Circuit Complexity Meets Discrete Ordinary Differential Equations: An Overview

Melissa Antonelli[1,2], Arnaud Durand[3] and Juha Kontinen[1]

[1]University of Helsinki, Pietari Kalmin katu 5, Helsinki, 00560, Finland
[2]Helsinki Institute for Information Technology - HIIT, Helsinki, Finland
[3]Université Paris Cité, 8, place Aurélie Nemours, Paris, France

Abstract

Boolean and arithmetic circuits are receiving increasing attention in AI. In particular, due to the connection with neural networks, the evaluation of their computational resources, as studied in circuit complexity, has been thoroughly investigated to address concrete issues, such as scalability. In this paper, we outline an original approach to explore this area using recursion schemas defined by discrete ordinary differential equations (ODEs). Relying on them, we provide new characterisations of various small circuit classes, offering a uniform and comprehensive framework to potentially unveil the relationships between them and clarify notions like counting in this context. Here, we present a systematic and integrated overview of the main results obtained thus far, namely the introduction of algebras characterised by different constraints on the linearity defining their ODE schemas and capable of capturing function classes that range from those computed by unbounded fan-in circuits working in constant depth ($\mathbf{FAC}^0$) to bounded Boolean circuits working in logarithmic depth ($\mathbf{FNC}^1$), including those with majority ($\mathbf{FTC}^0$) and counting modulo 2 gates ($\mathbf{FACC}[\mathbf{2}]$). This would provide a starting point for discussing the various ongoing directions of this research and the potential challenges associated with this new perspective on the study of circuit complexity.

Keywords

Circuit Complexity, Ordinary Differential Equations, Implicit Computational Complexity, Neural Networks

1. Introduction

In recent decades, Boolean and arithmetic circuits have received increasing attention not only in traditional fields of computer science, such as database theory and communication complexity, but also in multiple areas of AI, from reinforcement learning and Bayesian networks to language and ML models [1, 2]. For example, it has been shown that circuits play a crucial role in neuro-symbolic AI, providing a new formalism that integrates logical and probabilistic reasoning [3]. In particular, neural networks have been extensively studied through the lens of circuits and Boolean functions: already in the 1990s, the expressive power of feed-forward neural networks has been related to (threshold) Boolean circuits [4, 5, 6], while more recently the expressivity of graph neural networks have been shown linked to that of constant-depth arithmetic circuits over the reals [7]. In this context, the number of neurons and the running time of the network can be seen as computational resources. Accordingly, circuit complexity has emerged as a valuable tool for examining neural networks and providing solutions to practical problems, for instance when addressing scaling issues [8].

Within this framework, we aim to provide an original viewpoint for investigating circuit complexity and enhancing the understanding of the relationship between small circuit classes and with arithmetic circuits. In this paper, we present an (integrated) overview of the main results achieved so far by generalising the approach introduced in [9], which captures functions computable in polynomial time ($\mathbf{FP}$) using discrete ordinary differential equations (ODEs), to the study of circuit complexity. Indeed, although small circuit classes have been characterised in different ways, investigating them through ODEs has only been attempted very recently [10, 11, 12]. In particular, it has emerged that, by slightly modifying the ODE-based schema characterising $\mathbf{FP}$, we can perform computations in (and, in several

OVERLAY 2025, 7th International Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis, October 26th, Bologna, Italy.
*Corresponding author: melissa.antonelli@helsinki.fi (M. Antonelli)

cases, characterise) multiple function classes, ranging from those computed by Boolean circuits with unbounded fan-in working in constant depth ($\mathbf{FAC}^0$) to those working in logarithmic depth ($\mathbf{FAC}^1$). This approach has also been proven to be strikingly natural: by applying simple restrictions on linearity, and performing derivation along functions of specific growth rate, we can capture the essence of small circuit computation. Additionally, many classical functions – from poly-log bit sum to parity and majority, from iterated sum and product to binary search – are shown to be easily encapsulated in linear ODE families. Accordingly, we believe that this new line of research, which is still in its early stages, can shed light on circuit computation and offer a simple, machine-independent framework for analysing notoriously hard problems and bounds in terms of *natural* and *uniform* constraints on linear ODEs. The goal of this work is to provide a comprehensive overview that includes and compares the key results achieved thus far (even considering a few new schemas not appearing in [10, 11, 12], e.g. ℓ-sODE), as well as to discuss challenges and future research directions.

