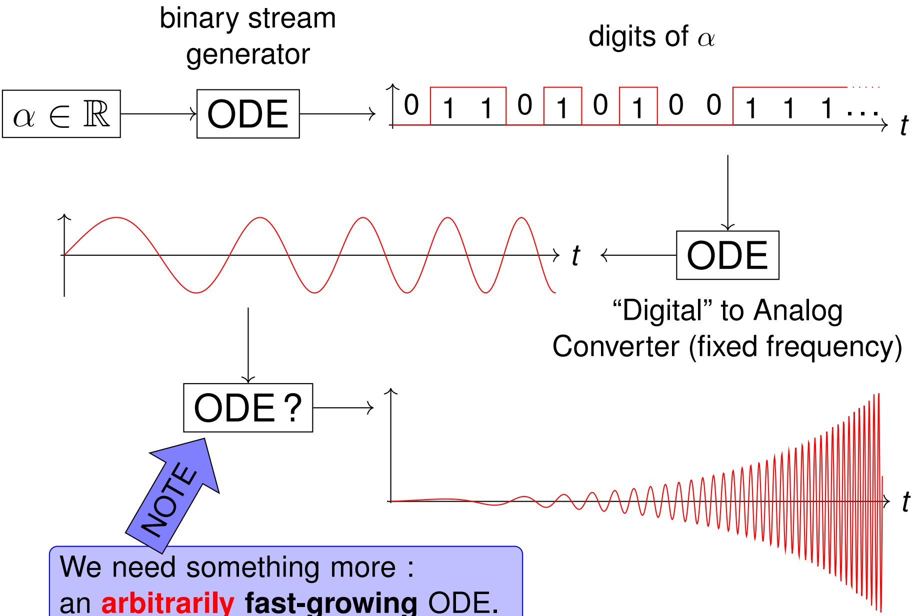

binary stream generator
digits of α
$\alpha \in \mathbb{R}$
ODE
0 1 1 0 1 0 1 0 0 1 1 1 ...
t
t
ODE
"Digital" to Analog Converter (fixed frequency)
NOTE
NOTE
Approximate Lipschitz and bounded functions with **fixed** precision.
That's the trickiest part.

A first idea



We need something more :
a **fast-growing** ODE.

A first idea



We need something more :
an **arbitrarily** **fast-growing** ODE.

An old question on growth

Building a fast-growing ODE :

$$y_1' = y_1 \qquad \rightsquigarrow \qquad y_1(t) = \exp(t)$$

An old question on growth

Building a fast-growing ODE :

$$\begin{aligned} y_1' &= y_1 & \rightsquigarrow \quad & y_1(t) = \exp(t) \\ y_2' &= y_1 y_2 & \rightsquigarrow \quad & y_1(t) = \exp(\exp(t)) \end{aligned}$$

An old question on growth

Building a fast-growing ODE :

$$
\begin{array}{lcl}
y_1' = y_1 & \rightsquigarrow & y_1(t) = \exp(t) \\
y_2' = y_1 y_2 & \rightsquigarrow & y_1(t) = \exp(\exp(t)) \\
\dots & & \dots \\
y_n' = y_1 \cdots y_n & \rightsquigarrow & y_n(t) = \exp(\cdots \exp(t) \cdots) := e_n(t)
\end{array}
$$

An old question on growth

Building a fast-growing ODE :

$$
\begin{array}{lcl}
y_1' = y_1 & \leadsto & y_1(t) = \exp(t) \\
y_2' = y_1 y_2 & \leadsto & y_1(t) = \exp(\exp(t)) \\
\ldots & & \ldots \\
y_n' = y_1 \cdots y_n & \leadsto & y_n(t) = \exp(\cdots \exp(t) \cdots) := e_n(t)
\end{array}
$$

Conjecture (Emil Borel, 1899)

With n variables, cannot do better than $\mathcal{O}_t(e_n(At^k))$.

