

An analog characterization of the subrecursive functions

Manuel Lameiras Campagnolo^{1,2}, Christopher Moore^{2,3,4}, and José Félix Costa⁵

¹ D.M./I.S.A., Universidade Técnica de Lisboa, Tapada da Ajuda, 1399 Lisboa Cx,
Portugal `mlc@math.isa.utl.pt`

² Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, New Mexico 87501
`{mlc,moore}@santafe.edu`

³ Computer Science Department, University of New Mexico, Albuquerque NM 87131

⁴ Physics Department, University of New Mexico, Albuquerque NM 87131

⁵ D.M./I.S.T., Universidade Técnica de Lisboa, Av. Rovisco Pais, 1049-001 Lisboa,
Portugal `fgc@math.ist.utl.pt`

Abstract. We study a restricted version of Shannon's General Purpose Analog Computer in which we only allow the machine to solve linear differential equations. This corresponds to only allowing local feedback in the machine's variables. We show that if this computer is allowed to sense inequalities in a differentiable way, then it can compute exactly the elementary functions. Furthermore, we show that if the machine has access to an oracle which computes a function $f(x)$ with a suitable growth as x goes to infinity, then it can compute functions on any given level of the Grzegorzczuk hierarchy. More precisely, we show that the model contains exactly the n th level of the Grzegorzczuk hierarchy if it is allowed to solve $n - 3$ non-linear differential equations of a certain kind. Therefore, we claim that there is a close connection between analog complexity classes, and the dynamical systems that compute them, and classical sets of subrecursive functions.

Key words: Analog computation, differential equations, recursion theory, dynamical systems, Grzegorzczuk hierarchy, elementary functions, primitive recursive functions, subrecursive functions.

1 Introduction

Analog computation, where the internal states of a computer are continuous rather than discrete, has enjoyed a recent resurgence of interest. This stems partly from a wider program of exploring alternative approaches to computation, such as quantum and DNA computation; partly as an idealization of numerical algorithms where real numbers can be thought of as quantities in themselves, rather than strings of digits; and partly from a desire to use the tools of computation theory to better classify the variety of continuous dynamical systems we see in the world (or at least in classical idealizations of it).

However, in most recent work on analog computation (e.g. [BSS89,Mee93,Sie98,Moo98] time is still discrete; just as in standard computation theory, the machines are updated with each tick of a clock. If we are to make the states of a computer continuous, it makes sense to consider making its progress in time continuous too. While a few efforts have been made in the direction of studying computation by continuous-time dynamical systems [Moo90,Moo96,Orp97a,Orp97b,SF98,Bou99,CMC99], no particular set of definitions has become widely accepted, and the various models do not seem to be equivalent to each other. Thus analog computation has not yet experienced the unification that digital computation did through Turing’s work in 1936.

In this paper, as in [CMC99], we take as our starting point Claude Shannon’s General Purpose Analog Computer (GPAC). This was defined as a mathematical model of an analog device, the Differential Analyser, the fundamental principles of which were described by Lord Kelvin in 1876 [Tho76]. The Differential Analyser was developed at MIT under the supervision of Vannevar Bush and was indeed built in for the first time in 1931 [Bow96]. The Differential Analyser’s input was the rotation of one or more drive shafts and its output was the rotation of one or more output shafts. The main units were interconnected gear boxes and mechanical friction wheel integrators.

Just as polynomial operations are basic to the Blum-Shub-Smale model of analog computation [BSS89], polynomial differential equations are basic to the GPAC. Shannon [Sha41] showed that the GPAC generates exactly the *differentially algebraic* functions, which are unique solutions of polynomial differential equations. This set of functions includes simple functions like e^x and $\sin x$ as well as sums, products, and compositions of these, and solutions to differential equations formed from them such as $f' = \sin f$. Pour-El [PE74] extended Shannon’s work and made it rigorous.

The GPAC also corresponds to the lowest level in a theory of recursive functions on the reals proposed by Moore [Moo96]. There, in addition to composition and integration, a zero-finding operator analogous to the minimization operator μ of classical recursion theory is included. In the presence of a liberal semantics that defines $f(x) \times 0$ as 0 even when f is undefined, this permits contraction of infinite computations into finite intervals, and renders the arithmetical and analytical hierarchies computable through a series of limit processes similar to those used by Bournez [Bou99]. However, such an operator is clearly unphysical, except when the function in question is smooth enough for zeroes to be found in some reasonable way.