2. Capturing Complexity Classes via ODEs

Implicit computational complexity is an active area of theoretical computer science that aims to provide machine-independent characterisations of relevant complexity classes. Specifically, one of the foundational results in the area of recursion theory was established by Cobham [13], who captured $\mathbf{FP}$ relying on a restricted recursion schema called bounded recursion on notation (BRN) and such that

$$\begin{aligned} f(0,\mathbf{y}) &= g(\mathbf{y}) \\ f(s_i(x),\mathbf{y}) &= h_i(x,\mathbf{y},f(x,\mathbf{y})) \quad x \neq 0 \\ f(x,\mathbf{y}) &\leq k(x,\mathbf{y}) \end{aligned}$$

for $i \in \{0,1\}$. In BRN, the growth of the defined function is controlled by another function k (to be in $\mathbf{FP}$), while the number of induction steps is kept under control by the application of the binary successor functions $s_i(x) = 2x + i, i \in \{0,1\}$. However, such a schema is in a sense not fully satisfactory as it imposes an explicit bound on recursion in the form of an already known function. Accordingly, Cobham's seminal work not only led to a variety of implicit characterisations for classes other than $\mathbf{FP}$, but also inspired alternative approaches to capture this class, e.g. relying on safe recursion [14] and ramification [15, 16]. Among them, the proposal by [9] has the advantage of neither imposing any explicit bound on the recursion schema nor assigning specific roles to variables. Indeed, it is based on special discrete ODEs, which combine two peculiar features: *derivation along specific functions*, so to control the number of computation steps, and *linearity*, namely a syntactic form of the equation allowing to control the object size.

Recall that the *discrete derivative of* $\mathbf{f}(x)$ is defined as $\Delta\mathbf{f}(x) = \mathbf{f}(x+1) - \mathbf{f}(x)$ and that ODEs are expressions of the form:

$$\frac{\partial \mathbf{f}(x,\mathbf{y})}{\partial x} = \mathbf{h}\big(x,\mathbf{y},\mathbf{f}(x,\mathbf{y})\big)$$

where $\frac{\partial \mathbf{f}(x,\mathbf{y})}{\partial x}$ stands for the derivative of $\mathbf{f}(x,\mathbf{y})$ considered as a function of x, for $\mathbf{y}$ fixed. When some initial value $\mathbf{f}(0,\mathbf{y}) = g(\mathbf{y})$ is added, this is called *Initial Value Problem* (IVP).

Notation 1. *Let* $\mathsf{sg} : \mathbb{Z} \to \mathbb{Z}$ *be the sign function over* $\mathbb{Z}$*, taking value 1 for* $x > 0$ *and 0 otherwise; let* $\mathsf{cosg} : \mathbb{Z} \to \mathbb{Z}$ *be the cosign function such that* $\mathsf{cosg}(x) = 1 - \mathsf{sg}(x)$*.*

A *(limited)* sg*-polynomial expression* $P(x_1,\ldots,x_h)$ is an expression built over the signature $\{+,-,\times\}$ ($\{+,-\}$, resp.), the function sg and a set of variables $X = \{x_1,\ldots,x_h\}$, plus integer constants. A sg-polynomial expression is said to be *essentially linear* in a set of variables $\mathbf{x}$, if there are sg-polynomial expressions Q_1, Q_2 such that it is of the form $Q_1 \times x + Q_2$, and x occurs in Q_1, Q_2 only under the scope of the sign function. The concept of linearity allows to control the growth of functions defined by ODEs.

Notation 2. *We use $\ell(x)$ to denote the length function, returning the length of x written in binary. Formally, it is defined as $\ell(0) = 0$ and $\ell(x) = \lceil log_2(x + 1)\rceil$.*

In what follows, we will consider functions $\mathbf{f} : \mathbb{N}^{p+1} \to \mathbb{Z}^d$, i.e. vectors of functions $\mathbf{f} = f_1, \dots, f_d$ from $\mathbb{N}^{p+1}$ to $\mathbb{Z}$ and introduce the linear ℓ-ODE schema, obtained by deriving along the length function ℓ.

Definition 1 (Linear ℓ-ODE)**.** *Given $\mathbf{g}, \mathbf{h}, \mathbf{u}$ and λ, the function $\mathbf{f}$ is said to be* linear λ-ODE *definable from $\mathbf{g}, \mathbf{h}$ and $\mathbf{u}$ if it is the solution of the IVP with initial value $\mathbf{f}(0, \mathbf{y}) = \mathbf{g}(\mathbf{y})$ and such that:*

$$\frac{\partial \mathbf{f}(x, \mathbf{y})}{\partial \lambda} = \mathbf{u}\big(x, \mathbf{y}, \mathbf{h}(x, \mathbf{y}), \mathbf{f}(x, \mathbf{y})\big),$$

with $\mathbf{u}$ essentially linear in $\mathbf{f}(x,\ \mathbf{y})$. If $\lambda = \ell$, this schema is called linear length-ODE, ℓ-ODE.

