# A Universal Ordinary Differential Equation*

**Olivier Bournez**[†1] **and Amaury Pouly**[2]

1 **Ecole Polytechnique, LIX, Palaiseau Cedex, France**
bournez@lix.polytechnique.fr

2 **MPI-SWS, Saarbrücken, Germany**
pamaury@mpi-sws.org

## Abstract

An astonishing fact was established by Lee A. Rubel (1981): there exists a fixed non-trivial fourth-order polynomial differential algebraic equation (DAE) such that for any positive continuous function $\varphi$ on the reals, and for any positive continuous function $\epsilon(t)$, it has a $\mathcal{C}^\infty$ solution with $|y(t) - \varphi(t)| < \epsilon(t)$ for all $t$. Lee A. Rubel provided an explicit example of such a polynomial DAE. Other examples of universal DAE have later been proposed by other authors.

However, while these results may seem very surprising, their proofs are quite simple and are frustrating for a computability theorist, or for people interested in modeling systems in experimental sciences. First, the involved notions of universality is far from usual notions of universality in computability theory because the proofs heavily rely on the fact that constructed DAE does not have unique solutions for a given initial data. Indeed, in general a DAE may not have a unique solution, given some initials conditions. But Rubel's DAE *never* has a unique solution, even with a countable number of conditions of the form $y^{(k_i)}(a_i) = b_i$. This is very different from usual notions of universality where one would expect that there is clear unambiguous notion of evolution for a given initial data, for example as in computability theory. Second, the proofs usually rely on solutions that are piecewise defined. Hence they cannot be analytic, while analycity is often a key expected property in experimental sciences. Third, the proofs of these results can be interpreted more as the fact that (fourth-order) polynomial algebraic differential equations is a too loose a model compared to classical ordinary differential equations. In particular, one may challenge whether the result is really a universality result.

The question whether one can require the solution that approximates $\varphi$ to be the unique solution for a given initial data is a well known open problem [Rubel 1981, page 2], [Boshernitzan 1986, Conjecture 6.2]. In this article, we solve it and show that Rubel's statement holds for polynomial ordinary differential equations (ODEs), and since polynomial ODEs have a unique solution given an initial data, this positively answers Rubel's open problem. More precisely, we show that there exists a **fixed** polynomial ODE such that for any $\varphi$ and $\epsilon(t)$ there exists some initial condition that yields a solution that is $\epsilon$-close to $\varphi$ at all times.

The proof uses ordinary differential equation programming. We believe it sheds some light on computability theory for continuous-time models of computations. It also demonstrates that ordinary differential equations are indeed universal in the sense of Rubel and hence suffer from the same problem as DAEs for modelization: a single equation is capable of modelling any phenomenon with arbitrary precision, meaning that trying to fit a model based on polynomial DAEs or ODEs is too general (if it has a sufficient dimension).

**1998 ACM Subject Classification** G.1.7 Ordinary Differential Equations F.1.1 Models of Computation. F.1.3 Complexity Measures and Classes

* Full version at https://arxiv.org/abs/1702.08328.

† Olivier Bournez was partially supported by *ANR PROJECT RACAF*.


44th International Colloquium on Automata, Languages, and Programming (ICALP 2017).
Editors: Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl;
Article No. 116; pp. 116:1–116:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

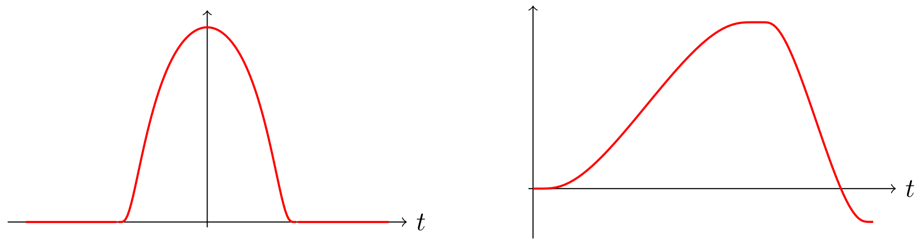


**Figure 1** On left, graphical representation of function $g$. On right, two $S$-modules glued together.

**Keywords and phrases** Ordinary Differential Equations, Universal Differential Equations, Analog Models of Computation, Continuous-Time Models of Computation, Computability, Computational Analysis, Computational Complexity

**Digital Object Identifier** 10.4230/LIPIcs.ICALP.2017.116

# 1 Introduction

A very astonishing result was established by Lee A. Rubel in 1981 [19]. There exists a universal fourth-order algebraic differential equation in the following sense.

▶ **Theorem 1** ([19]). *There exists a non-trivial fourth-order implicit differential algebraic equation*

$$P(y', y'', y''', y'''') = 0 \tag{1}$$

*where $P$ is a polynomial in four variables with integer coefficients, such that for any continuous function $\varphi$ on $(-\infty, \infty)$ and for any positive continuous function $\epsilon(t)$ on $(-\infty, \infty)$, there exists a $\mathcal{C}^\infty$ solution $y$ such that*

$$|y(t) - \varphi(t)| < \epsilon(t)$$

*for all $t \in (-\infty, \infty)$.*

Even more surprising is the fact that Rubel provided an explicit example of such a polynomial $P$ that is particularly simple:

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

While this result looks very surprising at first sight, Rubel's proofs turns out to use basic arguments, and can be explained as follows. It uses the following classical trick to build $\mathcal{C}^\infty$ piecewise functions: let $g(t) = e^{-1/(1-t^2)}$ for $-1 < t < 1$, and $g(t) = 0$ otherwise. It is not hard to see that function $g$ is $C^\infty$ and Figure 1 shows that $g$ looks like a "bump". Since it satisfies $\frac{g'(t)}{g(t)} = -\frac{2t}{(1-t^2)^2}$, then $g'(t)(1-t^2)^2 + g(t)2t = 0$ and $f(t) = \int_0^t g(u)du$ satisfies the polynomial differential algebraic equation $f''(1-t^2)^2 + f'(t)2t = 0$. Since this equation is homogeneous, it also holds for $af + b$ for any $a$ and $b$. The idea is then to obtain a fourth order DAE that is satisfied by every function $y(t) = \gamma f(\alpha t + \beta) + \delta$, for all $\alpha, \beta, \gamma, \delta$. After some computations, Rubel obtained the universal differential equation (2).