An old question on growth

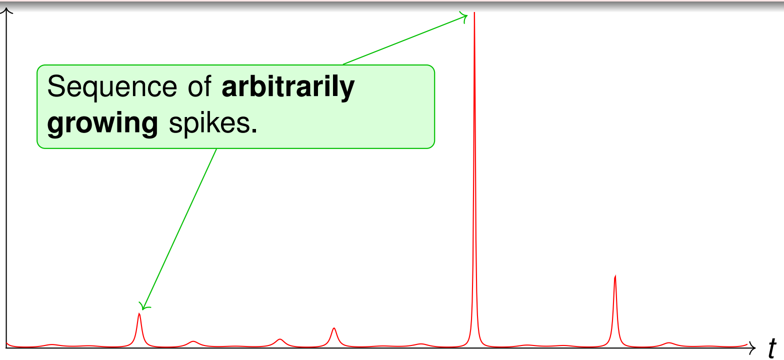
$e_n(t) = \exp(\cdots \exp(t) \cdots)$ (n compositions)

Conjecture (Emil Borel, 1899)

With n variables, cannot do better than $\mathcal{O}_t(e_n(At^k))$.

Counter-example (Vijayaraghavan, 1932)

$$\frac{1}{2 - \cos(t) - \cos(\alpha t)}$$



An old question on growth

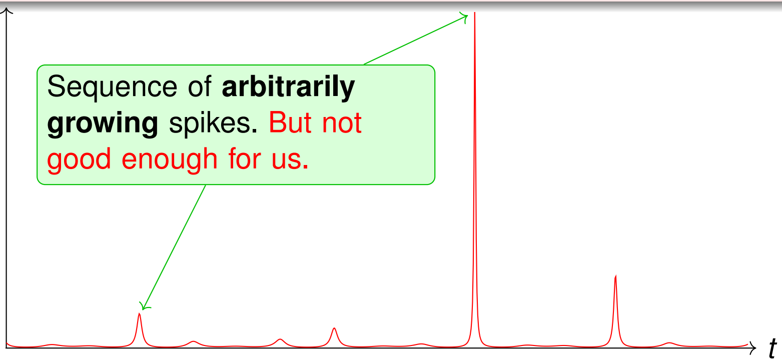
$e_n(t) = \exp(\cdots \exp(t) \cdots)$ (n compositions)

Conjecture (Emil Borel, 1899)

With n variables, cannot do better than $\mathcal{O}_t(e_n(At^k))$.

Counter-example (Vijayaraghavan, 1932)

$$\frac{1}{2 - \cos(t) - \cos(\alpha t)}$$



An old question on growth

$e_n(t) = \exp(\cdots \exp(t) \cdots)$ (n compositions)

Conjecture (Emil Borel, 1899)

With n variables, cannot do better than $\mathcal{O}_t(e_n(At^k))$.

Counter-example (Vijayaraghavan, 1932)

$$\frac{1}{2 - \cos(t) - \cos(\alpha t)}$$

Theorem (In the paper)

There exists a polynomial $p : \mathbb{R}^d \to \mathbb{R}^d$ such that for any continuous function $f : \mathbb{R}_+ \to \mathbb{R}$, we can find $\alpha \in \mathbb{R}^d$ such that

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

satisfies

$$y_1(t) \geqslant f(t) \qquad \forall t \geqslant 0.$$

An old question on growth

$e_n(t) = \exp(\cdots\exp(t)\cdots)$ (n compositions)

Conjecture (Emil Borel, 1899)

With n variables, cannot do better than $\mathcal{O}_t(e_n(At^k))$.

Counter-example (Vijayaraghavan, 1932)

$$\frac{1}{2-\cos(t)-\cos(\alpha t)}$$

Theorem (In the paper)

There exists a polynomial $p : \mathbb{R}^d \to \mathbb{R}^d$ such that for any continuous function $f : \mathbb{R}_+ \to \mathbb{R}$, we can find $\alpha \in \mathbb{R}^d$ such that

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

satisfies

$$y_1(t) \geqslant f(t) \qquad \forall t \geqslant 0.$$

Note : both results require α to be **transcendental**. Conjecture still open for **rational** coefficients.