In [CMC99] a new extension of GPAC was proposed. The operators of the GPAC were kept the same — integration and composition — but piecewise-analytic basis functions were added, namely $\theta_k(x) = x^k \theta(x)$, where $\theta(x)$ is the Heaviside step function, $\theta(x) = 1$ for $x \geq 0$ and $\theta(x) = 0$ for $x < 0$. Adding these functions as ‘oracles’ can be thought of as

allowing an analog computer to measure inequalities in a $(k-1)$ -times differentiable way. These functions are also unique solutions of differential equations such as $xy' = ky$ if we define two boundary conditions rather than just an initial condition, which is a slightly weaker definition of uniqueness than that used by Pour-El to define GPAC-computability. By adding these to the basis set, we get a class we denote by $\mathcal{G} + \theta_k$ for each k .

A basic concern of computation theory is whether a given class is closed under various operations. One such operation is *iteration*, where from a function $f(x)$ we define a function $F(x, t) = f^{[t]}(x)$, i.e. f applied t times to x , for $t \in \mathbb{N}$. The main result of [CMC99] is that $\mathcal{G} + \theta_k$ is closed under iteration for any k , while \mathcal{G} is not. (Here we adopt the convention that a function where one or more inputs are integers is in a given analog class if some extension of it to the reals is.) It then follows that $\mathcal{G} + \theta_k$ includes all primitive recursive functions, and has other closure properties as well.

To refine these results, in this paper we consider a restricted version of Shannon's GPAC. In particular, we restrict integration to linear integration, i.e. solving linear differential equations. In terms of analog circuits, this means that only local feedback between variables of the analog computer is allowed. We define then a class of computable functions \mathcal{L} whose operators are composition and linear integration along with, as before, the functions θ_k .

The model we obtain, $\mathcal{L} + \theta_k$, is weaker than $\mathcal{G} + \theta_k$. One of the main results of this paper is that, for any $k > 2$, $\mathcal{L} + \theta_k$ contains precisely the elementary functions, a subclass of the primitive recursive functions introduced by Kalmar [Kal43] which is closed under the operations of forming bounded sums and products. This class contains virtually any function that can be computed in a practical sense, as well as the important number-theoretic and metamathematical functions [Cut80, Ros84]. Thus we seem to have found a natural analog description of the elementary functions.

To generalize this further, we recall that Grzegorzczuk [Grz53] proposed a hierarchy of computable functions that stratifies the set of primitive recursive functions. The elementary functions are simply the third level of this hierarchy. We show that if $\mathcal{L} + \theta_k$ is extended with 'oracles,' i.e. additional basis functions, which grow sufficiently quickly as their input goes to infinity, the resulting class can reach any level of the Grzegorzczuk hierarchy. Alternately, we reach the n th level if and only if we allow the system to solve $n - 3$ non-linear differential equations of a certain kind.

Therefore, we claim that there is a surprising and elegant connection between classes of analog computers on the one hand, and subclasses of the recursive functions on the other. This suggests, at least in this region of the complexity hierarchy, that analog and digital computation may not be so far apart.

The paper is organized as follows. In Section 2 we review classical recursion theory, the elementary functions, and the Grzegorzczuk hierarchy. In Section 3 we recall some basic facts about linear differential equations. Then, in Section 4 we define a general model of computation in continuous time that can access a set of 'oracles' or basis functions, compose these, and solve linear differential equations. We call this class $\mathcal{L} + \theta_k$, or more generally $\mathcal{L} + \varphi$ for a set of oracles φ .

We then prove bounds on the growth of functions definable in $\mathcal{L} + \theta_k$. The existence of those bounds allows us to prove the main lemma of the paper, which shows that $\mathcal{L} + \theta_k$ is closed under forming bounded sums and bounded products. With this, we are able

to prove that $\mathcal{L} + \theta_k$ contains all elementary functions. Inversely, using Grzegorzcyk and Lacombe's definition of computable continuous real function [Grz55,Lac55], we show that all functions in $\mathcal{L} + \theta_k$ are elementarily computable, and that if a function $f \in \mathcal{L} + \theta_k$ is an extension to the reals of some function \tilde{f} on the integers, then \tilde{f} is elementary as well. This shows that the correspondence between $\mathcal{L} + \theta_k$ and the elementary functions is quite robust.

