$\frac{\partial i}{\partial}$FFERENCE

Complexity theory with discrete ODEs
Théorie de la complexité avec des équations différentielles discrètes

Summary table of persons involved in the project:

(PSL stands for "*Partner's Scientific Leader*").

Partner	Name	Current position	Role & responsibilities in the project	Tasks	Involvement
École Polytechnique	Bournez Olivier	Professor	Coordinator & PSL	A,B,C,D,E	60%
École Polytechnique	Guilmant Quentin	PhD student		B.b and D	40%
INRIA Saclay	Fages François	DR INRIA		E.c	20%
INRIA Saclay	Soliman Sylvain	CR INRIA		E.c	20%
INRIA Saclay	Bostan Alin	CR INRIA		D, E.a	10%
INRIA Saclay	Chyzak Frédéric	CR INRIA		D, E.a	10%
Univ. Créteil	Madelaine Florent	Professor	PSL	C	60%
Univ. Créteil	Thapper Johan	MdC		C	50%
Univ. Clermont	Kanté Mamadou	MdC		C	25%
Univ. Créteil	Cervelle Julien	Professor		B.b	20%
Univ. Créteil	Valarcher Pierre	Professor		B.a	20%
Univ. Créteil	Barksukov Alexey	PhD student		C	18/24 months
Univ. Paris	Durand Arnaud	Professor	PSL	A,B,C,D,E	50%
Univ. Rennes	Kerjean Marie	Post. Doc		C and B.b	25%
Univ. Paris	Pouly Amaury	CR CNRS		E.a.b.c	25%
Univ. Paris	Asarin Eugène	Professor		E.b	20%
Univ. Paris	Laroussinie François	Professor		E.b	20%
Univ. Toulouse	Sablik Mathieu	Professor	PSL	E.a	30%
Univ. Orléans	Durand-Lose Jérôme	Professor		B.b	30%
Univ. Limoge	Barkatou Moulay	Professor		D	20%
Univ. Limoge	Cluzeau Thomas	Professor		D	20%

Changes made in the full proposal compared to the pre-proposal

We proposed in the pre-proposal to have 4 poles, with 5 administrative centers: The pole "Alan Turing Building" was including LIX subpole + INRIA subpole (physically in the same building, but at the time of the submission we were not sure to succeed to make an agreement that was satisfying constraints of CNRS/LIX and INRIA). We succeeded meanwhile to find some agreement in order to merge LIX and INRIA administrative centers: Everything concerning the pole "Alan Turing Building" will be administratively managed by the (CNRS-)LIX. So we keep exactly the same division into 4 scientific poles as in the pre-proposal, but with a simpler and more rational administrative organization, with each pole corresponding to an administrative center.

Every other statement is identical to the pre-proposal. In particular:

- The proposed budget is precisely matching what was written in the pre-proposal. The total budget of 367 273,44 euros is in particular indeed in the interval 325 K€ +/- 15%. The only slight modifications (falling in the proposed +/- 15% accuracy of preproprosal) is explained by the fact that we included the funding of internships (4*8 months in total, at the cost of 19 520 euros), that we apologize to have forgotten to explicitly include in the preproposal, and that we updated the approximate value to the exact value for the cost of the Postdoc and PhD, after exchanges with administrative people. We also increased the budget for travel costs from 70 000 euros to 75 500 euros, with respect to the relatively high number of participants (23 participants with the PhD and Postdoc).
- The composition of the teams, as well as the % of involvement already explicit in the pre-proposal, are unchanged. The involvement in the tasks is also unchanged.
- The decomposition in objectives and tasks is identical to the pre-proposal. We only provide more details.

I. Proposal's context, positioning and objective(s)

a. Objectives and research hypothesis

Since the early times of computer science, classifying problems by difficulty has been a thriving research area. Various models of computation have been developed, in order to classify either by complexity or by computability properties. Nowadays, classical computer science problems also deal with continuous data coming from different areas and modeling involves the use of tools like numerical analysis, probability theory or differential equations. Thus new characterizations related to these fields have been proposed. On a dual way, the quest for new types of computers recently led to revisit the power of some models for analog machines based on differential equations, and to compare them to modern digital models. In both contexts, when discussing the related computability or complexity issues, one has to overcome the fact that today's (digital) computers are in essence discrete machines while the objects under study are continuous and naturally correspond to Ordinary Differential Equations (ODEs).

The objective of this project is to consider a novel approach in between the two worlds: discrete-oriented computations on the one side and differential equations on the other side.

The project aims at providing new insights on classical complexity theory, computability and logic through this prism and at introducing new perspectives in algorithmic methods for differential equations solving and computer science applications.

ODEs appear to be a natural way of expressing properties and are intensively used, in particular in applied sciences. The theory of classical (continuous) ODEs has an abundant literature (see e.g. [Arn78, CL55]) and is rather well understood under many aspects. We are mainly interested here in a **discrete** counterpart of classical continuous ODEs: **discrete** ODEs. This analogue of the notion of classical continuous derivative, sometimes also called *finite differences*, has been widely studied in numerical optimization for function approximation [Gel71]. It is reminiscent of *discrete calculus* [GKPL89, Gle05, IAB09, Lau] in combinatorial analysis. Historically, similarities between discrete and continuous statements have also been observed as early as in the 19th century, under the terminology of *umbral* or *symbolic calculus*. However, even if the underlying computational content of finite differences theory is clear and has been pointed out many times, **no fundamental connections with algorithms and complexity have been exhibited so far.**

b. Position of the project as it relates to the state of the art

We recently started to develop a systematic study of discrete differential equations under a computational perspective. In a recent article [BD19], we demonstrated that discrete ODEs are a very natural tool for algorithm design and for proving that complexity and computability notions can be elegantly and simply captured using discrete ordinary differential equations.

We illustrated this by providing a characterization of **FPTIME**, the class of polynomial time computable functions, and of its non deterministic analog **FNP**. To this aim, we demonstrated how some notions from the analog world such as linearity of differential equations or derivatives along some particular functions (i.e. changes of variables) are representative of a certain computational hardness and can be used to solve or to design efficient algorithms for some (classical, digital) problems.

Formally

The theory of discrete derivatives: The basic idea is to consider the following concept of derivative function $\mathbf{f}'(x) = \mathbf{f}(x+1) - \mathbf{f}(x)$ for functions of type $\mathbf{f}: \mathbb{Z} \to \mathbb{Z}^d$, or more generally functions of type $\mathbf{f}: \mathbb{R} \to \mathbb{R}^d$, and even more generally functions $\mathbf{f}: \mathbb{R}^p \to \mathbb{Z}^d$ by considering partial derivatives. Several results from classical (continuous) derivatives generalise to this discrete settings: this includes linearity of derivation $(a \cdot f(x) + b \cdot g(x))' = a \cdot f'(x) + b \cdot g'(x)$, formulas for products and division such as $(f(x) \cdot g(x))' = f'(x) \cdot g(x+1) + f(x) \cdot g'(x) = f(x+1)g'(x) + f'(x)g(x)$. This includes the concept of integral: given some function $\mathbf{f}(x)$, $\int_a^b \mathbf{f}(x)\delta x$ is defined as a synonym for $\sum_{x=a}^{x=b-1} \mathbf{f}(x)$ with the convention that it takes value 0 when $a = b$ and $\int_a^b \mathbf{f}(x)\delta x = -\int_b^a \mathbf{f}(x)\delta x$ when $a > b$. The telescope formula yields the so-called Fundamental Theorem of Finite Calculus: let $\mathbf{F}(x)$ be some function. Then, $\int_a^b \mathbf{F}'(x)\delta x = \mathbf{F}(b) - \mathbf{F}(a)$.

Discrete ODEs: One can then naturally consider *discrete Ordinary Differential Equations (ODE)*, that is to say for example on equations of the (possibly vectorial) form: $\frac{\partial \mathbf{f}(x,\mathbf{y})}{\partial x} = \mathbf{h}(\mathbf{f}(x,\mathbf{y}), x, \mathbf{y})$. As expected, when some some initial value $\mathbf{f}(0,\mathbf{y}) = \mathbf{g}(\mathbf{y})$ is added, this is called an *Initial Value Problem (IVP)*. The theory of discrete ODEs is traditionally mainly used in computer algebra, or to discuss discretisation of continuous problems, or as a tool in combinatorics, and also, of course as a very natural tool to model various systems from our natural or physical world: see e.g. [Mur93, HSD03].