Intuitively, one of the key properties of λ-ODE is its dependence on the number of distinct values of λ, i.e. that the value of $\mathbf{f}(x, \mathbf{y})$ changes only when the value of $\lambda(x, \mathbf{y})$ does. Moreover, if $\mathbf{u}$ is essentially linear in $\mathbf{f}(x, \mathbf{y})$, as linear length-ODE is, there exist matrices $\mathbf{A}$ and $\mathbf{B}$, such that $\mathbf{f}(0, \mathbf{y}) = \mathbf{g}(\mathbf{y})$ and $\frac{\partial \mathbf{f}(x,\mathbf{y})}{\partial \ell} = \mathbf{A}\big(x, \mathbf{y}, \mathbf{h}(x, \mathbf{y}), \mathbf{f}(x, \mathbf{y})\big) \times \mathbf{f}(x, \mathbf{y}) + \mathbf{B}(x, \mathbf{y}, \mathbf{h}(x, \mathbf{y}), \mathbf{f}(x, \mathbf{y}))$. Then, for all x and $\mathbf{y}$, the solution $\mathbf{f}(x, \mathbf{y})$ defined by linear ℓ-ODE is as follows:

$$\sum_{u=-1}^{\ell(x)-1} \left(\prod_{t=u+1}^{\ell(x)-1} \Big(1 + \mathbf{A}(\alpha(t), \mathbf{y}, \mathbf{h}(\alpha(t), \mathbf{y}), \mathbf{f}(\alpha(t), \mathbf{y}))\Big) \right) \times \mathbf{B}\big(\alpha(u), \mathbf{y}, \mathbf{h}(\alpha(u), \mathbf{y}), \mathbf{f}(\alpha(u), \mathbf{y})\big),$$

where $\alpha(u) = 2^u - 1$ denotes the greatest integer whose length is u (see [17] for further details).

Example 1 (Function $2^{\ell(x)}$)**.** *The function $x \mapsto 2^{\ell(x)}$ can be rewritten as the solution of the IVP with initial value $f(0) = 1$ and such that $\frac{\partial f(x)}{\partial \ell} = f(x)$, i.e. a restricted case of linear ℓ-ODE with $A = 1$ and $B = 0$; that is, $f(x) = \prod_{u=0}^{\ell(x)-1} 2 = 2^{\ell(x)}$.*

One of the main results of [9] is the characterisation of $\mathbf{FP}$ by the algebra made of basic functions $0, 1, \mathsf{sg}, \ell, +, -, \times$ and the projection function π^p_i and closed under composition ($\circ$) and ℓ-ODE:

$$\mathbb{LDL} = [0, 1, \ell, \mathsf{sg}, +, -, \pi^p_i; \circ, \ell\text{-ODE}].$$

3. A New Approach to Circuit Complexity

In this section we present an overview of the characterisations of small circuit classes obtained by relying on different (strict and non-strict) ℓ-ODE schemas defined by imposing natural constraints on the linearity allowed by linear length-ODE.[1] This investigation has already demonstrated a strong link between depth constraints in circuit computation and the syntactical constraints that define corresponding linear length ODEs. In particular, it has emerged that strict schemas, i.e., schemas defined by ODEs not involving calls to $f(x, \mathbf{y})$ occurring under the scope of the sign function, are linked to constant-depth circuit computation, possibly including counting. While constant-depth classes can be fully captured due to them, it seems that a similar strict characterisation is not feasible for larger classes. Indeed, *full and strict* linear computation along ℓ is in $\mathbf{FTC}^0$. Additionally, some constraints over (non-strict) schemas have emerged to be particularly interesting in the small circuit setting, namely A taking values 1 or −1, B taking only values 0 or 1 or being strictly positive, and $f(x, \mathbf{y})$ occurring under the scope of the sign function only in expressions of the form $f(x, \mathbf{y}) - c$, for $c \in \mathbb{N}$.