Functions of the type $y(t) = \gamma f(\alpha t + \beta) + \delta$ generate what Rubel calls $S$-modules: a function that values $A$ at $a$, $B$ at $b$, is constant on $[a, a+\delta]$, monotone on $[a+\delta, b-\delta]$, constant on $[b-\delta, b]$, by an appropriate choice of $\alpha, \beta, \gamma, \delta$. Summing $S$-modules corresponds to gluing then together, as is depicted in Figure 1. Note that finite, as well as infinite sums[1] of $S$-modules still satisfy the equation (2) and thus any piecewise affine function (and hence any continuous function) can be approximated by an appropriate sum of $S$-modules. This concludes Rubel's proof of universality.

As one can see, the proof turns out to be frustrating because the equation essentially allows any behavior. This may be interpreted as merely stating that differential algebraic equations is simply too lose a model. Clearly, a key point is that this differential equation does not have a unique solution for any given initial condition: this is the core principle used to glue a finite or infinite number of $S$-modules and to approximate any continuous function. Rubel was aware of this issue and left open the following question in [19, page 2].

> "It is open whether we can require in our theorem that the solution that approximates $\varphi$ to be the unique solution for its initial data."

Similarly, the following is conjectured in [4, Conjecture 6.2].

> "Conjecture. There exists a non-trivial differential algebraic equation such that any real continuous function on $\mathbb{R}$ can be uniformly approximated on all of $\mathbb{R}$ by its real-analytic solutions"

The purpose of this paper is to provide a positive answer to both questions. We prove that a fixed polynomial ordinary differential equations (ODE) is universal in above Rubel's sense. At a high level, our proofs are based on ordinary differential equation programming. This programming is inspired by constructions from our previous paper [7]. Here, we mostly use this programming technology to achieve a very different goal and to provide positive answers to these above open problems.

We also believe they open some lights on computability theory for continuous-time models of computations. In particular, it follows that concepts similar to Kolmogorov complexity can probably be expressed naturally by measuring the complexity of the initial data of a (universal-) polynomial ordinary differential equations for a given function. We leave this direction for future work.

## 1.1 Related work and discussions

First, let us mention that Rubel's universal differential equation has been extended in several papers. In particular, Duffin proved in [12] that implicit universal differential equations with simpler expressions exists, such as $n^2 y'''' y'^2 + 3n(1-n)y''' y'' y' + (2n^2 - 3n + 1){y''}^3 = 0$ for any $n > 3$. The idea of [12] is basically to replace the $\mathcal{C}^\infty$ function $g$ of [19] by some piecewise polynomial of fixed degree, that is to say by splines. Duffin also proves that considering trigonometric polynomials for function $g(x)$ leads to the universal differential equation $ny'''' y'^2 + (2-3n)y''' y'' y' + 2(n-1)y''^3 = 0$. This is done at the price of approximating function $\varphi$ respectively by splines or trigonometric splines solutions which are $\mathcal{C}^n$ (and $n$ can be taken arbitrary big) but not $\mathcal{C}^\infty$ as in [19]. Article [8] proposes another universal differential equation whose construction is based on Jacobian elliptic functions. Notice that [8] is also correcting some statements of [12].

[1] With some convergence or disjoint domain conditions.

All the results mentioned so far are concerned with approximations of continuous functions over the whole real line. Approximating functions over a compact domain seems to be a different (and somewhat easier for our concerns) problem, since basically by compactness, one just needs to approximate the function locally on a finite number of intervals. A 1986 reference survey discussing both approximation over the real line and over compacts is [4]. Recently, over compact domains, the existence of universal ordinary differential equation $\mathcal{C}^\infty$ of order 3 has been established in [11]: it is shown that for any $a < b$, there exists a third order $\mathcal{C}^\infty$ differential equation $y''' = F(y, y', y'')$ whose solutions are dense in $\mathcal{C}^0([a, b])$. Notice that this is not obtained by explicitly stating such an order 3 universal ordinary differential, and that this is a weaker notion of universality as solutions are only assumed to be arbitrary close over a compact domain and not all the real line. Order 3 is argued to be a lower bound for Lipschitzian universal ODEs [11].

Rubel's result has sometimes been considered to be related to be the equivalent, for analog computers, of the universal Turing machines. This includes Rubel's paper motivation given in [19, page 1]. We now discuss and challenge this statement.

Indeed, differential algebraic equations are known to be related to the General Purpose Analog Computer (GPAC) of Claude Shannon [20], proposed as a model of the Differential Analysers [9], a mechanical programmable machine, on which he worked as an operator. Notice that the original relations stated by Shannon in [20] between differential algebraic equations and GPACs have some flaws, that have been corrected later by [18] and [13]. Using the better defined model of GPAC of [13], it can be shown that functions generated by GPAC exactly correspond to polynomial ordinary differential equations. Some recent results have established that this model, and hence polynomial ordinary differential equations can be related to classical computability [5] and complexity theory [7].

However, we do not really follow the statement that Rubel's result is the equivalent, for analog computers, of the universal Turing machines. In particular, Rubel's notion of universality is completely different from the ones in computability theory. For a given initial data, a (deterministic) Turing machine has only one possible evolution. On the other hand, Rubel's equation does not dictate any evolution but rather some conditions that any evolution has to satisfy. In other words, Rubel's equation can be interpreted as the equivalent of an invariant of the dynamics of (Turing) machines, rather than a universal machine in the sense of classical computability.

Notice that while several results have established that (polynomial) ODEs are able to simulate the evolution of Turing machines (see e.g. [5, 15, 7]), the existence of a universal ordinary differential equation does not follow from them. To understand the difference, let us restate the main result of [15], of which [7] is a more advanced version for polynomial-time computable functions.