Discrete ODEs also seen as a way to model recursion. We showed that it turns out that discrete ODEs provide then a rather elegant way to express naturally some recursion/induction schematas, and revisit basic constructions from computation theory.

As an example, one can prove: **Theorem (Primitive Recursive functions & discrete ODEs).** The set of primitive recursive functions $\mathcal{PR}$ is the intersection with $\mathbb{N}^{\mathbb{N}}$ of the smallest set of functions that contains the zero function $\mathbf{0}$, the projection functions π_i^p, addition $+$, substraction $-$, and that is closed under composition and discrete ODE schemata.

Here, the *discrete ODE schemata* is the following: given $g: \mathbb{N}^p \to \mathbb{N}$ and $h: \mathbb{Z} \times \mathbb{N}^{p+1} \to \mathbb{Z}$, we say that f is defined by discrete ODE solving from g and h, denoted by $f = \mathrm{ODE}(g,h)$, if $f: \mathbb{N}^{p+1} \to \mathbb{Z}$ corresponds to the (necessarily unique) solution of Initial Value Problem $\frac{\partial f(x,\mathbf{y})}{\partial x} = h(f(x,\mathbf{y}), x, \mathbf{y})$, $f(0,\mathbf{y}) = g(\mathbf{y})$.

Intrinsic concepts from ODEs theory & Computations. The theory of (discrete) Ordinary Differential Equations comes with some natural concepts that have been derived from considerations far from computability and complexity theory. But it turns out that **this concepts have strong relations to complexity issues** and that such connections have not been yet fully explored, nor fully observed.

An example is the notion of **linear ODE**. It turns out that if the general discrete ODEs are replaced by *linear* ODEs, then one gets a characterisation of elementary functions. For example: **Theorem (Linear ODEs & Elementary functions).** The set of elementary functions $\mathcal{E}$ is the intersection with $\mathbb{N}^{\mathbb{N}}$ of the smallest set of functions that contains the zero function $\mathbf{0}$, the projection functions π_i^p, the successor function $\mathbf{s}$, addition $+$, subtraction $-$, and that is closed under composition and discrete linear ODE schemata (respectively: scalar discrete linear ODE schemata) LI.

Here, the *linear ODE schemata* is the following: given a vector $\mathbf{G} = (G_i)_{1 \leq i \leq k}$ matrix $\mathbf{A} = (A_{i,j})_{1 \leq i,j \leq k}$, $\mathbf{B} = (B_i)_{1 \leq i \leq k}$ whose coefficients corresponds to functions $g_i : \mathbb{N}^p \to \mathbb{N}^k$, and $a_{i,j} : \mathbb{N}^{p+1} \to \mathbb{Z}$ and $b_{i,j} : \mathbb{N}^{p+1} \to \mathbb{Z}$ respectively, we say that $\mathbf{f}$ is obtained by linear ODE solving from g, A and B, denoted by $\mathbf{f} = \mathrm{LI}(\mathbf{G}, \mathbf{A}, \mathbf{B})$, if $f : \mathbb{N}^{p+1} \to \mathbb{Z}^k$ corresponds to the (necessarily unique) solution of Initial Value Problem $\frac{\partial \mathbf{f}(x,\mathbf{y})}{\partial x} = \mathbf{A}(x,\mathbf{y}) \cdot \mathbf{f}(x,\mathbf{y}) + \mathbf{B}(x,\mathbf{y})$, $\mathbf{f}(0,\mathbf{y}) = \mathbf{G}(\mathbf{y})$.

Talking about complexity. Even more interestingly, it turns out that many constructions from **complexity theory** or **algorithm design** can be revisited using natural classes from discrete ODE theory. In particular, considering the concept of change of variable in computer algebra settings, it is natural to introduce the following variation on the notion of derivation: derivation along some function $\mathcal{L}(x,\mathbf{y})$. Let $\mathcal{L} : \mathbb{N}^{p+1} \to \mathbb{Z}$. We write $\frac{\partial \mathbf{f}(x,\mathbf{y})}{\partial \mathcal{L}} = \frac{\partial \mathbf{f}(x,\mathbf{y})}{\partial \mathcal{L}(x,\mathbf{y})} = \mathbf{h}(\mathbf{f}(x,\mathbf{y}), x, \mathbf{y})$, as a formal synonym for $\mathbf{f}(x+1,\mathbf{y}) = \mathbf{f}(x,\mathbf{y}) + (\mathcal{L}(x+1,\mathbf{y}) - \mathcal{L}(x,\mathbf{y})) \cdot \mathbf{h}(\mathbf{f}(x,\mathbf{y}), x, \mathbf{y})$.

This is motivated by the fact that the latter expression is similar to classical formula for classical continuous ODEs:

$$\frac{\delta f(x,\mathbf{y})}{\delta x} = \frac{\delta \mathcal{L}(x,\mathbf{y})}{\delta x} \cdot \frac{\delta f(x,\mathbf{y})}{\delta \mathcal{L}(x,\mathbf{y})} \ . \tag{1}$$

An important special case is when $\mathcal{L}(x,\mathbf{y})$ corresponds to the (binary) length $\mathcal{L}(x,\mathbf{y}) = \ell(x)$

This provides a way to simulate suitable changes of variables using this analogy, and even provide a way to characterise polynomial time.

As an example, one can prove: **Theorem (Polynomial Time & Linear Length ODEs).** The class of **FPTIME** of functions computable in polynomial time is precisely the smallest subset of functions, that contains the functions $\mathbf{0}$, $\mathbf{1}$, projections π_i^p, the length function $\ell(x)$, addition $+$, subtraction $-$, multiplication $\times$, the sign function $\mathsf{sg}(\cdot)$ and that is closed under composition (when defined) and linear length-ODE scheme.

There are strong evidences that this approach can be extended to characterize other complexity measures and exhibit new connections with classical notions of analysis.

This provides a way to reinterpret classical notions with these new eyes: for example, the Master Theorem (which explains for example the complexity of a recursive algorithm, see e.g. [CLRS09, Section 4.5]) at the basis of algorithm design can be basically read as a result on (the growth of) a particular class of discrete time length ODEs.

The purpose of the project is to develop this new perspective and to revisit logic, computability theory, complexity theory and even algorithm design from this original point of view.

Indeed, on the one hand, our results offer a new machine independent perspective on classical discrete computations, i.e. computations that deal with bits, words, or integers.

It has many consequences, and many expected impacts:

- A characterization of some computation theory class can very often be interpreted as some completeness result. For example, the above theorem about polynomial time can be read as. **Corollary.** Solving Linear length ODE is a **PTIME**-complete problem.

- It provides a way to say that constraints, in the spirit of **Constraint Satisfaction Problems (CSP)** that would be generated by linear discrete ODEs correspond precisely to polynomial time. Such constraints are then necessarily solvable in polynomial time. And conversely, an algorithm to solve a CSP that is polynomial can be seen as particular linear discrete Ordinary Differential Equations.

- It also points out the primary and very important role played by **differential equations in computation/complexity theory and logic**.

  In the context of proof theory and the so-called bounded arithmetic, following [Bus86] various characterizations of complexity classes have been obtained by restricting usual schematas of induction of Peano's axioms to restricted axioms such as polynomial or length induction: See e.g. [Bus98, KK+95, CN10]. Our settings offers the fascinating possibility of seeing such schematas, at least in spirit, as being closed to particular change of variables. This original point of view is clearly worth investigating.

  In the context of λ-calculus and linear logic, differentiation has been proved to be an adequate construction in λ-calculus and linear types [ER06, ER03]. Recently, research in this area showed how a primitive differential allowed to express and give simpler proofs for fundamentals results in λ-calculus [BM20]. In a parallel direction, Differential Linear Logic (DiLL) has been generalized to the handling of differential operators [Ker18]. The study of differential operators as primitive objects in proof-theory is thus a promising research direction—which could entertain links with the characterization of complexity classes via substructural logics [JYG92].

  This point of view is very original.