Notation 3. *An equation and, by extension, an ODE schema is said to be* strict *when it does not include any call to $f(x, \mathbf{y})$ in A or B, not even under the scope of the sign function; a call to $f(x, \mathbf{y})$ is said to be a* simple *if $f(x, \mathbf{y})$ occurs in A and B only in expressions of the form $s = \mathsf{sg}(f(x, \mathbf{y}))$. We use $k(x, \mathbf{y})$ (resp., $K(x, \mathbf{y}, h(x, \mathbf{y}), f(x, \mathbf{y}))$) for functions (resp., expressions) taking values in $\{0, 1\}$.*

[1]Although here we focus on schemas obtained by deriving along the function ℓ, some upper bounds have also been shown for similar linear schemas obtained by deriving along $\ell_2 = (\ell \circ \ell)$ (see [12, Sec. 5]).

In [10], to deal with $\mathbf{FAC}^0$ computation, we introduced the following strict ℓ-ODE schemas:

Definition 2 (Schemas ℓ-ODE$_1$, ℓ-ODE$_3$). *Given $g : \mathbb{N}^p \to \mathbb{N}$ and $k : \mathbb{N}^{p+1} \to \{0,1\}$, the function $f : \mathbb{N}^{p+1} \to \mathbb{N}$ is obtained by ℓ-ODE$_1$ from g and k, if it is the solution of the IVP defined by $f(0,\mathbf{y}) = g(\mathbf{y})$ and such that:*

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + k(x,\mathbf{y}).$$

Analogously, f is defined by ℓ-ODE$_3$ from g if it is the solution of the IVP defined by $f(0,\mathbf{y}) = g(\mathbf{y})$ and

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = -\left\lceil \frac{f(x,\mathbf{y})}{2} \right\rceil.$$

Intuitively, ℓ-ODE$_1$ corresponds to iteratively left-shifting (the binary representation of) a given number, possibly adding 1, while ℓ-ODE$_3$ defines basic right-shifting computation. Notably, the latter schema is enough to encode the function $\mathsf{BIT}(x,y)$, computing the $\ell(x)^{th}$ bit of y. Relying on them we introduced the algebra below:

$$\mathbb{ACDL} = [0, 1, \ell, \mathsf{sg}, +, -, \div 2, \#, \pi^p_i; \circ, \ell\text{-ODE}_1, \ell\text{-ODE}_3]$$

and, due to it, characterise $\mathbf{FAC}^0$ [10, Theorem 18]. Notably, both existential and universal sharply bounded quantification, as well as minimisation can be naturally rewritten in $\mathbb{ACDL}$.

Remark 1 (Sharply Bounded Quantification and Minimisation). *Let $R \subseteq \mathbb{N}^{p+1}$ and h_R be its characteristic function. Then, for all x and $\mathbf{y}$, it holds that*

$$(\exists z \leq \ell(x))R(z,\mathbf{y}) = \mathsf{sg}(f(x,\mathbf{y}))$$

where f is the solution of the IVP:

$$f(0,\mathbf{y}) = h_R(0,\mathbf{y})$$
$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + h_R(\ell(x+1),\mathbf{y})$$

which is an instance of ℓ-ODE$_1$. Intuitively, $f(x,\mathbf{y}) \neq 0$ when, for some z smaller than $\ell(x)$, $R(z, \mathbf{y})$ is satisfied (i.e. $h_R(z,\mathbf{y}) = 1$): if such instance exists, our bounded search ends with a positive answer.

Universally bounded quantification can be expressed in a similar way. Let's consider $\mathsf{cosg}(f(x,\mathbf{y}))$ for f defined substituting the value of h_R with its co-sign. So, $(\forall z \leq \ell(x))R(z,\mathbf{y}) = \mathsf{cosg}(f(x,\mathbf{y}))$ and

$$f(0,\mathbf{y}) = \mathsf{cosg}(h_R(0,\mathbf{y}))$$
$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + \mathsf{cosg}(h_R(\ell(x+1),\mathbf{y})).$$

Also sharply bounded μ-operator can be rewritten in $\mathbb{ACDL}$ via ℓ-ODE$_1$. Let $\mu x.R(x,\mathbf{y})$ returns the smallest $i \leq \ell(x)$ such that $R(i,\mathbf{y})$. This can be rewritten as:

$$\mathsf{mu}_R(x,\mathbf{y}) = \ell(x) - \ell(f_\mu(x,\mathbf{y}))$$

where f_μ is the solution of the IVP below:

$$f_\mu(0,\mathbf{y}) = 0$$
$$\frac{\partial f_\mu(x,\mathbf{y})}{\partial \ell} = f_\mu(x,\mathbf{y}) + \exists i \leq \ell(x+1)\big(h_R(i,\mathbf{y}) = 1\big)$$

for $h_R \in \{0,1\}$ being the characteristic function of $R(i,\mathbf{y})$.