► **Theorem 2.** *A function $f : [a, b] \to \mathbb{R}$ is computable (in the framework of Computable Analysis) if and only if there exists some polynomials $p : \mathbb{R}^{n+1} \to \mathbb{R}^n$, $p_0 : \mathbb{R} \to \mathbb{R}$ with computable coefficients and $\alpha_1, \dots, \alpha_{n-1}$ computable reals such that for all $x \in [a, b]$, the solution $y : [a, b] \to \mathbb{R}^n$ to the Cauchy problem*

$$y(0) = (\alpha_1, \dots, \alpha_{n-1}, p_0(x)), \qquad y' = p(y)$$

*satisfies that for all $t \geqslant 0$ that*

$$|f(x) - y_1(t)| \leqslant y_2(t) \quad \textit{and} \quad \lim_{t\to\infty} y_2(t) = 0.$$

Since there exists a universal Turing machine, there exists a "universal" polynomial ODE for computable functions. But there are major differences between Theorem 2 and the result of this paper (Theorem 3). Even if we have a strong link between the Turing machines's configuration and the evolution of the differential equation, this is not enough to guarantee what the trajectory of the system will be at all times. Indeed, Theorem 2 only guarantees that $y_1(t) \to f(x)$ asymptotically. On the other hand, Theorem 3 guarantees the value of $y_1(t)$ at all times. Notice that our universality result also applies to functions that are not computable (in which case the initial condition is computable from the function but still not computable).

We would like to mention some implications for experimental sciences that are related to the classical use of ODEs in such contexts. Of course, we know that this part is less formal from a mathematical point of view, but we believe this discussion has some importance: A key property in experimental sciences, in particular physics is analyticity. Recall that a function is analytic if its is equal to its Taylor expansion in any point. It has sometimes been observed that "natural" functions coming from Nature are analytic, even if this cannot be a formal statement, but more an observation. We obtain a fixed universal polynomial ODEs, so in particular all its solution must be analytic[2], and it follows that universality holds even with analytic functions. All previous constructions mostly worked by gluing together $\mathcal{C}^\infty$ or $\mathcal{C}^n$ functions, and as it is well known "gluing" of analytic functions is impossible. We believe this is an important difference with previous works.

As we said, Rubel's proof can be seen as an indication that (fourth-order) polynomial implicit DAE is too loose model compared to classical ODEs, allowing in particular to glue solutions together to get new solutions. As observed in many articles citing Rubel's paper, this class appears so general that from an experimental point of view, it makes littles sense to try to fit a differential model because a single equation can model everything with arbitrary precision. Our result implies the same for polynomial ODEs since, for the same reason, a single equation of sufficient dimension can model everything.

Notice that our constructions have at the end some similarities with Voronin's theorem. This theorem states that Riemann's $\zeta$ function is such that for any analytic function $f(z)$ that is non-vanishing on a domain $U$ homeomorphic to a closed disk, and any $\epsilon > 0$, one can find some real value $t$ such that for all $z \in U$, $|\zeta(z + it) - f(z)| < \epsilon$. Notice that $\zeta$ function is a well-known function known not to be solution of any polynomial DAE (and consequently polynomial ODE), and hence there is no clear connexion to our constructions based on ODEs. We invite to read the post [17] in "Gödel's Lost Letter and P=NP" blog for discussions about potential implications of this surprising result to computability theory.

## 1.2 Formal statements

▶ **Theorem 3** (Universal PIVP). *There exists a **fixed** polynomial vector $p$ in $d$ variables such that for any functions $f \in C^0(\mathbb{R})$ and $\varepsilon \in C^0(\mathbb{R}, \mathbb{R}_{>0})$, there exists $\alpha \in \mathbb{R}^d$ such that there exists a unique solution $y : \mathbb{R} \to \mathbb{R}^d$ to $y(0) = \alpha$, $y' = p(y)$. Furthermore, this solution satisfies that $|y_1(t) - f(t)| \leqslant \varepsilon(t)$ for all $t \in \mathbb{R}$, and it is analytic.*

It is well-known that polynomial ODEs can be transformed into DAEs that have the same analytic solutions, see [10] for example. The following then follows for DAEs.

▶ **Theorem 4** (Universal DAE). *There exists a **fixed** polynomial $p$ in $d+1$ variables such that for any functions $f \in C^0(\mathbb{R})$ and $\varepsilon \in C^0(\mathbb{R}, \mathbb{R}_{>0})$, there exists $\alpha_0, \ldots, \alpha_{d-1} \in \mathbb{R}$ such that*

[2] Which is not the case for polynomial DAEs.

*there exists a unique* ***analytic*** *solution* $y:\mathbb{R}\to\mathbb{R}$ *to* $y(0)=\alpha_0, y'(0)=\alpha_1,\dots,y^{(d-1)}(0)=\alpha_{d-1}$*,* $p(y,y',\dots,y^d)=0$*. Furthermore, this solution satisfies that* $|y(t)-f(t)|\leqslant\varepsilon(t)$ *for all* $t\in\mathbb{R}$*.*

▶ Remark. Notice that both theorems apply even when $f$ is not computable. In this case, the initial condition(s) $\alpha$ exist but are not computable. We believe that $\alpha$ is always *computable from* $f$ and $\varepsilon$, that is the mapping $(f,\varepsilon)\mapsto\alpha$ is computable in the framework of Computable Analysis, with an adequate representation of $f,\varepsilon$ and $\alpha$.

▶ Remark. Notice that we do not provide explicitly in this paper the considered polynomial ODE, nor its dimension $d$. But it can be derived by following the constructions. We currently estimate $d$ to be more than three hundred following the precise constructions of this paper (but also to be very far from the optimal). We did not try to minimize $d$ in the current paper, as we think our results are sufficiently hard to be followed in this paper for not beeing complicated by considerations about optimizations of dimensions.

▶ Remark. Both theorems are stated for *total functions* $f$ and $\varepsilon$ over $\mathbb{R}$. It trivially applies to any continuous partial function that can be extended to a continuous function over $\mathbb{R}$. In particular, it applies to any functions over $[a,b]$. It is not hard to see that it also applies to functions over $(a,b)$ by rescaling $\mathbb{R}$ into $(a,b)$ using the cotangent:

$$z(t)=y\left(-\cot\left(\tfrac{t-a}{b-a}\pi\right)\right)\quad\text{satisfies}\quad z'(t)=\phi'(t)p(z(t)),\quad \phi'(t)=\tfrac{\pi}{b-a}(1+\phi(t)^2).$$

More complex domains such as $[a,b)$ and $(a,b]$ (with $a$ possibly infinite) can also be obtain in a similar fashion.