Proof gem : iteration with differential equations

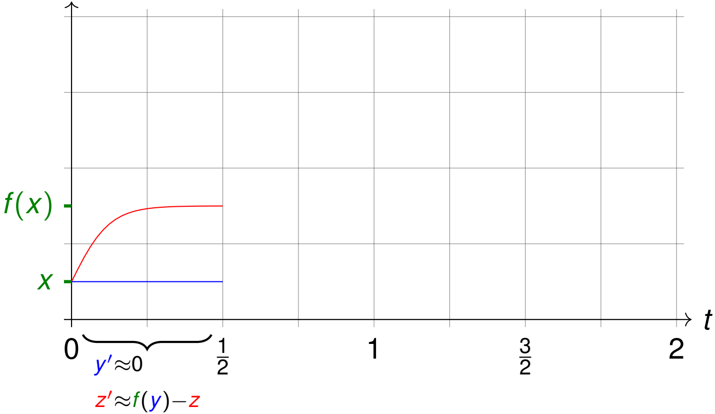
Goal

Iterate f with a GPAC : $y(n) \approx f^{[n]}([x])$

Proof gem : iteration with differential equations

Goal

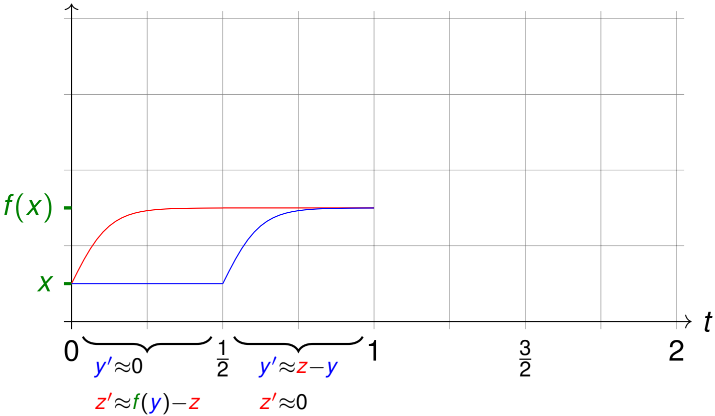
Iterate f with a GPAC : $y(n) \approx f^{[n]}([x])$



Proof gem : iteration with differential equations

Goal

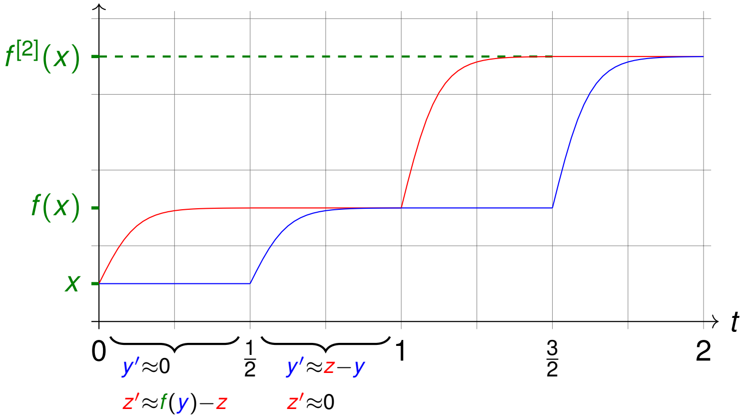
Iterate f with a GPAC : $y(n) \approx f^{[n]}([x])$



Proof gem : iteration with differential equations

Goal

Iterate f with a GPAC : $y(n) \approx f^{[n]}([x])$



Conclusion

This paper

positive answer to Rubel's open problem

Take home

ODE is a simple, nice and fun programming language

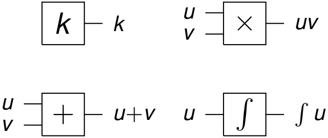
Possible development

Each universal ODE defines a map :

$$(f, \varepsilon) \in C^0 \times C^0 \mapsto \alpha \in \mathbb{R}$$

Kolmogorov-like complexity for continuous functions ?