- **From a computer algebra point of view**, many results and constructions have been derived to get efficient methods for handling and solving problems related to discrete ODEs, **often in a framework that is common to continuous ODEs**. In particular, the class of *holonomic functions* and of their coefficient sequences has been studied intensively, leading to fast algorithms, e.g., for manipulating recurrences [BCG+17], for computing selected terms in a multi-indexed sequence [MR05], or for the summation of parametrized sums [PWZ96]. Having this broad range of efficient algorithms at hand has allowed to open computer algebra towards applications in enumerative combinatorics, where generating series often satisfy holonomic equations. This has been especially fruitful in the domain of the enumeration of walks, where the type of input equation ("kernel equation") is in fact a sort of discrete Partial Differential Equation [BM16]. This interest in combinatorics often has to do with classification questions, whose goal is to relate the membership in a well-defined class (rational, algebraic, holonomic, differentially algebraic, differentially transcendental), with a formalization of the notion of *difficulty to compute* [Pak18]. For example, a big work at this interface has led to a complete classification of 2D walks models with so-called small steps. While holonomic recurrences correspond to usual shifts by 1, another form of induction on natural numbers corresponds to their representation in a given digital base. This relates to divide-and-conquer recurrences, classical in computer science (cf. the Master Theorem for the complexity of a recursive algorithm), another form that has recently begun to receive attention from computer algebra [CDDM18]. Of course, this type of recurrences also relates to the algorithm paradigm by the same name.

One clear purpose of the current proposal is to contribute to understand these aspects.

And, on the other hand, such results also relate classical (discrete) complexity classes to analog computations, i.e. computations over the reals, as analog computation have been related in various ways to continuous ordinary differential equations, and as discrete ordinary differential equations provide clear hints about their continuous counterparts. Many very intriguing and original results have been recently obtained on revisiting computation theory of analog models of computation: see e.g. [BGP17, Pou15] characterizing (classical) polynomial time as solutions of polynomial differential equations of polynomial lengths. We believe the current approach is a crucial step towards better understanding why such results hold.

c. Methodology and risk management

Thus there are several mid-term goals of this line of research:

- ⋆ Bring insights from complexity theory to the problem of solving ODEs (discrete and, hopefully also, continuous) and use such a descriptive approach to classify large classes of ODEs by their computational hardness and bring some uniformity to the methods of this field.
- ⋆ Conversely, bring insights from ODE solving to complexity theory and computability by relating well-studied notions of analysis to complexity measures.

Concrete objectives (Tasks/Workpackages)

A **Implicit Complexity.** Provide a characterization of the main computational complexity classes using discrete ordinary differential equations. This task is essential to understand the intimate connections between the different types of discrete ODE and the various complexity measures. It will permit to identify what computational power may be associated to some specific restrictions (linearity or polynomiallity of equations, multiple variable derivatives, partial derivative, etc.).

(a) We have obtained a very first characterization of **PTIME** (polynomial time) in [BD19]. However, the approach seems flexible enough to capture a wide range of algorithmic classes by restricting time or space (e.g. **NP** (non-deterministic polynomial time), **PSPACE** (polynomial space)). Once obtained, another objective would to tackle the case of parallel classes.

(b) As a useful generalization, and a first step in investigating the continuous setting, we plan to characterize complexity classes over the reals (i.e. classes whose control flow is discrete but that manipulate real number) by system of mixed discrete/continuous ODE. This framework is the one of algebraic complexity, where complexity of problems is measured in terms of arithmetic operations, independently of (binary) representations of objects. We believe the discrete ODE approach is a really promising way to understand in an original way associated problems. In particular, as this framework is motivated both by complexity theory aspects (separation of complexity classes, lower bounds for various decision problems) as well as by computer algebra aspects, we believe our results may have an impact on both fields.

(c) Other well-studied objects are counting functions i.e. functions that count the number of combinatorial objects of a given size and that are the main subject of enumerative combinatorics. Recursive methods to denumerate are reminiscent to the approach of discrete ODE (see e.g. [Pak18]). This gives hope to establish a connection between natural classes of counting functions such as #**P** or **GapP** (the difference of two #**P** functions) and natural systems of ODE. In this vein, also extending the work on deterministic and non deterministic computations, it would be worth, although more risky, to investigate probabilistic complexity classes.

B **Extend the framework**

a **To cover more general structures and classes of algorithms.** Using the approach of evolving algebra and abstract state machines [Gur00] we will

i. **generalize the settings to continuous or more general dynamics**: our characterization is deeply based on encoding of configurations by integers. We propose to extend the framework to deal with more general structures (e.g. lists, stacks, arrays), using approaches based on logic. This can in particular extend [BDN16] where a semantic for ODEs was given.

ii. despite the fact that this new framework expresses usual classes of functions (for example **PR** or **PTIME**), **explore whether we can go the level of algorithms and not only functions**. Indeed, it has been demonstrated that, while models such as the Turing machines capture the notion of computable function, it is not capturing at a relevant level the notion of algorithms. Approaches to do so have been proposed such as the evolving algebra [GV10] or abstract state machines [Gur00]. Two topics will be of particular interest to us: by comparing the algorithmic expressive power of class restriction languages (**PR** or **PTIME**) with existing languages [MV16]. It's now well-known that control structures and data structures are of great importance for algorithmic expressivity. Some natural questions arise: are there algorithms that can be expressed in one model and not in the other? Can we talk about the algorithmic completeness of the calculation models? And integrating this computation model in an in-depth work on the faithful characterization (step-by-step simulation, or even better, literal simulation) of the different computation models via extensions of the Evolving Multi-Algebras framework [GV10].

b **To cover more general classes than ordinary differential equations,** such as partial differential equations.

From a computability point of view, while discrete ordinary differential equations cover sequential models of computation, one expects partial differential equations to cover models of parallelism, and in particular models of massive parallelism such as cellular automata (CA for short), one of the simplest massively parallel models. We intend to extend our existing ordinary differential equation results to cover such models. One already existing contribution in this direction was addressed in [Cer13] by one of the participant of this project, proposing a way to build a partial differential equation (PDE for short) system behaving like a CA. This requires to explore ways to transform state, space and time from a discrete to continuous mode.

In [Cer13], this was done using an intermediate model, interacting stochastic particle system, which was turned into a PDE of the form $\frac{\partial s}{\partial t} = F(s, \frac{\partial s}{\partial x}, \frac{\partial^2 s}{\partial x^2}, \cdots)$ where F is a polynomial. The restriction to polynomials was directed by the way a CA rule is simulated by a continuous system. In order to get a complex PDE system, we proposed to transform an expansive CA. As expansivity is not known to be decidable (it is a long stated open problem), we looked for permutive CA. Unfortunately, none of them of radius less than 6 led to a PDE and limitation of the systems used for the computation made the search with larger radius impossible. However, doing the process by hand for some well known expansive CA give interesting PDE systems whose simulation showed chaotic behavior. We propose to investigate these new directions:

i. First, we want to reimplement the program which searched for permutive CA which have a corresponding PDE. This initial computation was done using *SageMath* (for Gröbner basis computation) and GNU *Prolog* (for its constraint solver). The latter had memory issues. Writing the solver in a classical programming language will allow us to search for higher radius.

ii. We also want to base ourselves on asynchronous CA. Indeed, though PDE systems seem to correspond to parallel and synchronous CA, the space-time diagram we observe have some similarities to asynchronous CA. This can be explained by the intermediate model which is close to interacting particle systems and has a behaviour which is similar to asynchronous models: instead of drawing the evolving cell at random among all cells, each cell picks a random delay before evolving. We hope that the most chaotic asynchronous CA will lead to some interesting PDE systems.

iii. This last point is more experimental: we also want to change the process which turns a classical CA into a PDE. In the first work, we had an intermediate model which has continuous time and state. It allowed us to mathematically prove that the PDE system was the limit of the intermediate system when the distance between cells tends to 0. We want to try to build the PDE system directly from a CA definition, even using a less automated procedure. This would allow us to preserve more the original CA behavior, or allow more rules to be turned into a PDE, at the price of having to prove that the obtained system is chaotic by hand.

C **Relate the settings to classical approaches in logic or for constraint satisfaction problems (CSP)**

We have a characterization of **P** using discrete ODEs and a candidate for **NP** was outlined in [BDO18, BD19]. We plan to explore connections of our ODE framework for problems related to CSPs.