## 2 Overview of the proof

A first a priori difficulty is that if one considers a fixed polynomial ODE $y'=p(y)$, one could think that the growth of its solutions is constrained by $p$ and thus cannot be arbitrary. This would then prevent us from building a universal ODE simply because it could not grow fast enough. This fact is related to Emil Borel's conjecture in [3] (see also [16]) that a solution, *defined over* $\mathbb{R}$, to a system with $n$ variables has growth bounded by roughly $e_n(x)$, the $n-$th iterate of exp. The conjecture is proved for $n=1$ [3], but has been proven to be false for $n=2$ in [21] and [2]. Bank [1] then adapted the previous counter-examples to provide a DAE whose non-unique increasing real-analytic solutions at infinity do not have any majorant. See the discussions (and Conjecture 6.1) in [4] for discussions about the growth of solutions of DAEs, and their relations to functions $e_n(x)$.

Thus, the first important part of this paper is to refine Bank's counter-example to build `fastgen`, a fast-growing function that satisfies even stronger properties. The second major ingredient is to be able to approximate a function with arbitrary precision everywhere. Since this is a difficult task, we use `fastgen` to our advantage to show that it is enough to approximate functions that are bounded and change slowly (think 1-Lipschitz, although the exact condition is more involved). That is to say, to deal with the case where there is no problem about the growth and rate of change of functions in some way. This is the purpose of the function `pwcgen` which can build arbitrary almost piecewise constant functions as long as they are bounded and change slowly.

It should be noted that the entire paper, we construct *generable functions* (in several variables) (see Section 3.1). For most of the constructions, we only use basic facts like the fact that generable functions are stable under arithmetic, composition and ODE solving. We know that generable functions satisfy polynomial partial equations and use this fact only at the very end to show that the generable approximation that we have built, in fact, translates to a polynomial ordinary differential equation.

The rest of the paper is organized as follows. In Section 3, we recall some concepts and results from other articles. The main purpose of this section is to present Theorem 10. This theorem is the analog equivalent of doing an assignment in a periodic manner. Section 4 is devoted to `fastgen`, the fast-growing function. In Section 5, we show how to generate a sequence of dyadic rationals. In Section 6, we show how to generate a sequence of bits. In Section 7, we show how to leverage the two previous sections to generate arbitrary almost piecewise constant functions. Section 8 is then devoted to the proof of our main theorem.

## 3 Concepts and results from other articles

### 3.1 Generable functions

The following concept can be attributed to [20]: a function $f : \mathbb{R} \to \mathbb{R}$ is said to be a PIVP (Polynomial Initial Value Problem) function if there exists a system of the form $y' = p(y)$, where $p$ is a (vector of) polynomial, with $f(t) = y_1(t)$ for all $t$, where $y_1$ denotes first component of the vector $y$ defined in $\mathbb{R}^d$. We need in our proof to extend this concept to talk about multivariable functions. In [6], we introduced the following class, which can be seen as extensions of [14].

▶ **Definition 5** (Generable function). Let $d, e \in \mathbb{N}$, $I$ be an open and connected subset of $\mathbb{R}^d$ and $f : I \to \mathbb{R}^e$. We say that $f$ is generable if and only if there exists an integer $n \geqslant e$, a $n \times d$ matrix $p$ consisting of polynomials with coefficients in $\mathbb{R}$ , $x_0 \in \mathbb{R}^d$, $y_0 \in \mathbb{R}^n$ and $y : I \to \mathbb{R}^n$ satisfying for all $x \in I$:

- $y(x_0) = y_0$ and $J_y(x) = p(y(x))$ ▶ $y$ satisfies a polynomial differential equation[3],
- $f(x) = (y_1(x), \ldots, y_e(x))$ ▶ the components of $f$ are components of $y$.

This class strictly generalizes functions generated by polynomial ODEs. Indeed, in the special case of $d = 1$ (the domain of the function has dimension 1), the above definition is equivalent to saying that $y' = p(y)$ for some polynomial $p$. The interested reader can read more about this in [6].

For the purpose of this paper, the reader only needs to know that the class of generable functions enjoys many stability properties that make it easy to create new functions from basic operations. Informally, one can add, subtract, multiply, divide and compose them at will, the only requirement is that the domain of definition must always be connected. In particular, the class of generable functions contains some common mathematical functions:

- (multivariate) polynomials,
- trigonometric functions: sin, cos, tan, etc,
- exponential and logarithm: exp, ln,
- hyperbolic trigonometric functions: sinh, cosh, tanh.

Two famous examples of functions that are *not* in this class are the $\zeta$ and $\Gamma$, we refer the reader to [6] and [14] for more information.

A nontrivial fact is that generable functions are always analytic. This property is well-known in the one-dimensional case but is less obvious in higher dimensions, see [6] for more details. Moreover, generable functions satisfy the following crucial properties.

[3] $J_y$ denotes the Jacobian matrix of $y$.

▶ **Lemma 6** (Closure properties of generable functions). *Let $f :\subseteq \mathbb{R}^d \to \mathbb{R}^n$ and $g :\subseteq \mathbb{R}^e \to \mathbb{R}^m$ be generable functions. Then $f+g$, $f-g$, $fg$, $\frac{f}{g}$ and $f \circ g$ are generable*[4].

▶ **Lemma 7** (Generable functions are closed under ODE). *Let $d \in \mathbb{N}$, $J \subseteq \mathbb{R}$ an interval, $f :\subseteq \mathbb{R}^d \to \mathbb{R}^d$ generable, $t_0 \in J$ and $y_0 \in \operatorname{dom} f$. Assume there exists $y : J \to \operatorname{dom} f$ satisfying*

$$y(t_0) = y_0 \qquad y'(t) = f(y(t))$$

*for all $t \in J$, then $y$ is generable (and unique).*

In fact, generable functions satisfy the stronger (albeit more obscure) theorem

▶ **Theorem 8** (Generable functions are closed under ODE). *Let $d, n \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, $t_0 \in \mathbb{R}$, $(J_\alpha)_{\alpha\in\Omega}$ a family of open intervals containing $t_0$ and $G : \Omega \to \mathbb{R}^n$, $F :\subseteq \mathbb{R}^d \to \mathbb{R}^d$ generable. Assume that $X = \{(\alpha, t) : \alpha \in \Omega, t \in J_\alpha\}$ is an open connected set and that there exists $f : X \to \operatorname{dom} F$ satisfying*

$$f(\alpha, t_0) = G(\alpha) \qquad \frac{\partial f}{\partial t}(\alpha, t) = F(f(\alpha, t))$$