Polynomial Differential Equations



General Purpose
Analog Computer

Differential Analyzer

Newton mechanics

Reaction networks :

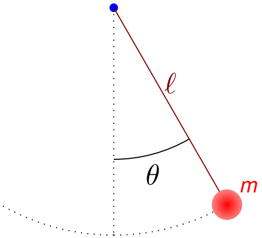
- chemical
- enzymatic

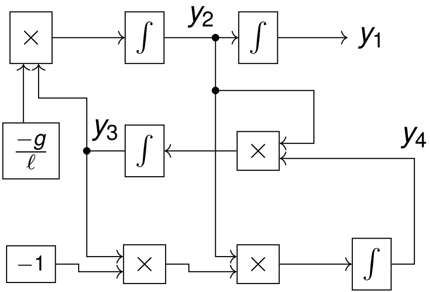
polynomial differential equations :

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

- Rich class
- Stable (+,×,∘,/,ED)
- No closed-form solution

Example of differential equation

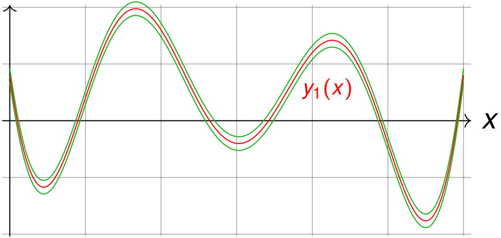




$$\ddot{\theta} + \frac{g}{\ell}\sin(\theta) = 0$$

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

Universal differential equation (DAE)



Theorem

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

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

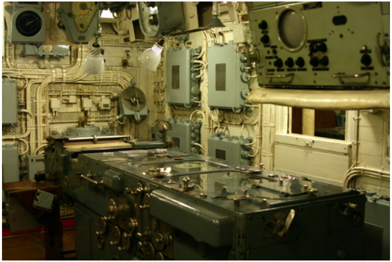
has a **unique analytic solution** and this solution satisfies

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

Digital vs analog computers

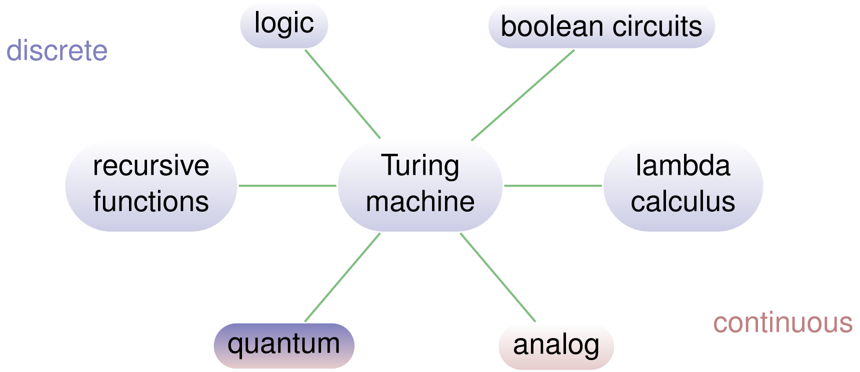
Digital vs analog computers

VS



Church Thesis

Computability

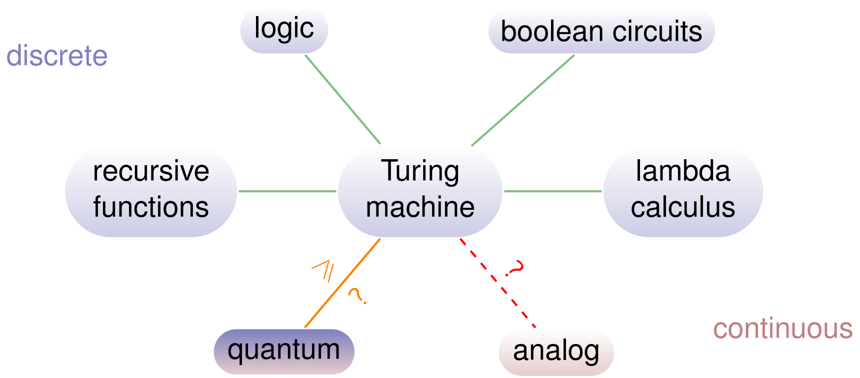


Church Thesis

All reasonable models of computation are equivalent.