Ladner's theorem [Lad75] tells us that unless **P** equals **NP**, there are problems that are not in **P** yet not as hard as **NP**-complete problems such as SAT. Yet, there are natural and rich classes of problems that lie in **NP** and do not exhibit this intermediate behaviour. For example, the class of *Constraint Satisfaction Problems* (CSP) exhibit a dichotomy between **P** and **NP**-complete as proved independently by Bulatov and Zhuk [Bul17, Zhu17]. This motivates us to investigate an analogue of CSP for ODEs.

(a) We plan to investigate **a suitable ODE characterisation for the class of CSP**: The dichotomy for CSP was first conjectured by Feder and Vardi [FV98] when they attempted to delineate within **NP** the limit between classes that have a dichotomy and those for which a theorem *à la* Ladner holds. Typically these classes are defined via syntactic fragments of Existential Second Order logic (ESO), an approach from Descriptive Complexity. One such fragment, MMSNP, is shown more expressive than CSP, yet proved to exhibit a dichotomy since every sentence in this logic is polynomially equivalent to a finite domain CSP.

Another way to consider this MMSNP fragment, is to consider infinite domain CSPs *à la* Bodirsky, that is where the constraint language can be described by a relational structure enjoying nice model theoretic properties (typically a homogeneous structure or an ω-categorical structure). Such nice properties means that the problems remain amenable to algebraic methods that were instrumental in deriving the Bulatov-Zhuk dichotomy. There is a dichotomy conjecture for infinite domain CSPs *à la* Bodirsky, a conjecture which holds for MMSNP [BMM18].

There are a number of candidate super fragments of MMSNP or super classes of problems above CSP that might exhibit dichotomy: for example, MMSNP_2 [Mad09], which enjoys a natural definition when considering ontology mediated queries [BtCLW14], Hell's M-partition problems [Hel14], or *Amalgamation SNP* [BKS20].

(b) We intend to **Investigate super fragments of the ODE which still exhibits dichotomy**. In the framework of ODEs we can observe a feature already present in Descriptive Complexity: one oscillates between description of concrete problems and description of generic computations. For a concrete ODE example, one needs to find a suitable *change of variables* to outline the efficient solving procedure. Yet in the proof to relate a proposed ODE framework with a complexity class, one plays a game reminiscent of Fagin's lift of Cook's proof: we describe a generic computation with ODEs.

For a concrete CSP that is tractable, the description of the constraint language can be studied, in particular its algebraic properties, to witness that one specific polynomial-time algorithm applies.

Building upon and refining this task, one can hope that we can highlight how the (few) well known algorithms for tractable CSPs can be revisited as change of variables for ODEs.

(c) We intend to **Investigate how tractability of a finite domain CSP (algebraic properties) translates to change of variables for ODEs.**

As a test bed, we will concentrate on well understood subclasses of finite domain CSP (Boolean case of generalized SAT, homomorphism problem for undirected graphs or for tournaments).

(d) Such approaches are also related to implicit characterizations of complexity classes based on various restrictions of the considered logic: In *Differential Linear Logic*, the differential operator is interpreted by a *single* operator D fixed from the start. To *characterize logically* *the existence of solutions*, one should be able to construct the operator D **within** the calculus. This would be made possible by the convolution, interpreting the co-contraction rule $\bar{c}$. We will propose a resulting system that will **bring together differential equations and implicit characterizations of complexity classes via sub-structural logics**.

D **Relate to computer algebra.** The exact complexity of solving natural problems for discrete and continuous ODEs in the context of computer-aided mathematics is a well-developed field. Computer algebra often goes far beyond merely distinguishing polynomiality from non-polynomiallity, by providing tight/optimal complexity exponents in the polynomial case [BCG+17, BCO+07]. We intend to relate and extend such approaches to our problems from computation theory. The purpose is to go in the two ways: bring insights from computer algebra for (discrete) ODEs to computation theory, and conversely, from the discrete ODE approach to help to determine the complexity of problems.

In particular, we propose to contribute concretely to the following tasks:

(a) Much like dealing algorithmically with integers is more complex than with polynomials, the treatment of sequences in computer algebra (difference algebra) is trickier than that of functions (differential algebra). Even if algorithms for parametrized sums and integrals often share similar design patterns, **a recent optimization of "creative telescoping" for integrals**, by the so-called reduction-based approach [vdH18, BCLS18], still waits for its recurrence counterpart, which we would like to complete both in theory and practice in the framework of this project. In passing, we expect that in turn, improvements of the recurrence case will provide further ideas for the continuous case.

(b) Counting lattice walks confined to cones is a hot topic, since such walks encode many combinatorial objects and probabilistic processes (systems of queues, Young tableaux, permutations, ...). They are studied both in a discrete and a continuous setting. From a technical point of view, counting walks is part of a general program aiming at solving **partial discrete differential equations**, either qualitatively or quantitatively (or both). An important step in this direction was the resolution of the difficult *Gessel model* [BM16], based among other things on the treatment of a special type of discrete PDEs, called *with one catalytic variable*. For such equations, the solution is known to be *algebraic*. We plan to **investigate efficient algorithms that automatically solve such equations**.

(c) In a pioneering paper [Wil82], Wilf proposed to judge the mathematical notion of *formula* from the complexity point of view. Recently, Pak [Pak18] studied more thoroughly the concept of Wilfian formulas and proposed several challenges in enumerative combinatorics from the computational complexity viewpoint. In this context, we aim at **studying his Open Problem 2.4**, on large classes of combinatorial sequences: an ordinary generating series and the associated exponential generating series cannot both satisfy non-linear diffential equations unless they both satisfy linear differential equations.

(d) The literature on divide-and-conquer (DAC) recurrence systems is so far most often limited to the first order, with results that focus on deriving the asymptotic behavior of the solutions. Higher-order DAC systems express intricate models that are defined by cases. This goes beyond merely distinguishing between even and odd, and corresponds more generally to fixing the last digits when numbers are written. Even the question of the consistency of a given system of DAC equations has not been answered yet. To broaden the class of models that can be dealt with algorithmically, we will **develop algorithmic tests to determine the consistency of a DAC system**. In passing, we will be able to determine the degree of freedom of solutions of a given system, and to describe what "initial conditions" fully determine a solution.

E **Applications:**

a **To the analysis of dynamical systems:** Generally the properties of true interest in dynamical systems are about the long-term evolution, they are captured by some invariants such as entropy, attractors, asymptotic measures... An important line of research over the last decade is to determine the classes of algorithmic complexity of these invariants and search if this allows to characterize all accessible values. This is studied for specific discrete dynamical systems as subshift of finite type [HM10, Sim10, Mey11, Hoc09, AS13] or cellular automata [GZ12, HdMS16] or in a general framework for the study of the algorithmic complexity of the entropy [GHRS20]. We plan to use the framework to **derive computability and complexity bounds on the hardness of natural problems for dynamical systems** and contribute to establish the dictionary (frontier) between dynamical properties and class of complexity. Specifically, we plan to explore the following problems:

   i. Attractors describe the asymptotic behavior of a dynamical system, they correspond to the set observed during a long-term simulation. A natural question is to understand the **algorithmic complexity of an attractor**. Even for a very simple system, this complexity can be prohibitively high. For instance, in [RY19], the authors showed that a quadratic map of the interval (the simplest example of a non-linear dynamical system) can have an attractor with an arbitrarily high computational complexity – which would render any computer simulation of long-term dynamics impractical. However the Lorenz attractor introduced in 1963 by E. N. Lorenz is proved to be computable [RGZ19] but we don't know which property on differential equations make the attractor computable. We plan to study the general conditions which guarantee computability of attractors, and, conversely, investigate how prevalent the phenomenon of non-computability is for attractors of a typical dynamical system.

   ii. To study intermediate behavior we can also consider the **prediction problem**: given some initial condition and some duration t, what is the state of the system after time t? Two aspects can make this problem hard: the topological dynamics of the system (from the most stable, like equicontinuous, to the most chaotic, like expansive) and its computational complexity (varying from constant time to P-complete). We want to explore the interplay between these two aspects. For example, the prediction problem of CA with strong algebraic properties is LOGSPACE [Moo97].

