# ANALOG CHARACTERIZATION OF COMPLEXITY CLASSES

**Riccardo Gozzi**
Instituto Superior Técnico, Universidade de Lisboa
Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
& Instituto de Telecomunicações, Lisboa, Portugal
**Daniel S. Graça**
Universidade do Algarve
C. Gambelas, 8005-139 Faro, Portugal
& Instituto de Telecomunicações, Lisboa, Portugal

The first time ODEs have been considered has possible paradigm of computation was in 1941, when Claude Shannon introduced his model called General Purpose Analog Computation model, or GPAC. The goal of this model was to provide a theoretical foundation that could describe the behavior of differential analyzers: analog machines in use in the 1940's composed by complex systems of disks and wheels that could compute simple operations such as addition, subtraction or multiplication between functions, as well as solving some basic ODEs. The model has been later modified by Marian Boykan Pour-El in [1] and then generalized by Daniel Graça and Jos Flix Costa in [6] in order to include multivariate functions. The last formulation concretely established a connection between the GPAC model and polynomial initial value problems, or PIVPs; indeed in [6] the authors proved that all functions generated by this analog model of computation can be obtained as components of solutions of PIVPs. Based on this result, it started to emerge the idea of using PIVPs and polynomial ODEs to simulate the behavior of other models of computation used to describe continuous objects, such as computable analysis [2]. These investigations led O. Bournez, M.L. Campagnolo, D.S. Graa and E. Hainry to show in [7] that oracle Turing machines used in computable analysis can be simulated just by using specific systems of ODEs, and vice versa, therefore establishing a full equivalence between the two different models on a computability level. Nevertheless, this result did not contain any information on how to take in account complexity when comparing the two models. More generally, analog models of computation are very delicate from a complexity perspective, since many different complexity measures have been suggested for the same models, and changing the complexity measure can give rise to unexpected phenomenons, such as super-Turing behaviors [4], [5]. One problematic phenomenon that is common for different analog models is the Zeno phenomenon, which allows to exploit the continuity of the models in order to arbitrarily contract (or enlarge) the working time interval of the computations [3]. A key intuition for an effective definition of a complexity measure on this updated version of the GPAC model came in [9], where the authors decided to consider the length of the solution of the dynamical systems involved as criterion to measure complexity. The dynamical systems involved included only polynomial ODEs with polynomially bounded solution, and these solution were made converging to the correct outcome of the computation only once a polynomial length was reached. This idea could rule out unnatural behaviors connected with the Zeno phenomenon, since the length of the solution is invariant over rescaling time and space boundaries defining the system. In this model, the space boundary is represented by the maximum value of the norm of the solution, and the time boundary by the moment in which the solution starts converging to the correct outcome. With this definition, the authors of [9] managed to describe all functions (and sets) computable (decidable) in polynomial time by means of systems of polynomial ODEs, therefore establishing an interesting connection between a discrete model of computation (the Turing machine model) and an analog model of computation (the GPAC model) on a polynomial complexity level. More precisely, the way this connection is performed is through the notion of *emulation*. Given a discrete function $f$ belonging to a standard complexity class, we say that $f$ is *emulable* by a function $g$ in a certain analog class (in our case the class of solutions of polynomial ODEs) if and only if $\Psi(f(w)) = g(\Psi(w))$, where $\Psi$ is a specific encoding of discrete words $w$ from a fixed, finite alphabet into rational numbers. This result represented a full analog characterization of the classes FP and P as well as the class of functions computable in polynomial time in the sense of computable analysis. Nevertheless, the main construction they developed in the article was relying on particular properties of polynomials, such as closure by composition, and in principle it was not possible to repeat the same proofs in order to characterize greater complexity classes, such as FEXPTIME or EXPTIME. With our work we found a solution that could circle around this problem, introducing a modification on the core definitions of the analog classes included in [9] that splits the boundaries into the product of two parts, one exponential (in the initial condition of the polynomial ODE), one polynomial (in the other parameters used to define the continuous computation – time and accuracy). More precisely, we were able to correctly identify which part of the construction had to remain polynomial in order to repeat the proofs of [9], and which part had to be raised to exponential to obtain the new characterization. This procedure not only allowed us to completely characterize exponential complexity classes, but also led us to identify and list all the necessary conditions that the considered analog boundaries have to satisfy in order to obtain an equivalence with a discrete complexity class. In this way we have been able to generalize the characterization process and successfully apply it to describe the class of all elementary functions.

What we discussed until now referred to time complexity classes. A path that was still unexplored was the case of space complexity classes. In this particular case, the encoding that has been used for the continuous simulation of Turing machine in [9] is not helpful anymore. Indeed, by encoding configurations of the machine into rational numbers between 0 and 1 we loose information related to the length of the working tape visited by the machine. Therefore, we had the idea of making use of another, more robust, encoding taken from [8], and we applied it to construct a dynamical system of ODEs that could capture the essential features of polynomial space bounded Turing machine computation. This study revealed that one essential property that the dynamical has to satisfy is a strong robustness to perturbation, in order to allow us to sample different values of the solution at different times without destroying the simulation. This condition is natural, since it mimics the stability property of Turing machine computations, where saving the content of the working tape at any moment does not ruin the final outcome of the computation. In this way we succeeded on providing a full description of the classes FPSPACE and PSPACE by means of systems of polynomial ODEs, enlarging the possibilities of this research area.

## References


[1] M. B. Pour-El, Abstract computability and its relations to the general purpose analog computer, Trans. Amer. Math. Soc. 199 (1974), 128, 1974.

[2] Ker-I Ko, Complexity Theory of Real Functions, Progress in Theoretical Computer Science. Birkhaser, Boston, 1991.

[3] C. Moore, Recursion theory on the reals and continuous-time computation. Theoretical Computer Science, 162(1):23 44, 1996.

[4] E.B. Davies, Building infinite machines, The British Journal for the Philosophy of Science, 52:671682, 2001.

[5] B.J. Copeland, Accelerating Turing machines, Minds and Machines, 12:281- 301, 2002.

[6] D.S. Graa and J.F. Costa, Analog computers and recursive functions over the reals, Journal of Complexity, 19(5):644664, 2003.

[7] O. Bournez and M.L. Campagnolo and D.S. Graa and E. Hainry, The general purpose analog computer and computable analysis are two equivalent paradigms of analog computation, International Conference on Theory and Applications of Models of Computation, Springer, Berlin, Heidelberg, 2006.

[8] D.S. Graa and M.L. Campagnolo and J. Buescu, Computability with polynomial differential equations, Advances in Applied Mathematics, Elsevier, 2008.

[9] O. Bournez and D.S. Graa and A. Pouly, Polynomial time corresponds to solutions of polynomial ordinary differential equations of polynomial length, Journal of the ACM (JACM), 2017.