Church Thesis

Complexity



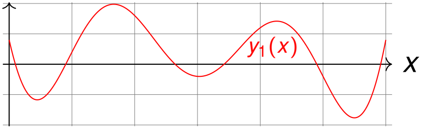
Effective Church Thesis

All reasonable models of computation are equivalent for complexity.

Generable functions

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

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



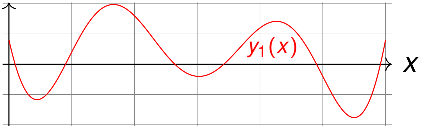
Shannon's notion

Computing with the GPAC

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, ...

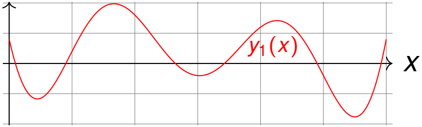
Strictly weaker than Turing machines [Shannon, 1941]

Computing with the GPAC

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

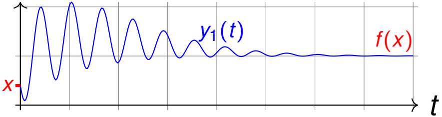
sin, cos, exp, log, ...

Strictly weaker than Turing machines [Shannon, 1941]

Computable

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

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



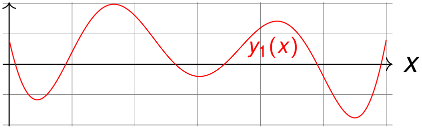
Modern notion

Computing with the GPAC

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

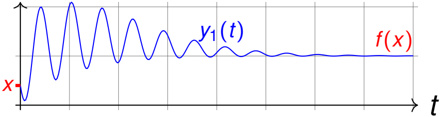
$\sin, \cos, \exp, \log, \ldots$

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

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

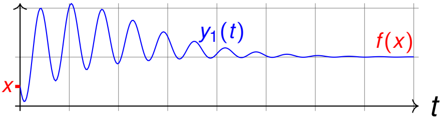
Turing powerful [Bournez et al., 2007]

Universal differential equations

Generable functions

subclass of analytic functions

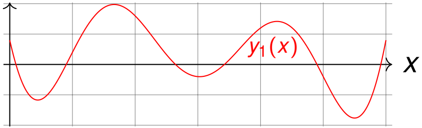
Computable functions



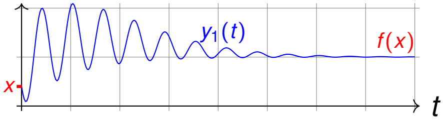
any computable function

Universal differential equations

Generable functions

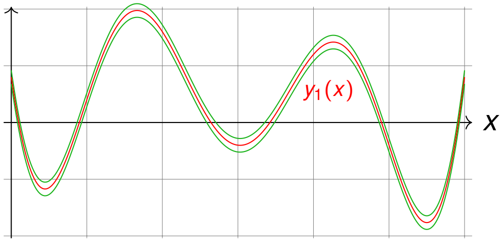


Computable functions



subclass of analytic functions

any computable function



A new notion of computability

Almost-Theorem

$f : [0,1] \to \mathbb{R}$ is **computable** if and only if there exists $\tau > 1$, $y_0 \in \mathbb{R}^d$ and p polynomial such that

$$y'(0) = y_0, \qquad y'(t) = p(y(t))$$

satisfies

$$|f(x) - y(x + n\tau)| \leqslant 2^{-n}, \qquad \forall x \in [0,1], \forall n \in \mathbb{N}$$