   iii. The notion of parametrized complexity, which has emerged in computable analysis, emerges also in the study of dynamical systems. The computability of properties, functions or reals is related to some external parameter, possibly uncomputable, but which measures in a natural way how far we are from the frontier with intractability. For example, while analytic functions cannot be simulated in general, they can be simulated in polynomial time with respect to the distance to its singularities [KTZ18], or in their length [BGP17]. We plan to continue in this direction of **quantifying the hardness of problems according to continuity or smoothness of relevant functions** depending on natural parameters such as their modulus of continuity or smoothness.

   iv. Finally, one can decide to put constraints on the algorithms used to compute dynamical invariants. A very natural instance of this question deals with spatial discretisations of continuous systems: if a quantity is computable for a certain class of systems, can it also be computed from the naive algorithm of spatial discretisation? In general this algorithm is extremely ineffective when coming to actually compute invariants, but are there systems for which the discretisation algorithm is in some sense the fastest one?

b **To verification of systems:** The purpose of automated verification (model-checking) is given the description of some system, and of some property of the system, to provide as automated as possible procedure to check that the system is satisfying the property. In the context of hybrid and cyber-physical system, models are often described as particular dynamical systems, using suitable formalisms: A hybrid system is a system mixing continuous evolutions, typically described using ordinary differential equations, with discrete transitions, typically modeling the evolution of some devised digital controller. A cyber-physical systems aims at modeling a digital system acting over some continuum environment, also often described using ordinary differential equations. All of this can still be seen as particular classes of dynamical systems. Compared to previous item, the main difference lie in the typical properties of interest: In the context of verification, a key property is reachability: given some description of the initial state, and some description of a subset of states, one needs to determine if this subset of states can be reached by the system.

Determining the computational hardness of this problem is very closely related to discussions of the computational power of various continuous time and space models of computation. Actually, many of the models for continuous time computations, as well as for discrete time computations have been proposed initially in the framework of Verification. Many works have been then devoted to identify decidable, or computably tractable subclasses: See Survey [BP18] for full discussions on these aspects.

We believe the discrete ODE approach of this project may be of a clear help to discuss the hardness of various classes of hybrid and cyber-physical systems, because of the very close distance between existing results in computation theory for computations over the continuum and results in verification (or control) community.

Concretely, we propose in the current project to investigate the following aspects:

- **Study the impact of finite precision on the hardness of reachability issues** in hybrid automata with a continuous space and a discrete time (which correspond typically to the class of systems most naturally modeled by discrete ordinary differential equations), or for other models of cyber-physical systems. One clear aspect of models for cyber-physical systems compared to hybrid automata based models is on the fact that they may be distributed systems.
- Relate the approach to **results known for models for recurrent neural networks (RNNs)**. There have been historically many results established in the 90's concerning the computational power of such models. We plan to revisit these results with respect to the approach based on discrete ODEs of the current proposal. In particular, this requires to also compare these results to the more modern way deep-learning models are used to compute in today's models of deep-learning.

c **To intrinsically continuous models of computation**

We will relate the approach to **(continuous time) analog models of computations**, in particular to continuous time models of computations. Notice that we came to the discrete ODE settings by an attempt to understand continuous models of computation, in particular continuous time analog models of computation. Refer to [BP18] for a survey.

It must be understood that, while models based on ODEs may look at first sight similar, this seems not an easy task, as the continuous time and discrete time models are different models, with different notions of complexity, and with different ways on which this complexity is measured. But, we intend to investigate the connection, in particular in relations with **models of computations considered in bioinformatics**. In particular, the spirit of [FLGBP17] for which the authors involved in this proposal received the *2019 Prix du journal "La Recherche"* for a proof or Turing universality of proteins computations using ODE methods.

Indeed, **models of bioinformatics** naturally lead to continuous time analog models, described with **(classical continuous) ordinary differential equations**.

Namely, in computational systems biology, the study of high-level functions in the cell, and of their implementation with continuous chemical reaction networks, challenge our theories of analog and hybrid discrete-continuous computation models. For example, the particular structure of cell signalling networks can now be understood as biochemical implementations of input-output real functions, e.g. by sigmoid functions for the implementation of an analog-digital converters in the cell, producing a digital response to analog signals. Similarly, biochemical oscillators can be analysed as biochemical implementations of functions of time, and the progression in cell processes, such as the cell division cycle, as an implementation of sequentially with analog switches [TAG+08]. The proof of Turing completeness of finite continuous CRNs [FLGBP17] provides a formal framework from which a compiler in CRN of computable real functions, presented as solutions to a polynomial IVP, has been implemented in the Biochemical Abstract Machine (Biocham) software and used for first applications in synthetic biology [CAF+18].

This raises however **a number of fundamental questions**, together with some original intuitions, **on the definition of relevant low analog computation complexity classes**, on the complexity of binomialisation (reduction without loss of generality to polynomial degrees at most 2 for reactions with at most two reactants), on the expressive power of rate independent CRN (when the result is independent of rate constants), on complete stabilization [BFS18], on CRN model reductions [SfR14], and on the extension of error control to a robustness control *a priori* (by moment closure methods for stochastic CRNs).

These deep questions **will be investigated in the project with the support of the discrete ODE framework for the purpose of contributing to the links to complexity theory and algorithm design for analog computation**.

Concretely, during the project, we propose:

- To formalize various notions of computability by discrete ODEs (the framework at the basis of this proposal) and continuous time analog computations in continuous chemical reaction networks. In particular, we will investigate notions of computations based on stabilisation of components, and compare these notions to the ones considered in [Pou15, BGP17]. The objective will be to compare these notions of computation, and provide equivalence results.
- Rate independent Chemical Reaction Networks form a restricted class of CRNs of high practical value since they enjoy a form of absolute robustness in the sense that the result is completely independent of the reaction rates and depends solely on the input concentrations. The functions computed by rate-independent CRNs have already been characterized mathematically by Soloveichik *et al.* as the set of piecewise linear functions from input species. We will consider rate independent computations and study their relevance to define low complexity classes of analog computation, in the spirit of preliminary results in [DFS20].
- Use the framework of discrete ODEs to look for a characterization of non-deterministic polynomial time for continuous time analog models of computation. Possibly, go to a characterization of polynomial space.

II. Organisation and implementation of the project

a. a. Scientific coordinator and its consortium / its team

Scientific Coordinator **Olivier Bournez** (Full professor at École Polytechnique, 60%). O. Bournez is recognized as an expert on computability and complexity issues with ordinary differential equations, and on analog models of computations. He had several experiences of management, including being the head of the LIX laboratory during 5.7 years. He is currently the leader of the pole "Proofs and Algorithms" of LIX, and of the research team “Algorithms and Complexity” of LIX. He is author of several reference surveys on various models of computations based on ordinary differential equations, and in particular of the most recent chapter [BP18].

Participants The consortium includes 4 scientific poles. For each participant is indicated the % of implication and the main tasks of contribution below. We indicate here the main expected contribution tasks for each participants (without being exclusive).

The consortium is built using the idea of including experts of all the main related fields, in order to reach the proposed objectives. In particular, the tasks, are concerning **various, distinct and complementary fields**: complexity theory (Task A, E), logic in computer science (Tasks B and C), computability (Task A, E), constraints and applications (Task C), verification (E.b) and bioinformatics (Task E.c) and dynamical systems (Task E.a) and computer algebra (Task D). The distribution of participants over the tasks reflects their expertise, and can be read below.

Notice that related expertises, and fields are actually often crossed among the poles.

- **The "Turing Building" pole** includes members from the LIX with O. Bournez (60%, Tasks A, B, C, D, E) and Q. Guilmant (PhD Student, 40%, Task B.b and D). It also include members of INRIA Saclay with François Fages (20%, Task E.c) and Sylvain Soliman (20%, Task E.c), Alin Bostan (10%, Task D, E.a), Frédéric Chyzak (10%, Task D, E.a).
- **The "Créteil" pole** includes Florent Madelaine (local coordinator, 60%, Task C), Johan Thapper (50%, Task C), Mamadou Kanté (25%, Task C), Julien Cervelle (20%, Task B.b), Pierre Valarcher (20%, Task B.a), and Alexey Barsukov (PhD Student 18 months/24, Task C).
- **The "Paris Diderot" pole** includes Arnaud Durand (local coordinator, 50%, Tasks A, B, C, D,E), Marie Kerjean (25%, Task C and B,b), Amaury Pouly (25%, Task E.a,b,c), Eugène Asarin (20%, Task E.b) and François Laroussinie (20%, Task E.b).
- **The "Nationale 20" pole** includes Mathieu Sablik (local coordinator, Toulouse, 30%, Task E.a), Jerôme Durand-Lose (Orléans, 30%, Task B.b), Moulay Barkatou (Limoges, 20%, Task D), Thomas Cluzeau (Limoges, 20%, Task D).

