Chapter 5

Conclusions and perspectives

In this document, we presented several new results, we summarized the state of art of several research domains, and we presented some points of view.

First, in chapter 1 and 2, we recalled some of the complexity and richness of continuous dynamical systems. We presented several complex systems, with interesting and sometime subtle dynamics.

The examples from chapter 1 are almost all from the excellent book [Hirsch et al., 2003]. The examples from chapter 2 illustrate models of systems with some concurrency, coming from various sources such as bioinformatics, virology, and computer virology. It presents also several models coming from game theory, or linked with distributed algorithms.

By all these examples, we wanted to outline several facts. First, all these systems correspond to a particular class of ordinary differential equations, that we named polynomial Cauchy problems. We come back to this class of systems several times in other chapters. Second, many natural or artificial systems, can intrinsically be considered as computational models. Third, we argued through several examples, that even for discrete space and time systems, the good level of abstraction is often the continuous.

More specifically, in chapter 2, one of our purposes is to show the interest and the potential of the use of continuous models for distributed algorithmic. We discussed several problems that appear then.

In Chapter 3, we presented a survey, which we estimate to be rather complete, of the theory, or of the theories, of continuous time systems. We showed that these theories allow both to understand the hardness of questions related to continuous time dynamical systems, and the power of continuous time analog models. From these two perspectives, the obtained and presented results are motivated by various domains such as verification, control theory, VLSI design, neural networks, numerical analysis, or recursion theory over the reals. We discussed both the computability, the complexity, as well as the robustness to noise and imprecision aspects, letting a wide part in the text to discussions about open problems and perspectives.

The chapter 4, 5, and the appendix A present synthetic overview of some of our results.

In the chapter 4, we showed that this is possible to relate the functions solution of a polynomial Cauchy problem to GPAC computable functions, that was known, but also to computable functions in the sense of recursive analysis: the functions that are GPAC computable, in a natural sense, are precisely computable functions in the sense of recursive analysis. In a complementary way, we also show that this is possible to relate computable (and elementary computable) functions in the sense of recursive analysis to $\mathbb{R}$-recursive functions. As far as we know, this is the first time that a so-natural algebraic characterization of computable functions of recursive analysis is provided.

In Chapter 5, we presented a catalogue of our characterizations of complexity classes in the Blum Shub Smale model. All these characterizations are in the spirit of polynomial time characterization of Bellantoni and Cook in [Bellantoni and Cook, 1992]. We presented a characterization of almost all the complexity classes considered in the monograph [Blum et al., 1998] on Blum Shub Smale model (actually, only parallel complexity class NC is missing).

In Appendix A, we discussed the possible superpower of continuous systems with respect to classical models, such as Turing machines. We presented several recurrent incorrect understandings of what Church thesis says, and we characterize the power of several classes of continuous systems. This appendix contains several new results, and can also be considered as an upgrade of results that existed when we started our PhD.

Perspectives

Each of the chapters, is voluntarily written with an important part let to perspectives: in particular the chapter 3 has a whole section devoted to perspectives and interesting future work related to continuous time systems.

But actually, the chapters 1, and 2 also contain numerous questions that are explicitly quoted as such. Each of these questions corresponds to several years of research work. The chapter 4 has also a conclusion and perspective part with several promising directions for future work.

In front of all these questions, we now outline the one that we think the most essential today.

Understand if there is a unifying concept for continuous time systems such as Church-Turing thesis.

The situation for continuous time systems is far from being so clear than for discrete systems. Although it has been shown that some continuous time models exhibit super Turing power, these results rely on the use of an infinite amount of resources such as time, space, precision, or energy. In general, it is believed that "reasonable" continuous time models cannot compute beyond Turing machines.

Since analytic and robust continuous time systems can simulate Turing machines in an unbounded state space, we believe that digital computation and analog continuous time computation are equally powerful from the computability point of view. Furthermore, as we showed, several recent results establish the equivalence between functions solution of polynomial Cauchy problems, GPAC computable functions, and computable functions in the sense of recursive analysis. These kinds of results reinforce the idea that there could be a unified framework for continuous time computations, analogous to what occurs in classical computation theory.

We argued in several parts of this document about the modelling power and the remarkable properties of functions solution of polynomial Cauchy problems. We believe that this class of functions could provide a real candidate for a general paradigm of continuous time computation. We believe that this idea deserves further investigation.

Understand if there is a nice and robust theory of complexity for continuous time systems.

We showed in Chapter 3, that several directions have been explored in literature to build complexity theories for continuous time systems. Actually, for now, there has not been an agreement between authors on basic definitions such as computation time or input size. The results obtained at this day are mostly derived from concepts that are intrinsic to the continuous time systems under study.

As computable analysis is a well-established and understood framework for the study of computational complexity of continuous time systems, we believe that better understanding relations between different approaches and computable analysis from a complexity point of view is of first importance.