Additionally, the strict schema below is introduced to characterise $\mathbf{FTC}^0$:

Definition 3 (Schema ℓ-ODE$_2^*$). *Given $g : \mathbb{N}^p \to \mathbb{N}, k : \mathbb{N}^{p+1} \to \{0,1\}$ and $h : \mathbb{N}^p \to \mathbb{N}$, the function $f : \mathbb{N}^{p+1} \to \mathbb{N}$ is defined by ℓ-ODE$_2^*$ from g, h and k when it is the solution of the IVP with initial value $f(0,\mathbf{y}) = g(\mathbf{y})$ and such that*

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = \left(2^{\ell(h(\mathbf{y}))} - 1\right) \times f(x,\mathbf{y}) + k(x,\mathbf{y}).$$

If we add the constraint that $h(\mathbf{y}) \geq 1$, then ℓ-ODE$_2^*$ is called ℓ-ODE$_2$. Notably, the latter schema is not only in $\mathbf{FAC}^0$, but, together with ℓ-ODE$_3$ (or BIT), it is also enough to capture $\mathbf{FAC}^0$ without including $\#$ among the basic functions. In [12], an alternative, strict schema, called ℓ-pODE, is introduced to capture $\mathbf{FTC}^0$.

Definition 4 (Schema ℓ-pODE). *Given $g : \mathbb{N}^p \to \mathbb{N}$ and $h : \mathbb{N}^{p+1} \to \mathbb{N}$, the function $f : \mathbb{N}^{p+1} \to \mathbb{N}$ is defined by ℓ-pODE from g and h, if it is the solution of the IVP with initial value $f(0,\mathbf{y}) = g(\mathbf{y})$ and*

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = A\big(x,\mathbf{y},h(x,\mathbf{y})\big) \times f(x,\mathbf{y}) + B\big(x,\mathbf{y},h(x,\mathbf{y})\big)$$

where $A, B \geq 1$ are limited sg-polynomial expressions.

Computation through both ℓ-ODE$_2^*$ and ℓ-pODE is shown in $\mathbf{FTC}^0$ and sufficient to capture this class:

$$\begin{aligned}\mathbf{FTC}^0 &= [0,1,\ell,\mathsf{sg},+,-,\div 2,\pi_i^p;\circ,\ell\text{-ODE}_2^*,\ell\text{-ODE}_3]\\ &= [0,1,\ell,\mathsf{sg},\mathsf{BIT},+,-,\div 2,\#,\pi_i^p;\circ,\ell\text{-pODE}].\end{aligned}$$

In the context of a broader understanding of the hierarchy of small circuit classes, in [12], we also considered non-strict ℓ-ODE schemas. As mentioned, it is shown that IVP defined by linear equations in which $A \in \{-1,1\}$ and $B = 0$ or such that $f(x,\mathbf{y})$ occurs only under the scope of the sign function in expressions of the form $s = \mathsf{sg}(f(x,\mathbf{y}) - c)$ for $c \in \mathbb{N}$ (or $c = 0$, for *simple* calls) are particularly relevant. As long as $f(x,\mathbf{y})$ occurs in $A = K - 1$ only in expressions of the form $\mathsf{sg}(f(x,\mathbf{y}) - c)$, the corresponding computation is in $\mathbf{FAC}^0$. Indeed, the solution of such system corresponds to (sharply) bounded search. Here, we additionally consider the following more general, non-strict schema:

Definition 5 (Schema ℓ-sODE). *Given $g : \mathbb{N}^p \to \mathbb{N}$ and $k : \mathbb{N}^{p+1} \to \{0,1\}$, the function $f : \mathbb{N}^{p+1} \to \mathbb{N}$ is defined by ℓ-sODE if it is the solution of the IVP with initial value $f(0,\mathbf{y}) = g(\mathbf{y})$ and such that*

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + K(\mathbf{y}, f(x,\mathbf{y})) \times k(x,\mathbf{y})$$

where $K \in \{0,1\}$ is a limited sg-polynomial expression such that calls to $f(x,\mathbf{y})$ are simple.

Computation through this schema is in $\mathbf{FAC}^0$ and strong enough to rewrite ℓ-ODE$_1$.