*for all $\alpha \in \Omega$ and $t \in J_\alpha$. Then $f$ is generable (and unique).*

## 3.2 Helper functions and constructions

We mentioned earlier that a number of common mathematical functions are generable. However, for our purpose, we will need less common functions that one can consider to be programming gadgets. One such operation is rounding (computing the nearest integer). Note that, by construction, generable functions are analytic and in particular must be continuous. It is thus clear that we cannot build a perfect rounding function and in particular we have to compromise on two aspects:

- we cannot round numbers arbitrarily close to $n + \frac{1}{2}$ for $n \in \mathbb{Z}$: thus the function takes a parameter $\lambda$ to control the size of the "zone" around $n + \frac{1}{2}$ where the function does not round properly,
- we cannot round without error: thus the function takes a parameters $\mu$ that controls how good the approximation must be.

▶ **Lemma 9** (Round, [6]). *There exists a generable function* round *such that for any $n \in \mathbb{Z}$, $x \in \mathbb{R}$, $\lambda > 2$ and $\mu \geqslant 0$:*

- *if $x \in \left[n - \frac{1}{2}, n + \frac{1}{2}\right]$ then $|\operatorname{round}(x, \mu, \lambda) - n| \leqslant \frac{1}{2}$,*
- *if $x \in \left[n - \frac{1}{2} + \frac{1}{\lambda}, n + \frac{1}{2} - \frac{1}{\lambda}\right]$ then $|\operatorname{round}(x, \mu, \lambda) - n| \leqslant e^{-\mu}$.*

The other very useful operation is the analog equivalent of a discrete assignment, done in a periodic manner. More precisely, we consider a particular class of ODEs

$$y'(t) = \operatorname{pereach}(t, \phi(t), y(t), g(t))$$

adapted from the constructions of [7].

This equation alternates between two behaviors, for all $n \in \mathbb{N}$.

- During $J_n = [n, n + \frac{1}{2}]$, it performs $y(t) \to \overline{g}$ where $\min_{t\in J_n} g(t) \leqslant \overline{g} \leqslant \max_{t\in J_n} g(t)$. So in particular, if $g(t)$ is almost constant over this time interval, then it is essentially $y(t) \to g$. Then $\phi$ controls how good the convergence is: the error is of the order of $e^{-\phi}$.

[4] With the obvious dimensional condition associated with each operation.

- During $J'_n = [n + \frac{1}{2}, n + 1]$, the systems tries to keep $y$ constant, ie $y' \approx 0$. More precisely, the system enforces that $|y'(t)| \leqslant e^{-\phi(t)}$.

▶ **Theorem 10** (Periodic reach). *There exists a generable function* $\operatorname{pereach} : \mathbb{R}^2_{\geqslant 0} \times \mathbb{R}^2 \to \mathbb{R}$ *such that for any* $I = [n, n+1]$ *with* $n \in \mathbb{N}$, $y_0 \in \mathbb{R}$, $\phi, \psi \in C^0(I, \mathbb{R}_{\geqslant 0})$ *and* $g \in C^0(I, \mathbb{R})$, *the unique solution to*

$$y(n) = y_0, \qquad y'(t) = \psi(t) \operatorname{pereach}(t, \phi(t), y(t), g(t))$$

*exists over* $I$.

- *If there exists* $\bar{g} \in \mathbb{R}$ *and* $\eta \in \mathbb{R}_{\geqslant 0}$ *such that* $|g(t) - \bar{g}| \leqslant \eta$ *for all* $t \in [n, n + \frac{1}{2}]$, *then* $|y(t) - \bar{g}| \leqslant \eta + \exp\left(-\int_n^t \psi(u)\phi(u)du\right)$ *whenever* $\int_n^t \psi(u)\phi(u)du \geqslant 1$ *for all* $t \in [n, n + \frac{1}{2}]$, *and* $|y(t) - \bar{g}| \leqslant \max(\eta, |y(n) - \bar{g}|)$ *for all* $t \in [n, n + \frac{1}{2}]$ *without condition.*
- *For all* $t \in [n + \frac{1}{2}, n]$, $|y(t) - y(n + \frac{1}{2})| \leqslant \int_{n+\frac{1}{2}}^t \psi(u) \exp(-\phi(u))\, du$.

*In particular, the first item implies that* $y(t) \geqslant \min_{u \in [n,t]} g(t) - \exp\left(-\int_n^t \phi(u)du\right)$ *whenever* $\int_n^t \phi(u)du \geqslant 1$ *for all* $t \in [n, n + \frac{1}{2}]$, *and* $y(t) \geqslant \min\left(y(n), \min_{u \in [n,t]} g(t)\right)$.

## 4 Generating fast growing functions

Our construction crucially relies on our ability to build functions of arbitrary growth. At the end of this section, we obtain a function `fastgen` with a straightforward specification: for any infinite sequence $a_0, a_1, \dots$ of positive numbers, we can find a suitable $\alpha \in \mathbb{R}$ such that $\operatorname{fastgen}(\alpha, n) \geqslant a_n$ for all $n \in \mathbb{N}$. Furthermore, we can ensure that $\operatorname{fastgen}(\alpha, \cdot)$ is increasing. Notice, and this is the key point, that the definition of `fastgen` is independent of the sequence $a$: a single generable function (and thus differential system) can have arbitrary growth by simply tweaking its initial value.

Our construction builds on the following lemma proved by [1], based on an example of [2]. The proof essentially relies on the function $\frac{1}{2 - \cos(x) - \cos(\alpha x)}$ which is generable and well-defined for all positive $x$ if $\alpha$ is irrational. By carefully chosing $\alpha$, we can make $\cos(x)$ and $\cos(\alpha x)$ simultaneously arbitrary close to 1.