We insist on the fact that **this project is clearly proposing to create a new working group on these topics, and is original, and clearly transverse to any already existing funding, and even existing working groups**: It has been conceived in some transverse way, with the clear motivation of progressing on this original new scientific questions.

Originality and Related awards and visible facts. The approach of devising computability and complexity, and more generally computations, with ordinary differential equations has been awarded recently several prizes and awards:

1. Marie Kerjean has been just selected as one of the laureates of the **"L'Oréal-Unesco Pour les Femmes et la Science" 2019** grants for her work relating logic (type theory) and differential equations.
2. Le **Prix La Recherche 2019** for Computer Science have been attributed to an article coauthored by F. Fages, G. le Guludec, O. Bournez and A. Pouly, **best paper award CMSB 2017**.
3. Amaury Pouly co-supervised by O. Bournez and D. Graça (Portugal) has been awarded **the Ackermann Award 2017** for his characterization of polynomial time with (continuous) Ordinary Differential Equations.
4. The characterization of analog algorithms (with ODEs) from O. Bournez, N. Dershowitz and P. Néron has received the **Best Paper award** at Computability in Europe in 2016.
5. The **Track B best paper award of ICALP 2016** has been attributed to O. Bournez, D. Graça and A. Pouly for a characterization of polynomial time with ODEs of polynomial length.

Implication of the scientific coordinator and partner's scientific leader in on-going project(s)

Name of the researcher	Person. month	Call, funding agency, grant allocated	Project's title	Name of the scientific coordinator	Start–end
Olivier Bournez	0	None	$\emptyset$	$\emptyset$	
Florent Madelaine	0	None	$\emptyset$	$\emptyset$	
Arnaud Durand	10	ANR-19-CE48-0019-03	EQUUS Efficient Query answering Under UpdateS	Stefan Mengel	2020-2023
Mathieu Sablik	0	None	$\emptyset$	$\emptyset$	
François Fages	4	ANR-DFG	SYMBIONT Symbolic methods for biochemical networks	T. Sturm, A. Weber	2018-2021
François Fages	12	ANR-MOST	BIOPSY Biochemical programming system	F. Molina, J.H. Jiang	2016-2020

b. Implemented and requested resources to reach the objectives

Work program and requested resources. The project is planned for a duration of **48 months**. In addition to usual cooperations between members, we plan to organize 2 meetings of the consortium per year (with more than half outside of Paris), with a scientific part fully open to non-members of the project. With the resources of the project, we plan to invite experts on implicit complexity and more generally complexity theory, computer algebra, numerical methods. We will also organize a fully open workshop for diffusion of the project outcomes.

Budget.

Requested means by item of expenditure and by partner*
In Euros.

		Partner *Turing Building*	Partner *Créteil*	Partner *Paris Diderot*	Partner *Nationale 20*
Staff expenses		85 448	123 860	4 880	4 880
Instruments and material costs (including the scientific consumables)		4 000	6 000	3 000	2 500
Building and ground costs		0	0	0	0
Outsourcing / subcontracting		0	0	0	0
General and administrative costs & other operating expenses	Travel costs	31 000	37 000	21 000	16 500
	Admin. management & structure costs**	9 635,84	13 348,80	2 310,40	1 910,40
Sub-total		130 083,84	180 208,80	31 190,40	25 790,40
Requested funding		367 273,44			

The proposed budget is **367 273,44 k€**. Staff expenses correspond to 18 months of postdoc (80 568 euros, administratively at Turing Building Pole), 1 PhD fellowship (36 months, 118 980 euros, administratively at Créteil Pole) as well as 8*4 months of M2 internships (total 19 520 euros).

Instruments and material costs (total: 15 500 euros) correspond to laptops and computers, mostly devoted to students and postdocs participating to the project.

The travel cost include 75 500 euros (for the 23 participants, which corresponds to a mean funding of 820 euros/year/participant): This amount is intended to cover participations to the main international events of the various related fields (LICS, ISSAC, ICALP, . . .), with the will to encourage young people to travel. These costs also include the funding of 3 days/year of annual meeting for the project (4 500 euros). In particular, a *kickoff* meeting will be organised as soon as possible in the first year. At least half of the meetings will happen outside of Paris. Meetings will include presentations from members of the consortium, but also invited talks from experts on subjects of interest for the project. This budget also includes 3*1 500 euros = 4 500 euros corresponding to 3 visits of 2 weeks per year per student, 2 000 euros for the postdoc.

Travel cost also include the funding of a workshop (30 000 euros): This will cover invitations at the international level, assuming that some invited people will stay about 1 week to work on the project. This also include 18 000 euros for travel costs of participants, and grants to fund the participation of students.

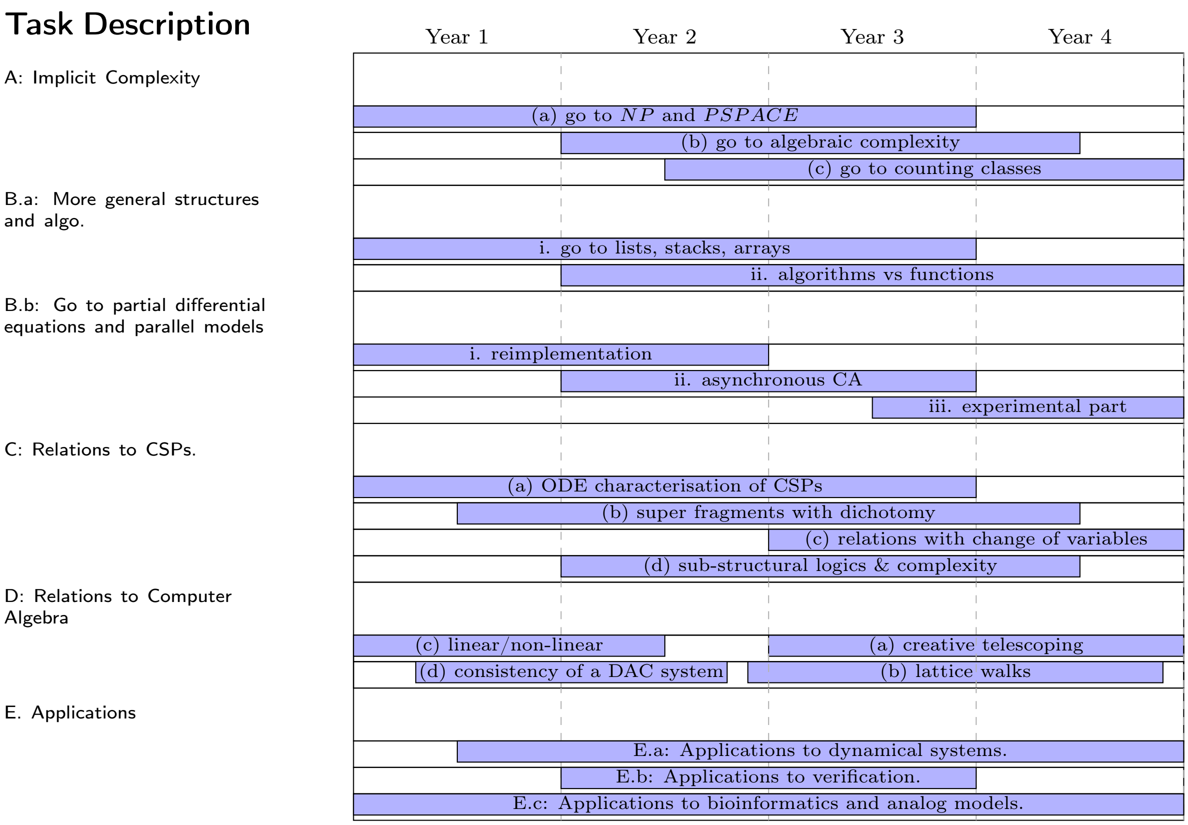
The administrative management and structure costs corresponds to 8% of costs.