Lemma 1. *If $f(x,\mathbf{y})$ is defined by ℓ-sODE from functions in $\mathbf{FAC}^0$, then $f(x,\mathbf{y})$ is in $\mathbf{FAC}^0$ as well.*

Proof. For any $i \in \{0,1\}$, we use $K_i(\mathbf{y})$ as a shorthand for $K(i,\mathbf{y})$, where i substitutes $\mathsf{sg}(f(x,\mathbf{y}))$. The expression $K_i(\mathbf{y})$ is computable in $\mathbf{FAC}^0$ by hypothesis. We consider the main cases:

- $K_0(\mathbf{y}) = K_1(\mathbf{y})$. If $K_1(\mathbf{y}) = 0$, then for any x, $f(x,\mathbf{y}) = g(\mathbf{y})$, which can be computed in $\mathbf{FAC}^0$ by hypothesis. If $K_1(\mathbf{y}) = 1$, then ℓ-sODE is nothing but ℓ-ODE$_1$, the computation of which is shown in $\mathbf{FAC}^0$ in [10].
- $K_0(\mathbf{y}) \neq K_1(\mathbf{y})$. If $g(\mathbf{y}) \geq 1$, then for any x, $\mathsf{sg}(f(x,\mathbf{y})) = 1$, so $K(f(x,\mathbf{y}),\mathbf{y}) = K_1(\mathbf{y})$, and the proof is precisely as before, i.e. if $K_1(\mathbf{y}) = 0$, $f(x,\mathbf{y}) = g(\mathbf{y})$; otherwise, this is just an instance of ℓ-sODE. If $g(\mathbf{y}) = 0$, then there are two possible sub-cases: (i) if $K_0(\mathbf{y}) = 0$, then $f(x,\mathbf{y}) = 0$; (ii) if $K_0(\mathbf{y}) = 1$ (and $K_1(\mathbf{y}) = 0$), then if $k(0,\mathbf{y}) = 1$, $f(x,\mathbf{y}) = 2^{\ell(\mathbf{y})-1}$ and if $k(0,\mathbf{y}) = 0$, $f(x,\mathbf{y}) = 0$. Both solutions are computable in $\mathbf{FAC}^0$.

□

The schema ℓ-sODE provides the first characterisation of $\mathbf{FAC}^0$ based on a *non-strict* schema, thus offering a way to compare the defining schema of this class with those defining classes from the levels $\mathbf{FNC}^1$ and $\mathbf{FAC}^1$ and below (recall that these classes have not characterised by strict schemas).

Remark 2. *If $K(f(x,\mathbf{y}),\mathbf{y}) = 1$, then ℓ-sODE is equal to ℓ-ODE$_1$.*

Proposition 1. $\mathbf{FAC}^0 = [0, 1, \ell, \mathsf{sg}, +, -, \div 2, \#, \pi^p_i; \circ, \ell\text{-sODE}, \ell\text{-ODE}_3]$

Proof. ($\supseteq$) Analogous to [10, Th. 8], plus Lemma 1. ($\subseteq$) By Remark and [10, Th. 8]. □

In [12], two non-strict schema, obtained deriving along ℓ and restricting linearity so that $A = -1$ (in both cases) and B is either in $\{0, 1\}$ or non-negative, are introduced to capture the class of functions computed by constant-depth unbounded fan-in circuits including modulo 2 counting gates ($\mathbf{FACC}[\mathbf{2}]$) and by logarithmic-depth bounded fan-in Boolean circuits ($\mathbf{FNC}^1$), respectively.

Definition 6 (Schemas ℓ-b$_0$ODE and ℓ-bODE). *Given $g : \mathbb{N}^p \to \mathbb{N}$ and $h : \mathbb{N}^{p+1} \to \mathbb{N}$, the function $f : \mathbb{N}^{p+1} \to \mathbb{N}$ is defined by ℓ-bODE if it is the solution of the IVP with initial value $f(0,\mathbf{y}) = g(\mathbf{y})$ and such that*

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = -f(x,\mathbf{y}) + B\big(x,\mathbf{y},h(x,\mathbf{y}),f(x,\mathbf{y})\big)$$

where, for any x and $\mathbf{y}$, $B(x,\mathbf{y},h(x,\mathbf{y}),f(x,\mathbf{y})) \geq 0$ is a sg-polynomial expression, and $f(x,\mathbf{y})$ occurs in it only in expressions of the form $\mathsf{sg}(f(x,\mathbf{y}) - c)$, being c constant. In the special case of $B = K \in \{0,1\}$ and being a limited sg-polynomial expression such that $f(x,\mathbf{y})$ occurs in it only under the scope of the sign function, this schema is called ℓ-b$_0$ODE.

