Proposal for L3 Research Internship for ENS Lyon for 2023

Programming with Ordinary Differential Equations and Continuous Time: Towards a programming language or pseudo-programming language.

Olivier Bournez

Keywords: Continous time models of computation. Analog machines and models. Ordinary Differential Equations. Computability. Complexity.

Superviser: Olivier Bournez.

Administrative Position: Professor (of Computer Science) at Ecole Polytechnique.

Location: Laboratoire d'Informatique de l'X, LIX, Ecole Polytechnique , 91128 Palaiseau Cedex

Phone: +33 (0)1 77 57 80 78

Email: olivier.bournez@lix.polytechnique.fr.

Www: http://www.lix.polytechnique.fr/~bournez.

Language: French or English. This proposal is intentionally written in English for Non-French speaking students who may be interested.

General Introduction

It has been understood quite recently that it is possible to program with Ordinary Differential Equations (ODEs). This actually was obtained as a side effect from attempts to relate the computational power of analog computational models to classical computability. The former are working over continuous data (for example voltage, or concentations) with a continuous time (ex: according to a system of ordinary differential equations corresponding to some analog electronic circuits, or the kinetic of reactions), unlike the latter, such as Turing machines, working on discrete entities such as words, with a discrete time.

Refer to [6, 3, 14] for surveys on analog computation with a point of view based on computation theory aspects, relating these models to discrete models, or to [17, 16, 13] for surveys discussing historical and technological aspects related to analog models of computation. Analog models of computation include in particular the differential analyzers.

This has been obtained by realizing that continuous time processes defined by ODEs, and even defined by polynomial ODEs, can simulate various discrete time processes. They hence can be used to simulate models such as Turing machines [7, 11], and even some more exotic models working with a discrete time but over continuous data. This is based on various refinements of constructions done in [7, 11, 12, 15, 9]. We call this *ODE programming*, as this is indeed some kind of programing with various continuous constructions.

Forgetting analog machines or models of computation, it is important to realize that ODEs is a kind of universal language of mathematics that is used in many, if not all, experimental sciences: Physics, Biology, Chemistry, Consequently, once it is known that one can program with ODEs, many questions asking about universality, or computations, in experimental contexts can be solved. This is exactly what has been done by several authors, including ourselves, to solve various open problems, in various contexts such as applied maths, computer algebra, biocomputing...

Applications include: Characterization of computability and complexity classes using ODEs [12, 11, 15, 9]; Proof of the existence of a universal (in the sense of Rubel) ODE [5]; Proof of the strong Turing completeness of biochemical reactions [8], or more generally various statements about the completeness of reachability problems (e.g. PTIME-completeness of bounded reachability) for ODEs [4].

Description of the work

However, when going to say more about proofs, their authors including ourselves, often feel frustrated: Currently, the proofs are mostly based on technical lemmas and constructions done with ODEs, often mixing both the

ideas behind these constructions, with numerical analysis considerations about errors and error propagation in the equations. We believe this is one factor hampering a more widespread use of this technology in other contexts.

The article [2, 1] are born from an attempt to popularize this ODE programming technology to a more general public, and in particular master and even undergraduate students. We show how some constructions can be reformulated using some notations, that can be seen as a pseudo programming language. This provides a way to explain in an easier and modular way the main intuitions behind some of the constructions, focusing on the algorithm design part. We focused here, as an example, on how the proof of the universality of polynomial ODEs (a result due to [10], and fully developed in [11]) can be reformulated and presented.

The purpose of the internship will be to develop this approach, in order to reformulate or improve some of the constructions. New results include the characterization of complexity classes with ordinary differential equations, or applications in experimental sciences.

Comment

The actual topic of the work is related to computability and complexity theory. This requires only common and basic knowledge in ordinary differential equations, or in computability or complexity theory, at the level of 1st year of ENS Lyon. Most of the intuitions of our today's constructions come from classical computability and complexity.

Possibilities of funding according to the administrative situation of candidates.

The subject can also be adapted according to requests, knowledge and skills of candidates. Please contact me if interested or in case of questions.

References

- [1] Olivier Bournez. Informatique Mathématique Une photographie en 2022. Cours donnés à l'Ecole Jeunes Chercheurs en Informatique Mathématiques., chapter Le Calcul Analogique.
- [2] Olivier Bournez. Programming with ordinary differential equations: Some first steps towards a programming language. In Ulrich Berger, Johanna N. Y. Franklin, Florin Manea, and Arno Pauly, editors, *Revolutions and Revelations in Computability - 18th Conference on Computability in Europe, CiE 2022, Swansea, UK, July 11-15, 2022, Proceedings*, volume 13359 of *Lecture Notes in Computer Science*, pages 39–51. Springer, 2022.
- [3] Olivier Bournez and Manuel L. Campagnolo. New computational paradigms. changing conceptions of what is computable. chapter A Survey on Continuous Time Computations, pages 383–423. Springer-Verlag, New York, 2008.
- [4] Olivier Bournez, Daniel S. Graça, and Amaury Pouly. Polynomial Time corresponds to Solutions of Polynomial Ordinary Differential Equations of Polynomial Length. *Journal of the ACM*, 64(6):38:1–38:76, 2017.
- [5] Olivier Bournez and Amaury Pouly. A universal ordinary differential equation. Logical Methods in Computer Science, 16(1), 2020.
- [6] Olivier Bournez and Amaury Pouly. A survey on analog models of computation. In Handbook of Computability and Complexity in Analysis, pages 173–226. Springer, 2021.
- [7] M. S. Branicky. Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theoretical Computer Science*, 138(1):67–100, 6 February 1995.
- [8] Francois Fages, Guillaume Le Guludec, Olivier Bournez, and Amaury Pouly. Strong turing completeness of continuous chemical reaction networks and compilation of mixed analog-digital programs. In *Computational Methods in Systems Biology-CMSB 2017*, 2017.
- [9] Riccardo Gozzi. *Analog Characterization of Complexity Classes*. PhD thesis, Instituto Superior Técnico, Lisbon, Portugal and University of Algarve, Faro, Portugal, 2022.
- [10] D. S. Graça, M. L. Campagnolo, and J. Buescu. Computability with polynomial differential equations. Adv. Appl. Math., 40(3):330–349, 2008.
- [11] Daniel S. Graça. *Computability with Polynomial Differential Equations*. PhD thesis, Instituto Superior Técnico, 2007.
- [12] Emmanuel Hainry. *Modèles de calculs sur les réels. Résultats de Comparaisons*. PhD thesis, LORIA, 7 Décembre 2006.

- [13] Bruce J MacLennan. Analog computation. In *Encyclopedia of complexity and systems science*, pages 271–294. Springer, 2009.
- [14] Pekka Orponen. A survey of continous-time computation theory. In D.-Z. Du and Ker-I Ko, editors, *Advances in Algorithms, Languages, and Complexity*, pages 209–224. Kluwer Academic Publishers, 1997.
- [15] Amaury Pouly. Continuous models of computation: from computability to complexity. PhD thesis, Ecole Polytechnique and Unidersidade Do Algarve, Defended on July 6, 2015. 2015. https://pastel.archivesouvertes.fr/tel-01223284, Prix de Thèse de l'Ecole Polyechnique 2016, Ackermann Award 2017.
- [16] Bernd Ulmann. Analog computing. Walter de Gruyter, 2013.
- [17] Bernd Ulmann. Analog and hybrid computer programming. De Gruyter Oldenbourg, 2020.