In relation with Chapter 4, we believe that a way to tackle the problem is to try to characterize the polynomial time computable functions in recursive analysis as a class of $\mathbb{R}$-recursive functions. Can we extend the characterization à la Bellantoni and Cook [Bellantoni and Cook, 1992] to recursive analysis? We showed that GPAC computable functions correspond to computable functions in the sense of recursive analysis. The simulation that is used in the proof seems to closely relate the computation time of the involved GPAC and of the Turing machine. Can we establish such a correspondence at the complexity level, and not only at the computability level?

If the class of functions solutions of a polynomial Cauchy problem allows, as we conjecture above, to characterize in an elegant way the notion of "reasonable" computations in continuous time, can we formulate in a simple way a complexity theory based on these functions?

Better understand models, and their properties.

We have seen in Chapter 3 that only a few works have been devoted to the effect of noise and imprecisions on continuous time computations. At this day, related results concern mainly only discrete time systems.

We have discussed several ways to model noise and imprecision: for example, using a probabilistic model of noise, or a non-deterministic model of noise. We have shown that each of this way to model noise do lead to really very different results. Better understanding all that seems fundamental.

For example, numerous questions arise when one asks if undecidability results for continuous time systems still hold for robust systems. This is of first importance for example for the verification of hybrid systems, since this question is closely related to the question of termination of automatic verification procedures. Better understandings of the hypothesis under which noise yield decidability or undecidability seem of fundamental importance.

Furthermore, we think that up to this day the question has been addressed only at the computability level, and not at the complexity level. When problems are proved decidable, what is the effect of these notions of noise on the complexity of problems? How does it increase with precision and noise?

We think that these questions are at the heart of fundamental problems linked with very deep questions about our today's models of the physical world: how to model noise? What is a robust model? How to model imprecision? How pertinent, and what is the sense, of trying to do formal proof, hence somehow certain, about uncertain phenomena or systems?

Use continuous systems in distributed algorithmic.

We voluntarily insisted in Chapter 2 on the fact that continuous systems arise as soon as one discusses huge populations.

At this day, distributed algorithmic is not considered with these tools. A fascinating question is to understand if the theories of computations with continuous systems can help to understand, program, and control highly parallel models. Given the size of today's networks, and the call in question of several strong hypotheses of classical distributed algorithmic in some applications such as sensor networks models, we think this is urgent to put some attention to this question.

We presented in Chapter 2 several models that go in this direction. For example, we showed that the model of population protocol have some original computational properties in term of relations that can be defined in Presburger's arithmetic. We showed that these protocols can be generalized to talk about huge populations. The exact power of models in this spirit remains very widely to be understood.

More generally, to determine if continuous systems can contribute in a deep way to distributed algorithmic, several research directions seem priority.

First, one must understand well and better the hypotheses that makes it possible to go, in domains such as physics, biology, virology, computer virology, from a discrete system to a continuous abstraction, to understand in a fine way when this is possible in distributed algorithmic. Going from discrete to continuous on distributed algorithms does lead to specific problems. For example, up to which point can we ignore the topological informations, as this is done in many population models?

Considering that in a distributed system of huge size, one can hope to program or control at better at most a part of the agents, we think that models based on game theory provide valuable tools to do so. These models are intrinsically continuous, and the recent literature abounds of examples of systems that can be modelled naturally through these tools.

However, at this day, this theory has been developed in a mathematical framework, rather abstract, and that mixes sometime with difficulties with classical models and concepts from distributed algorithmic.

For example, in distributed algorithmic, the evaluation of the complexity of algorithms is often realized in the worst case. Worst case corresponds to some notion of adversary, but which is rather different from the one that is used in game theory. Succeeding in mixing the notions of adversary of game theory and classical algorithmic is a true challenge.

Furthermore, up to this day, there is almost now works on the dynamic aspects. Game theory allows discussing the notions of rational equilibrium, but not dynamism. Models such as the one that are mentioned in Chapter 2, based on repeated games or evolutionary game theory are conceived to model dynamism of situations of concurrency. But almost no applications of these theories to distributed algorithmic has been done at this day. Succeeding in modelling the dynamism, to build better distributed algorithms, is also a true challenge of first importance.

We think that the set of the results in this document can help in a significant way to the understanding of today's and tomorrow's algorithmic. This is the sense of several of our current research investigations about the use of algorithmic game theory for distributed algorithmic.

In particular, we strongly believe that the future spectrum of the applications of continuous time systems is far from being limited to the list of today's applications that is listed in Chapter 3.

Bibliography

[Bellantoni and Cook, 1992] Bellantoni, S. and Cook, S. (1992). A new recursion-theoretic characterization of the poly-time functions. *Computational Complexity*, 2(2):97–110.

[Blum et al., 1998] Blum, L., Cucker, F., Shub, M., and Smale, S. (1998). *Complexity and Real Computation.* Springer-Verlag.

[Hirsch et al., 2003] Hirsch, M. W., Smale, S., and Devaney, R. (2003). *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Elsevier Academic Press.