It is shown that each of them is sufficient to characterize the corresponding class, namely:

$$\mathbf{FACC}[\mathbf{2}] = [0, 1, \ell, \mathsf{sg}, \mathsf{BIT}, +, -, \div 2, \#, \pi^p_i; \circ, \ell\text{-pODE}, \ell\text{b}_0\text{ODE}]$$
$$\mathbf{FNC}^1 = [0, 1, \ell, \mathsf{sg}, \mathsf{BIT}, +, -, \div 2, \#, \pi^p_i; \circ, \ell\text{-pODE}, \ell\text{-bODE}].$$

Additionally, computation through the non-strict and simple schema such that if $g : \mathbb{N}^p \to \mathbb{N}$ and $h : \mathbb{N}^{p+1} \to \mathbb{N}$, then $f : \mathbb{N}^{p+1} \to \mathbb{N}$ defined by the IVP with initial value $f(0,\mathbf{y}) = g(\mathbf{y})$ and such that:

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = f(x,\mathbf{y}) + K(x,\mathbf{y},h(x,\mathbf{y}),f(x,\mathbf{y}))$$

where $K \in \{0,1\}$ is a sg-polynomial expression, and $f(x,\mathbf{y})$ appears in it only in expressions of the form $\mathsf{sg}(f(x,\mathbf{y}))$, is in $\mathbf{FNC}^1$. Remarkably, when $A = 1$ and $f(x,\mathbf{y})$ occurs under the scope of the function BIT only, this schema not only is in $\mathbf{FNC}^1$, but it also provides an alternative characterisation of this class. In this case completeness can be proven towards a direct encoding of the corresponding machine model. On the other hand, when $A = 0$ and $B = K \in \{0,1\}$ or $A = K \in \{0,1\}$ and $B = 0$, i.e. when dealing with IVP with initial value $f(0,\mathbf{y}) = g(\mathbf{y})$ and such that

$$\frac{\partial f(x,\mathbf{y})}{\partial \ell} = K(x,\mathbf{y},h(x,\mathbf{y}),f(x,\mathbf{y})) \qquad \frac{\partial f(x,\mathbf{y})}{\partial \ell} = K(x,\mathbf{y},h(x,\mathbf{y}),f(x,\mathbf{y})) \times f(x,\mathbf{y})$$

the corresponding computation is not only in $\mathbf{FNC}^1$, but also in $\mathbf{FTC}^0$.

4. Future Work

In this paper, we presented the global picture of the main results obtained so far to characterise small circuit computation via discrete ODEs and integrate them with a few new achievements, providing a *uniform* and *novel* approach for studying circuit complexity and provide original strategies (e.g., coming from the calculus of finite differences) to tackle notoriously hard open problems in this field, such as separation results. This would serve as a starting point for future research. Indeed, given the novelty of this study, many directions are open for investigation. Natural ones include ODE-based

Table 1
Summary of the features defining main ℓ-ODE schemas known to be in the corresponding class or (in blue) to characterise it. For readability, the arguments h and $\mathbf{y}$ are omitted. Calls in non-strict schemas are simple or restricted to $s = \mathrm{sg}(f(x) - c)$, for $c \in \mathbb{N}$, except for $\mathbf{FACC}[\mathbf{2}]$, and B is assumed to take positive values only.

	strict schemas	non-strict schemas
$\mathbf{FAC}^0$	$(k(x) - 1) \times f(x)$ $f(x) + k(x)$	$(K(x, f) - 1) \times f(x)$ $f + K(f) \times k(x)$
$\mathbf{FACC}[\mathbf{2}]$	$(k(x) - 2) \times f(x)$	$-f(x) + K(x, f)$
$\mathbf{FTC}^0$	$(2^{\ell(h(\mathbf{y}))} - 1) \times f(x) + k(x)$ $A(x) \times f(x) + B(x)$	$K(x, f)$ $K(x, f) \times f(x)$
$\mathbf{FNC}^1$		$-f(x) + B(x, f)$

characterisations of the $\mathbf{FAC}$ and $\mathbf{FNC}$ hierarchies, obtained by endowing algebras defining their base levels with (variously restricted) schemas obtained deriving along slowly growing function. Related to this is the exploration of the expressive power of schemas obtained by deriving along $\ell_2 (= \ell \circ \ell)$, in which calls to $f(x, \mathbf{y})$ occur only under the scope of the function BIT. Another work in progress involves the investigation of (strict) ℓ-ODEs that correspond to constant-depth classes with modulo counting gates, which lie between $\mathbf{FACC}[\mathbf{2}]$ and $\mathbf{FTC}^0$; this would possibly enhance our understanding of the concept of counting within the framework of small circuits, and more broadly. Other intriguing directions include the generalisation of our approach to the continuous setting by analysing small circuit classes over the reals, as outlined in [18, 19] for $\mathbf{FP}$ and $\mathbf{FPSPACE}$, and the development of proof-theoretical counterparts to our ODE-style algebras, in the form of natural rule systems inspired by the ODE design, allowing to syntactically characterise the corresponding classes.

