Solving Analytic Differential Equations in Polynomial Time over Unbounded Domains

Amaury Pouly

École normale supérieure de Lyon, LIP, 46 allée d'Italie, 69008 Lyon, France
amaury.pouly@ens-lyon.fr

We consider the following initial-value problem defined by an ODE

$$\begin{cases} x' = f(x) \\ x(0) = x_0 \end{cases} \tag{1}$$

where f is defined in some (possibly unbounded) domain.

In this paper we show that if $f : \mathbb{R}^n \to \mathbb{R}^n$ admits an analytic extension over $\mathbb{C}^n$ and $x : \mathbb{R} \to \mathbb{R}$ admits an analytic extension over $\mathbb{R}$ and both satisfies a very generous assumption about its growth rate, this solution can be computed in polynomial time from f and x_0 over $\mathbb{R}$. Actually, our constructions also works when considering solutions over $\mathbb{C}$ and assuming $f : \mathbb{C}^n \to \mathbb{C}^n$ analytic. Notice that, as it is well known, Equation (1) covers the case of an ODE of type $x' = f(t, x)$, as this latter case can be reduced to (1) by using a new variable x_{n+1} satisfying $x'_{n+1} = 1$.

Motivation 1 & Digression: Analog models of computation. We got to this result by trying to understand whether analog continuous-time models of computations do satisfy (some variant) of Church-Turing thesis: as such systems can usually be described by particular classes of ordinary differential equations, understanding whether they can compute more than Turing machines is equivalent to understanding whether they can always be simulated by Turing machines.

For example, the most well known example of analog model of computations is the General Purpose Analog Computer (GPAC) introduced by Claude Shannon in [7] as the idealization of an analog computer, the Differential Analyzer [1]. Shannon worked as an operator early in its career on these machines.

As it can be proved [4, 7] that any GPAC can be described by an ordinary differential equation of the form of (1) with f componentwise polynomial, proving that the GPAC does satisfy the Church-Turing thesis is equivalent to proving that solutions of such an ODE is always computable. It has been proved only recently that this holds for the case of f componentwise polynomial [5], [2]. Hence, the GPAC do satisfy the Church-Turing thesis. Notice that computability of solutions doesn't hold for general computable f [6], but in general known uncomputability results can only be obtained when the system is "ill behaved" (e.g. non-unique solutions in [6]). These kind of phenomena does not appear in models physically inspired by true machines like the GPAC.

Here, we are dealing with the next step. Do analog models like the GPAC satisfy the *effective* (in the sense of computable complexity) version of Church Turing thesis: all (sufficiently powerful) "*reasonable*" models of computations with "*reasonable*" measure of time are polynomially equivalent. In other words, we want to understand whether analog systems can provably (not) compute faster than usual classical digital models like Turing machines.

Taking time variable t as a measure of time (which is the most natural measure), proving that the GPAC can not compute more than Turing machines would require to prove that solutions of ODE (1) are always computable (in the classical sense) in a time polynomial in t, for f (componentwise) polynomial.

We here don't get exactly this result: for f componentwise polynomial, corresponding to GPACs, f is clearly analytic. But here, in our results, we have to suppose furthermore f to admit an analytic extension over $\mathbb{C}^n$. Although this case is stronger that when f is real analytic (it is well known that analyticity in the complex plane implies analyticity over the real line, but that the converse direction does not hold), we believe that our results are interesting on their own and provide a promising step towards the case of the real line.

Motivation 2: Recursive analysis. The results obtained in this paper turn out to be new and not known in a recursive analysis or classical computability or complexity perspective.

Being able to compute efficiently solutions of general ordinary differential equations is clearly of interest. Observe that all usual methods for numerical integrations (including basic Euler's method, Runge Kutta's methods, ...) do not provide the value of $x(t)$ in a time polynomial in t, whereas we do for general analytic functions under our hypotheses. Actually, as all these numerical methods falls in the general theory of n-order methods for some n, this is possible to use this theory (developed for example in [3]) to prove that none of them produce the value of $x(t)$ in a time polynomial in t. This has been already observed in [8], and claimed possible in [8] for some classes of functions by using methods of order n with n depending on t, but without a full proof. The method proposed here is different from the ideas proposed in this latter paper but prove that this is indeed possible to produce $x(t)$ in a time polynomial in t.

References

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

[2] P. Collins and D. S. Graça. Effective computability of solutions of differential inclusions — the ten thousand monkeys approach. *Journal of Universal Computer Science*, 15(6):1162–1185, 2009.

[3] Jean-Pierre Demailly. *Analyse Numérique et Equations Différentielles*. Presses Universitaires de Grenoble, 1991.

[4] D. S. Graça and J. F. Costa. Analog computers and recursive functions over the reals. *J. Complexity*, 19(5):644–664, 2003.

[5] D.S. Graça, N. Zhong, and J. Buescu. Computability, noncomputability and undecidability of maximal intervals of IVPs. *Trans. Amer. Math. Soc.*, 361(6):2913–2927, 2009.

[6] M. B. Pour-El and J. I. Richards. A computable ordinary differential equation which possesses no computable solution. *Ann. Math. Logic*, 17:61–90, 1979.

[7] C. E. Shannon. Mathematical theory of the differential analyzer. *J. Math. Phys. MIT*, 20:337–354, 1941.

[8] Warren D. Smith. Church's thesis meets the N-body problem. *Applied Mathematics and Computation*, 178(1):154–183, 2006.