Both postdocs and PhD will visit the various poles, and in particular the Nationale 20 pole which is physically less close to Paris than others. The instruments and material costs have been attributed mostly with respect to the man/month involvements of the poles (putting aside some rounding effects). The part of the costs related to travel costs of participants to the project have also been attributed attributed mostly with respect to the man/month involvements of the poles, but with more funding for the pole *Nationale 20* less close to Paris.

The recruited PhD will mainly contribute to tasks B and C. The recruited Postdoc will mainly contribute to tasks E. We believe these latter task require more expertise in computation theory, or in related application fields, and this is why a postdoc is preferred on these aspects (exact task will also depend on the selection of candidates).

We also plan to recruit a PhD student on task A, funded by some external resource (Funding from École Normale Supérieure of Paris). This PhD student will work in *Paris Diderot* pole. We are finalizing its recruitment, and the name of the student is already known, and the funding confirmed.

The **Gantt chart** of the project is the following:



Concerning **dependencies between tasks**: Tasks $B.a$, $C(a)$, $E.a$, $E.b$ and $E.c$ depends on the results of the task $A(a)$: But actually, because characterizations for new complexity classes such as NP or $PSPACE$ are expected. Notice that we have some preliminary characterizations of such classes. However, in order to mitigate risks, it is possible to start the work for all these classes using the already existing characterization for polynomial time. Task $E.a$, $B.b$ and $D(b)$ are related, as part of them is dealing with partial ODEs, and spatial models, and will be developed in parallel and closed cooperation. Tasks related to A., and B.a and B.b are also closely interlated, and will be developed in close connections. The first part of the year 1 will be devoted to unification of techniques and approaches for the various tasks, and contexts involved in the consortium.

III. Impact and benefits of the project

Impact: We believe the approach of the current project is very original, transverse to all existing working groups, and could have a clear impact in various other fields: computer algebra (new methods, characterization of the computational hardness of solving some particular ODEs), verification (hardness of reachability analysis), dynamical systems (frontier between tractability/non-tractability for natural properties), bio-informatics (computation with bio-chemical reactions), as well as very fundamentally to the foundations of computer science (logic, computation theory, mathematics).

Concerning diffusion of knowledge to general audiences: Several participants have visible experiences in communication to general (non-academic, non-computer science or mathematics) audience. Our constant efforts towards diffusion will clearly be pursued during the duration of the project. Several participants, including the coordinator, is regularly using social networks to communicate about science (*Twitter*, *Facebook*, *Instagram*). In particular, we can mention that he has many followers with Twitter coming from French *classes préparatoires*, *lycée* (and sometimes *collège*) teachers, in Maths and other scientific fields. Partly also because his teaching are made available to these audience online, and communicated via these medias.

In particular, notice that the article [BD19] at the source of the current proposal have been commented by this audience, under twitter, at the time of its publication, as being explicitly some accessible science. This were not restricted to mathematical audience, but also commented by researchers and *lycée* professors from physics.

As another illustration, observe that the bioinformatics applications of our results have received the *2019 Prix du journal "La Recherche"*, with an article of several pages in the magazine about our approach.

Several participants have also already contributed to blogs such as *Binaire* of the *Le Monde*, and have been invited to do so on topics, closely related to the scientific subject of this proposal.

Notice also that the *UNESCO-L'Oréal grant* to Marie Kerjean is also clearly concretely a whole set of very visible actions in terms of diffusion of knowledge, and to encourage young women to go to science, and through her work, also to a related science, emphasizing the role of ordinary differential equations as a fundamental object for computations.

We cannot guarantee at this moment what will be precisely the actions for the communication of the related science, but clearly we will continue our efforts toward these directions, as several participants are clearly convinced of the interest of diffusing science, not only to academia.

Other contributions Notice also that several of the participants are contributing to software that is freely distributed. For example, F. Fages is the main designer and S. Soliman the main developper of the *Biochemical Abstract Machine* (*Biocham*) software for modeling, analyzing and synthesizing CRNs. *Biocham*, currently version 4.4.5, is developed since 2003 and distributed under the GNU General Public Licence 2. This software will be used in relation with aspects of E.c, in order to demonstrate the approach.

IV. References related to the project

[Arn78] V. I. Arnold. *Ordinary Differential Equations.* MIT Press, 1978.

[AS13] Nathalie Aubrun and Mathieu Sablik. Simulation of Effective Subshifts by Two-dimensional Subshifts of Finite Type. *Acta Appl. Math.*, 126:35–63, 2013.

[BCG+17] Alin Bostan, Frédéric Chyzak, Marc Giusti, Romain Lebreton, Grégoire Lecerf, Bruno Salvy, and Éric Schost. *Algorithmes Efficaces en Calcul Formel.* Frédéric Chyzak (self-pub.), Palaiseau, September 2017. 686 pages. Printed by CreateSpace. Also available in electronic format.

[BCLS18] Alin Bostan, Frédéric Chyzak, Pierre Lairez, and Bruno Salvy. Generalized Hermite reduction, creative telescoping and definite integration of D-finite functions. In *ISSAC'18—Proceedings of the*

2018 ACM International Symposium on Symbolic and Algebraic Computation, pages 95–102. ACM, New York, 2018.

[BCO+07] Alin Bostan, Frédéric Chyzak, François Ollivier, Bruno Salvy, Éric Schost, and Alexandre Sedoglavic. Fast computation of power series solutions of systems of differential equations. In *18th ACM-SIAM Symposium on Discrete Algorithms*, pages 1012–1021, 2007. New Orleans, January 2007.

[BD19] Olivier Bournez and Arnaud Durand. Recursion schemes, discrete differential equations and characterization of polynomial time computation. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th Int Symposium on Mathematical Foundations of Computer Science, MFCS*, volume 138 of *LIPIcs*, pages 23:1–23:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[BDN16] Olivier Bournez, Nachum Dershowitz, and Pierre Néron. Axiomatizing analog algorithms. In Arnold Beckmann, Laurent Bienvenu, and Natasa Jonoska, editors, *CIE 2016*, volume 9709 of *Lecture Notes in Computer Science*, pages 215–224. Springer, 2016.

[BDO18] O. Bournez, A. Durand, and S. Ouazzani. Recursion schemes, discrete differential equations and characterization of polynomial time computation. Technical report, October 2018.

[BFS18] Adrien Baudier, François, and Sylvain Soliman. Graphical requirements for multistationarity in reaction networks and their verification in biomodels. *Journal of Theoretical Biology*, 459:79–89, December 2018.

[BGP17] 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.

[BKS20] Manuel Bodirsky, Simon Knäuer, and Florian Starke. ASNP: a tame fragment of existential second-order logic. *CoRR*, abs/2001.08190, 2020.

[BM16] Mireille Bousquet-Mélou. An elementary solution of Gessel's walks in the quadrant. *Adv. Math.*, 303:1171–1189, 2016.

[BM20] Davide Barbarossa and Giulio Manzonetto. Taylor subsumes Scott, Berry, Kahn and Plotkin. *PACMPL*, 4(POPL):1:1–1:23, 2020.

[BMM18] Manuel Bodirsky, Florent R. Madelaine, and Antoine Mottet. A universal-algebraic proof of the complexity dichotomy for monotone monadic SNP. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS*, pages 105–114. ACM, 2018.

[BP18] Olivier Bournez and Amaury Pouly. *Handbook of Computability and Complexity in Analysis*, chapter A Survey on Analog Models of Computation. Springer. To appear (arXiv version https://arxiv.org/abs/1805.05729), 2018.

[BtCLW14] Meghyn Bienvenu, Balder ten Cate, Carsten Lutz, and Frank Wolter. Ontology-based data access: A study through disjunctive datalog, csp, and MMSNP. *ACM Trans. Database Syst.*, 39(4):33:1–33:44, 2014.

[Bul17] Andrei A. Bulatov. A dichotomy theorem for nonuniform csps. In Umans [Uma17], pages 319–330.

[Bus86] S. Buss. *Bounded arithmetic.* Bibliopolis, 1986.

[Bus98] Samuel R Buss. First-order proof theory of arithmetic. *Handbook of proof theory*, 137:79–147, 1998.

[CAF+18] Alexis Courbet, Patrick Amar, François Fages, Eric Renard, and Franck Molina. Computer-aided biochemical programming of synthetic microreactors as diagnostic devices. *Molecular Systems Biology*, 14(4), 2018.