5. Acknowledgments

Since 2023, the research of Melissa Antonelli has been funded by the Helsinki Institute for Information Technology. The research of Arnaud Durand is partially funded by ANR Grant ∂ifference. Melissa Antonelli and Arnaud Durand thank Maupertuis' SMR Programme, on behalf of the Institute Français in Helsinki, the French Embassy to Finland, the French Ministry of Higher Education, Research and Innovation in partnership with the Finnish Society for Science and Letters and the Finnish Academy of Sciences and Letters, for the financial support. Melissa Antonelli, Arnaud Durand and Juha Kontinen thank Magnus Ehrnrooth for supporting the CARACAL project.

References

[1] A. Darwiche, Tractable boolean and arithmetic circuits, in: P. Hitzler, M. K. Sarker (Eds.), Neuro-Symbolic Artificial Intelligence: The State of the Art, IOS Press, 2005, pp. 146–172.
[2] A. R. A. Arora, S. Eyuboglu, Efficient language models as arithmetic cicuits, 2024. URL: https://hazyresearch.stanford.edu/blog/2024-06-22-ac.
[3] D. Fierens, G. V. den Broeck, J. Renkens, D. Shterionov, B. Gutmann, I. Thon, G. Hanssens, L. D. Readt, Inference and learning in probabilistic logic programs using weighted boolean formulas, Theory Pract. Log. Program 15 (2015) 358–401.
[4] E. Allender, K. W. Wagner, Counting hierarchies: Polynomial time and constant, Bulletin of the EATCS 40 (1990) 182–194.
[5] G. S. W. Maass, E. D. Sontag, On the computational power of sigmoid versus boolean threshold circuits, in: Proc. FOCS, 1991.
[6] W. Maass, Bounds for the computational power and learning complexity of analog neural nets, SIAM J. Comput. 26 (1997) 708–732.

[7] T. Barlag, V. Holzapfel, L. Strieker, J. Virtema, H. Vollmer, Graph neural networks and arithmetic circuits, in: Proc. NeurIPS, 2024.
[8] I. Parberry, Circuit Complexity and Neural Networks, The MIT Press, 1994.
[9] O. Bournez, A. Durand, Recursion schemes, discrete differential equations and characterization of polynomial time computation, in: Proc. MFCS, 2019.
[10] M. Antonelli, A. Durand, J. Kontinen, A new characterization of FAC^0 via discrete ordinary differential equations, in: Proc. MFCS, 2024, pp. 10:1–10:18.
[11] M. Antonelli, A. Durand, J. Kontinen, Towards new characterizations of small circuit classes via discrete ordinary differential equations (short paper), in: Proc. ICTCS, 2024, pp. 166–172.
[12] M. Antonelli, A. Durand, J. Kontinen, Characterizing small circuit classes from FAC^0 to FAC^1 via discrete ordinary differential equations, in: Proc. MFCS, 2025.
[13] A. Cobham, The intrinsic computational difficulty of functions, in: Logic, Methodology and Phylosophy of Science: Proc. 1964 International Congress, 1965, pp. 24–30.
[14] S. Bellantoni, S. Cook, A new recursion-theoretic characetrization of poly-time functions, Comput. Complex. 2 (1992) 97–110.
[15] D. Leivant, Predicative recurrence and computational complexity i: Word recurrence and poly-time, in: Feasible Mathematics, 1994, pp. 320–343.
[16] D. Leivant, J.-Y. Marion, Ramified recurrence and computational complexity ii: Substitution and poly-space, in: Proc. CSL, 1995, pp. 369–380.
[17] O. Bournez, A. Durand, A characterization of functions over the integers computable in polynomial time using discrete differential equations, Comput. Complex. 32 (2023).
[18] M. Blanc, O. Bournez, A characterization of functions computable in polynomial time and space over the reals with discrete ordinary differential equations: simulation of Turing Machines with analytic discrete ODEs, in: Proc. MFCS, 2023, pp. 21:1–21:15.
[19] M. Blanc, O. Bournez, The complexity of computing in continuous time: Space complexity is precision, in: Proc. ICALP, 2024, pp. 129:1–129:22.