▶ **Lemma 11** ([1]). *There exists a positive nondecreasing generable function* $g$ *and an absolute constant* $c > 0$ *such that for any increasing sequence* $a \in \mathbb{N}^{\mathbb{N}}$ *with* $a_n \geqslant 2$ *for all* $n$, *there exists* $\alpha \in \mathbb{R}$ *such that* $g(\alpha, \cdot)$ *is defined over* $[1, \infty)$ *and for any* $n \in \mathbb{N}$ *and* $t \geqslant 2\pi b_n$, $g(\alpha, t) \geqslant c a_n$ *where* $b_n = \prod_{k=0}^{n-1} a_k$.

Essentially, this lemma proves that there exists a function $g$ such that for any $n \in \mathbb{N}$, $g(\alpha, a_0 a_1 \cdots a_{n-1}) \geqslant a_n$. Note that this is not quite what we are aiming for: the function $g$ is indeed $\geqslant a_n$ but at times $a_0 a_1 \cdots a_{n-1}$ instead of $n$. Since $a_0 a_1 \cdots a_{n-1}$ is a very big number, we need to “accelerate” $g$ so that it reaches this values faster. This is a chicken-and-egg problem because to accelerate $g$, we need to build a fast growing function. We now try to explain how to solve this problem. Consider the following sequence:

$$x_0 = a_0, \qquad x_{n+1} = x_n g(x_n).$$

Then observe that

$$x_1 = x_0 g(x_0) = a_0 g(a_0) \geqslant a_0 a_1, \qquad x_2 = x_1 g(x_1) \geqslant a_0 a_1 g(a_0 a_1) \geqslant a_0 a_1 a_2, \qquad \dots$$

It is not hard to see that $x_n \geqslant a_0 a_1 \cdots a_n \geqslant a_n$. We then use our generable gadget of Section 3.2 to simulate this discrete sequence with a differential equation. Intuitively, we

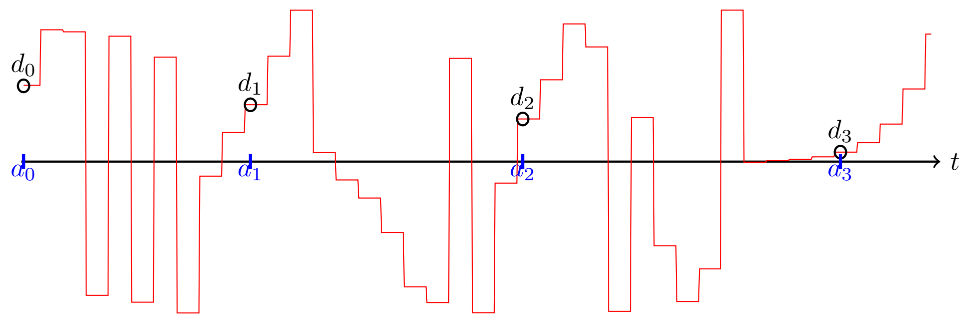


**Figure 2** Graph of dygen for $d_0 = 2^{-1}$, $d_1 = 2^{-3} + 2^{-1}$, $d_2 = 2^{-5} + 2^{-2}$ and $d_3 = 2^{-4}$ (other values ignored) assuming that $\delta = 9$. We get that $a_0 = 0$, $a_1 = 10$, $a_2 = 22$, $a_3 = 36$.

build a differential equation such that the solution $y$ satisfies $y(n) \approx x_n$. More precisely, we use two variables $y$ and $z$ such that over $[n, n+1/2]$, $z' \approx 0$ and $y(t) \to zg(z)$ and over $[n+1/2, n+1]$, $y' \approx 0$ and $z(t) \to y$. Then if $y(n) \approx z(n) \approx x_n$ then $y(n+1) \approx z(n+1) \approx x_{n+1}$.

▶ **Theorem 12.** *There exists* $\Gamma \subseteq \mathbb{R}$ *and a positive generable function* $\operatorname{fastgen} : \Gamma \times \mathbb{R}_{\geqslant 0} \to \mathbb{R}$ *such that for any* $x \in \mathbb{R}^{\mathbb{N}}_{\geqslant 0}$*, there exists* $\alpha \in \Gamma$ *such that for any* $n \in \mathbb{N}$ *and* $t \in \mathbb{R}_{\geqslant 0}$*,*

$$\operatorname{fastgen}(\alpha, t) \geqslant x_n \qquad \text{if } t \geqslant n.$$

*Furthermore,* $\operatorname{fastgen}(\alpha, \cdot)$ *is nondecreasing.*

## 5 Generating a sequence of dyadic rationals

A major part of the proof requires to build a function to approximate arbitrary numbers over intervals $[n, n+1]$. Ideally we would like to build a function that gives $x_0$ over $[0, 1]$, $x_1$ over $[1, 2]$, etc. Before we get there, we solve a somewhat simpler problem by making a few assumptions:

- we only try to approximate dyadic numbers, *i.e.* numbers of the form $m2^{-p}$, and furthermore we only approximate with error $2^{-p-3}$,
- if a dyadic number has size $p$, meaning that it can be written as $m2^{-p}$ but not $m'2^{-p+1}$ then it will take a time interval of $p$ units to approximate: $[k, k+p]$ instead of $[k, k+1]$,
- the function will only approximate the dyadics over intervals $[k, k+\frac{1}{2}]$ and not $[k, k+1]$.

This processus is illustrated in Figure 2: given a sequence $d_0, d_1, \ldots$ of dyadics, there is a corresponding sequence $a_0, a_1, \ldots$ of times such that the function approximate $d_k$ over $[a_k, a_k + \frac{1}{2}]$ within error $2^{-p_k}$ where $p_k$ is the size of $d_k$. The theorem contains an explicit formula for $a_k$ that depends on some absolute constant $\delta$.

Let $\mathbb{D}_p = \{m2^{-p} : m \in \{0, 1, \ldots, 2^p - 1\}\}$ and $\mathbb{D} = \bigcup_{n \in \mathbb{N}} \mathbb{D}_p$ denote the set of dyadic rationals in $[0, 1)$. For any $q \in \mathbb{D}$, we define its *size* by $\mathfrak{L}(q) = \min\{p \in \mathbb{N} : q \in \mathbb{D}_p\}$.