[CDDM18] Frédéric Chyzak, Thomas Dreyfus, Philippe Dumas, and Marc Mezzarobba. Computing solutions of linear Mahler equations. *Math. Comp.*, 87(314):2977–3021, 2018.

[Cer13] Julien Cervelle. Constructing continuous systems from discrete cellular automata. In Paola Bonizzoni, Vasco Brattka, and Benedikt Löwe, editors, *Conference on Computability in Europe*, volume 7921 of *Lecture Notes in Computer Science*, pages 55–64. Springer, 2013.

[CL55] E. A. Coddington and N. Levinson. *Theory of Ordinary Differential Equations.* Mc-Graw-Hill, 1955.

[CLRS09] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms (third edition).* MIT press, 2009.

[CN10] Stephen Cook and Phuong Nguyen. *Logical foundations of proof complexity*, volume 11. Cambridge University Press Cambridge, 2010.

[DFS20] Élisabeth Degrand, François Fages, , and Sylvain Soliman. Graphical conditions for rate independence in chemical reaction networks. Technical report, INRIA, 2020.

[ER03] Thomas Ehrhard and Laurent Regnier. The differential lambda-calculus. *Theoretical Computer Science*, 309(1-3):1–41, 2003.

[ER06] Thomas Ehrhard and Laurent Regnier. Differential interaction nets. *Theoretical Computer Science*, 364(2):166–195, 2006.

[FLGBP17] 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.

[FV98] Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1998.

[Gel71] Aleksandr Gelfond. *Calculus of finite differences*. Hindustan Publ. Corp., 1971.

[GHRS20] Silvère Gangloff, Alonso Herrera, Cristobal Rojas, and Mathieu Sablik. On the computability properties of topological entropy: a general approach. *A paraître dans Discrete and Continuous dynamical systems*, 2020.

[GKPL89] Ronald L Graham, Donald E Knuth, Oren Patashnik, and Stanley Liu. Concrete mathematics: a foundation for computer science. *Computers in Physics*, 3(5):106–107, 1989.

[Gle05] David Gleich. Finite calculus: A tutorial for solving nasty sums. *Stanford University*, 2005.

[Gur00] Yuri Gurevich. Sequential abstract-state machines capture sequential algorithms. *ACMTCL: ACM Transactions on Computational Logic*, 1(1):77–111, 2000.

[GV10] Serge Grigorieff and Pierre Valarcher. Evolving multialgebras unify all usual sequential computation models. In Jean-Yves Marion and Thomas Schwentick, editors, *27th International Symposium on Theoretical Aspects of Computer Science, STACS 2010, March 4-6, 2010, Nancy, France*, volume 5 of *LIPIcs*, pages 417–428. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010.

[GZ12] Pierre Guillon and Charalampos Zinoviadis. Densities and entropies in cellular automata. In S. Barry Cooper, Anuj Dawar, and Benedikt Löwe, editors, *How the World Computes - Turing Centenary Conference and 8th Conference on Computability in Europe, CiE 2012, Cambridge, UK, June 18-23, 2012. Proceedings*, volume 7318 of *Lecture Notes in Computer Science*, pages 253–263. Springer, 2012.

[HdMS16] Benjamin Hellouin de Menibus and Mathieu Sablik. Characterisation of sets of limit measures after iteration of a cellular automaton on an initial measure. *Ergodic Theory and Dynamical System*, 2016.

[Hel14] Pavol Hell. Graph partitions with prescribed patterns. *Eur. J. Comb.*, 35:335–353, 2014.

[HM10] Michael Hochman and Tom Meyerovitch. A characterization of the entropies of multidimensional shifts of finite type. *Annals of Mathematics*, 171(3):2011–2038, 2010.

[Hoc09] Michael Hochman. On the dynamics and recursive properties of multidimensional symbolic systems. *Invent. Math.*, 176(1):131–167, 2009.

[HSD03] Morris W. Hirsch, Stephen Smale, and Robert Devaney. *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Elsevier Academic Press, 2003.

[IAB09] FA Izadi, N Aliev, and G Bagirov. *Discrete Calculus by Analogy*. Bentham Science Publishers, 2009.

[JYG92] P. Scott J.-Y. Girard, A. Scedrov. A modular approach to polynomial-time computability. *Theoretical Computer Science*, 97(1):1–66, 1992.

[Ker18] Marie Kerjean. A logical account for linear partial differential equations. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS*, 2018.

[KK+95] Jan Krajicek, Jan Krajíček, et al. *Bounded arithmetic, propositional logic and complexity theory*, volume 60. Cambridge University Press, 1995.

[KTZ18] Akitoshi Kawamura, Holger Thies, and Martin Ziegler. Average-Case Polynomial-Time Computability of Hamiltonian Dynamics. In Igor Potapov, Paul Spirakis, and James Worrell, editors, *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*, volume 117 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 30:1–30:17, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Lad75] Richard E. Ladner. On the structure of polynomial time reducibility. *J. ACM*, 22(1):155–171, 1975.

[Lau] Gustavo Lau. Discrete calculus.

[Mad09] Florent R. Madelaine. Universal structures and the logic of forbidden patterns. *Logical Methods in Computer Science*, 5(2), 2009.

[Mey11] Tom Meyerovitch. Growth-type invariants for $\mathbb{Z}^d$ subshifts of finite type and arithmetical classes of real numbers. *Inventiones Mathematicae*, 184:567–589, 2011.

[Moo97] C. Moore. Quasi-linear cellular automata. *Physica D*, 103, 1997.

[MR05] P. Massazza and R. Radicioni. On computing the coefficients of bivariate holonomic formal series. *Theoret. Comput. Sci.*, 346(2-3):418–438, 2005.

[Mur93] J. D. (James Dickson) Murray. *Mathematical Biology*, volume 19 of *Biomathematics*. Springer Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., second edition, 1993.

[MV16] Yoann Marquer and Pierre Valarcher. An imperative language characterizing ptime algorithms. In Patrick Cégielski, editor, *CSLI Publications, Studies in Weak Artihmetics*, volume 3. Stanford Press, 2016.

[Pak18] Igor Pak. Complexity problems in enumerative combinatorics. In *Proceedings of the International Congress of Mathematicians—Rio de Janeiro 2018. Vol. IV. Invited lectures*, pages 3153–3180. World Sci. Publ., Hackensack, NJ, 2018.

[Pou15] 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.archives-ouvertes.fr/tel-01223284, Prix de Thèse de l'Ecole Polyechnique 2016, Ackermann Award 2017.

[PWZ96] Marko Petkovšek, Herbert S. Wilf, and Doron Zeilberger. $A = B$. A K Peters, Ltd., Wellesley, MA, 1996. With a foreword by Donald E. Knuth, With a separately available computer disk.

[RGZ19] Cristobal Rojas, Daniel Graca, and Ning Zhong. Computing geometric lorenz attractors with arbitrary precision. *Trans. Am. Math. Soc.*, (370):2955–2970, 2019.

[RY19] Cristobal Rojas and Michael Yampolsky. Computational intractability of attractors in the real quadratic family. *Advances in Math.*, 349:941–958., 2019.

[SfR14] Sylvain Soliman, François fages, and Ovidiu Radulescu. A constraint solving approach to model reduction by tropical equilibration. *Algorithms for Molecular Biology*, 9(24), December 2014.

[Sim10] Stephen G Simpson. Medvedev degrees of 2-dimensional subshifts of finite type. *To appear in Ergodic Theory and Dynamical Systems*, 2010.

[TAG+08] John J. Tyson, Reka Albert, Albert Goldbeter, Peter Ruoff, and Jill Sible. Biological switches and clocks. *J R Soc Interface*, 5(Suppl 1):S1–S8, August 2008.

[Uma17] Chris Umans, editor. *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*. IEEE Computer Society, 2017.

[vdH18] J. van der Hoeven. Creative telescoping using reductions, 2018. Technical Report, HAL 01773137.

[Wil82] Herbert S. Wilf. What is an answer? *Amer. Math. Monthly*, 89(5):289–292, 1982.

[Zhu17] Dmitriy Zhuk. A proof of CSP dichotomy conjecture. In Umans [Uma17], pages 331–342.