▶ **Theorem 13.** *There exists* $\delta \in \mathbb{N}_{>0}$*,* $\Gamma \subseteq \mathbb{R}^2$ *and a generable function* $\operatorname{dygen} : \Gamma \times \mathbb{R}_{\geqslant 0} \to \mathbb{R}$ *such that for any dyadic sequence* $q \in \mathbb{D}^{\mathbb{N}}$*, there exists* $(\alpha, \beta) \in \Gamma$ *such that for any* $n \in \mathbb{N}$*,*

$$|\operatorname{dygen}(\alpha, \beta, t) - q_n| \leqslant 2^{-\mathfrak{L}(q_n)-3} \qquad \text{for any } t \in [a_n, a_n + \tfrac{1}{2}]$$

*where* $a_n = \sum_{k=0}^{n-1}(\mathfrak{L}(q_k) + \delta)$*. Furthermore,* $|\operatorname{dygen}(\alpha, \beta, t)| \leqslant 1$ *for all* $\alpha, \beta$ *and* $t$*.*

# 6 Generating a sequences of bits

We saw in the previous section how to generate a dyadic generator. Unfortunately, we saw that it generates dyadic $d_n$ at times $a_n$, whereas we would like to get $d_n$ at time $n$ for our approximation. Our approach is to build a signal generator that will be high exactly at times $a_n$. Each the signal will be high, the system will copy the value of the dyadic generator to a variable and wait until the next signal. Since the signal is binary, we only need to generate a sequence of bits. Note that this theorem has a different flavour from the dyadic generator: it generates a more restrictive set of values (bits) but does so much better because we have better control of the timing and we can approximate the bits with arbitrary precision.

▶ Remark. Although it is possible to define `bitgen` using `dygen`, it does not, in fact, gives a shorter proof but definitely gives a more complicated function.

▶ **Theorem 14.** *There exists $\Gamma \subseteq \mathbb{R}$ and a generable function* $\operatorname{bitgen} : \Gamma \times \mathbb{R}^2_{\geqslant 0} \to \mathbb{R}$ *such that for any bit sequence $b \in \{0,1\}^{\mathbb{N}}$, there exists $\alpha_b \in \Gamma$ such that for any $\mu \in \mathbb{R}_{\geqslant 0}$, $n \in \mathbb{N}$ and $t \in [n, n+\frac{1}{2}]$,*

$$|\operatorname{bitgen}(\alpha_b, \mu, t) - b_n| \leqslant e^{-\mu}.$$

*Furthermore, $|\operatorname{bitgen}(\alpha, \mu, t)| \leqslant 1$ for all $\alpha, \mu$ and $t$.*

# 7 Generating an almost piecewise constant function

We have already explained the main intuition of this section in previous sections. Using the dyadic generator and the bit generator as a signal, we can construct a system that "samples" the dyadic at the right time and then holds this value still until the next dyadic. In essence, we just described an almost piecewise constant function. This function still has a limitation: its rate of change is small so it can only approximate slowly changing functions.

▶ **Theorem 15.** *There exists an absolute constant $\delta \in \mathbb{N}$, $p \in \mathbb{N}$, $\Gamma \subseteq \mathbb{R}^p$ and a generable function* $\operatorname{pwcgen} : \Gamma \times \mathbb{R}_{\geqslant 0} \to \mathbb{R}$ *such that for any dyadic sequence $q \in \mathbb{D}^{\mathbb{N}}$ then there exists $\alpha \in \Gamma$ such that for any $n \in \mathbb{N}$,*

$$|\operatorname{pwcgen}(\alpha, t) - q_n| \leqslant 2^{-\mathfrak{L}(q_n)} \qquad \textit{for any } t \in [a_n + \tfrac{1}{2}, a_{n+1}]$$

*and*[5]

$$\operatorname{pwcgen}(\alpha, t) \in \big[\operatorname{pwcgen}(\alpha, a_n), \operatorname{pwcgen}(\alpha, a_n + \tfrac{1}{2})\big] \qquad \textit{for any } t \in [a_n, a_n + \tfrac{1}{2}]$$

*where $a_n = \sum_{k=0}^{n-1}(\delta + \mathfrak{L}(q_k))$.*

# 8 Proof of the main theorem

The proof works in several steps. First we show that using an almost constant function, we can approximate functions that are bounded and change very slowly. We then relax all these constraints until we get to the general case. In the following, we only consider total functions over $\mathbb{R}$. See Remark on page 6 for more details.

[5] With the convention that $[a, b] = [\min(a, b), \max(a, b)]$.

▶ **Definition 16** (Universality). Let $I \subseteq \mathbb{R}$ and $\mathcal{C} \subseteq C^0(I) \times C^0(I, \mathbb{R}_{>0})$. We say that the universality property holds for $\mathcal{C}$ if there exists $d \in \mathbb{N}$ and a generable function $\mathfrak{u}$ such that for any $(f, \varepsilon) \in \mathcal{C}$, there exists $\alpha \in \mathbb{R}^d$ such that

$$|\mathfrak{u}(\alpha, t) - f(t)| \leqslant \varepsilon(t) \qquad \text{for any } t \in \operatorname{dom}(f).$$

▶ **Lemma 17.** *There exists a constant $c > 0$ such that the universality property holds for all $(f, \varepsilon)$ on $\mathbb{R}_{\geqslant 0}$ such that for all $t \in \mathbb{R}_{\geqslant 0}$:*

- *$\varepsilon$ is decreasing and $-\log_2 \varepsilon(t) \leqslant c' + t$ for some constant $c'$,*
- *$f(t) \in [0, 1]$,*
- *$|f(t) - f(t')| \leqslant c\varepsilon(t + 1)$ for all $t' \in [t, t + 1]$.*

**Proof Sketch.** This is essentially a application of `pwcgen` with a small twist. Indeed the bound on $f$ guarantees that dyadic rationals are enough. The bound on the rate of change of $f$ guarantees that a single dyadic can provide an approximation for a long enough time. And the bound on $\varepsilon$ guarantees that we do not need too many digits for the approximations. ◀

▶ **Lemma 18.** *The universality property holds for all $(f, \varepsilon)$ on $\mathbb{R}_{\geqslant 0}$ such that $f$ and $\varepsilon$ are differentiable, $\varepsilon$ is decreasing and $f(t) \in [0, 1]$ for all $t \in \mathbb{R}_{\geqslant 0}$.*

**Proof Sketch.** Consider $F = f \circ h^{-1}$ and $E = \varepsilon \circ h^{-1}$ where $h$ is a fast-growing function like `fastgen`. Then the faster $h$ grows, the slower $E$ and $F$ change and thus we can apply Lemma 17 to $(F, E)$. We recover an approximation of $f$ from the approximation of $F$. ◀

▶ **Lemma 19.** *The universality property holds for all $(f, \varepsilon)$ on $\mathbb{R}_{\geqslant 0}$ such that $f$ is differentiable and $\varepsilon$ is decreasing.*

**Proof Sketch.** Consider $F = \frac{1}{2} + \frac{f}{h}$ and $\varepsilon = \frac{f}{h}$ where $h$ is a fast-growing function like `fastgen`. By taking $h$ big enough, we can ensure that $F(t) \in [0, 1]$ and apply Lemma 18 to $(F, E)$. We then recover an approximation of $f$ from the approximation of $F$. ◀

▶ **Lemma 20.** *The universality property holds for all continuous $(f, \varepsilon)$ on $\mathbb{R}_{\geqslant 0}$.*

**Proof Sketch.** Observe that the set of differential functions is dense in the set of continuous functions and apply Lemma 19. ◀

▶ **Lemma 21.** *The universality property holds for all continuous $(f, \varepsilon)$ on $\mathbb{R}$.*

**Proof Sketch.** Let $f^+$ be the approximating of $f$ over $\mathbb{R}_{\geqslant 0}$, using Lemma 20. Then we extend and modify $f^+$ to $\mathbb{R}$ in such a way that the approximation is still good over $\mathbb{R}_{\geqslant 0}$ but the function almost vanishes over $(-\infty, 0]$. We then do the same to $t \mapsto f(-t) - f^+(-t)$ and sum the two functions. ◀

We can now show the main theorem.

**Proof of Theorem 3.** Lemma 21 gives a generable function $\mathfrak{u}$. There exists $\alpha$ such that

$$|\mathfrak{u}(\alpha, t) - f(t)| \leqslant \varepsilon(t).$$

And since $\mathfrak{u}$ is generable, $t \mapsto \mathfrak{u}(\alpha, t)$ satisfies a PIVP. ◀

## References


1 Steven B. Bank. Some results on analytic and meromorphic solutions of algebraic differential equations. *Advances in Mathematics*, 15(1):41 – 62, 1975. doi:10.1016/0001-8708(75)90124-3.

2 N. M. Basu, S. N. Bose, and T. Vijayaraghavan. A simple example for a theorem of vijayaraghavan. *Journal of the London Mathematical Society*, s1-12(4):250–252, 1937. doi:10.1112/jlms/s1-12.48.250.

3 Emile Borel. Mémoire sur les séries divergentes. *Annales Scientifiques de l'Ecole Normale Supérieure*, 16:9–136, 1899.

4 Michael Boshernitzan. Universal formulae and universal differential equations. *Annals of mathematics*, 124(2):273–291, 1986.

5 Olivier Bournez, Manuel L. Campagnolo, Daniel S. Graça, and Emmanuel Hainry. Polynomial differential equations compute all real computable functions on computable compact intervals. *Journal of Complexity*, 23(3):317–335, June 2007. doi:10.1016/j.jco.2006.12.005.

6 Olivier Bournez, Daniel S. Graça, and Amaury Pouly. On the functions generated by the general purpose analog computer. *CoRR*, abs/1602.00546, 2016. URL: http://arxiv.org/abs/1602.00546.

7 Olivier Bournez, Daniel S. Graça, and Amaury Pouly. Polynomial Time corresponds to Solutions of Polynomial Ordinary Differential Equations of Polynomial Length. The General Purpose Analog Computer and Computable Analysis are two efficiently equivalent models of computations. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 109:1–109:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.109.

8 Keith Briggs. Another universal differential equation. *arXiv preprint math/0211142*, 2002.

9 V. Bush. The differential analyzer. A new machine for solving differential equations. *J. Franklin Inst.*, 212:447–488, 1931.

10 D. C. Carothers, G. E. Parker, J. S. Sochacki, and P. G. Warne. Some properties of solutions to polynomial systems of differential equations. *Electron. J. Diff. Eqns.*, 2005(40), April 2005.

11 Etienne Couturier and Nicolas Jacquet. Construction of a universal ordinary differential equation $\rfloor^\infty$ of order 3. *arXiv preprint arXiv:1610.09148*, 2016.

12 Richard J. Duffin. Rubel's universal differential equation. *Proceedings of the National Academy of Sciences*, 78(8):4661–4662, 1981.

13 Daniel S. Graça and José Félix Costa. Analog computers and recursive functions over the reals. *Journal of Complexity*, 19(5):644–664, 2003.

14 D. S. Graça, J. Buescu, and M. L. Campagnolo. Computational bounds on polynomial differential equations. *Appl. Math. Comput.*, 215(4):1375–1385, 2009.

15 D. S. Graça, M. L. Campagnolo, and J. Buescu. Computability with polynomial differential equations. *Adv. Appl. Math.*, 40(3):330–349, 2008.

16 G. H. Hardy. Some results concerning the behaviour at infinity of a real and continuous solution of an algebraic differential equation of the first order. *Proceedings of the London Mathematical Society*, 2(1):451–468, 1912.

17 R. J. Lipton and K. W. Regan. The amazing zeta code. Post on Blog "Gödel's Lost Letter and P=NP", https://rjlipton.wordpress.com/2012/12/04/the-amazing-zeta-code/, December 4, 2012.

18 M. B. Pour-El. Abstract computability and its relations to the general purpose analog computer. *Trans. Amer. Math. Soc.*, 199:1–28, 1974.

19 L. A. Rubel. A universal differential equation. *Bulletin of the American Mathematical Society*, 4(3):345–349, May 1981.
20 C. E. Shannon. Mathematical theory of the differential analyser. *Journal of Mathematics and Physics MIT*, 20:337–354, 1941.
21 T. Vijayaraghavan. Sur la croissance des fonctions définies par les équations différentielles. *CR Acad. Sci. Paris*, 194:827–829, 1932.