Then, in Section 5 we consider the higher levels in the Grzegorzcyk hierarchy by defining a hierarchy of analog classes $\mathcal{G}_n + \theta_k$. Each one of these classes is defined similarly to $\mathcal{L} + \theta_k$ except that a new oracle is added to its basis. This new oracle is a solution of a non-linear differential equation, which produces iterations of some function in $\mathcal{G}_{n-1} + \theta_k$. We then show that this hierarchy coincides, level by level, with the Grzegorzcyk hierarchy in the same sense that $\mathcal{L} + \theta_k$ coincides with the elementary functions.

Finally, we end with some remarks and open questions.

2 Subrecursive classes over \mathbb{N} and the Grzegorzcyk hierarchy

In classical recursive function theory, where the inputs and output of functions are natural numbers \mathbb{N} , computational classes are often defined as the smallest set containing a set of initial functions and closed under certain operations, which take one or more functions in the class and create new ones. Thus the set consists of all those functions that can be generated from the initial ones by applying these operations a finite number of times. Typical operations include (where \mathbf{x} represents a vector of variables, which may be absent):

1. *Composition*: Given an n -ary function f and a function g with n components, define $(f \circ g)(\mathbf{x}) = f(g_1(\mathbf{x}), \dots, g_n(\mathbf{x}))$.
2. *Primitive recursion*: Given f and g of the appropriate arity, define h such that $h(\mathbf{x}, 0) = f(\mathbf{x})$ and $h(\mathbf{x}, y + 1) = g(\mathbf{x}, y, h(\mathbf{x}, y))$ for all $y \geq 0$.
3. *Limited recursion*: Given f , g and b , define h as in primitive recursion but only on the condition that $h(\mathbf{x}, y) \leq b(\mathbf{x}, y)$. Thus h is only allowed to grow as fast as another function already in the class.
4. *Bounded sum*: Given $f(\mathbf{x}, y)$, define $h(\mathbf{x}, y) = \sum_{z < y} f(\mathbf{x}, z)$, with $h(\mathbf{x}, 0) = 0$.
5. *Bounded product*: Given $f(\mathbf{x}, y)$, define $h(\mathbf{x}, y) = \prod_{z < y} f(\mathbf{x}, z)$, with $h(\mathbf{x}, 0) = 1$.
6. *Minimization* or *Zero-finding*: Given $f(\mathbf{x}, y)$, define $h(\mathbf{x})$ as the smallest y such that $f(\mathbf{x}, y) = 0$. If no such y exists, h is undefined.
7. *Bounded minimization*: Given $f(\mathbf{x}, y)$, define $h(\mathbf{x}, y_{\max})$ as the smallest $y < y_{\max}$ such that $f(\mathbf{x}, y) = 0$ and $f(\mathbf{x}, z)$ is defined for all $z \leq y$, and as y_{\max} if no such y exists.

Note that minimization is the only one of these that can be undefined; all the others lead to total functions. In bounded minimization, we only check values of y up to a certain maximum, and return a definite value if we fail to find any.

By starting with various basis sets and demanding closure under various properties, we can define various natural complexity classes:

1. *Partial recursive* functions are those that can be generated from the constant zero, the successor function $\mathcal{S}(x) = x + 1$, and projections $U_i(\mathbf{x}) = x_i$ using composition, primitive recursion and minimization.
2. *Primitive recursive* functions are those that can be generated from zero, successor, and projections using composition and primitive recursion.
3. *Elementary* functions are those that can be generated from zero, successor, projections, addition, and cut-off subtraction using composition and the operation of forming bounded sums and bounded products. Here *cut-off subtraction* is defined as $x \dot{-} y = x - y$ if $x \geq y$ and 0 if $x < y$.

The class of elementary functions, which we will call \mathcal{E} , was introduced by Kalmar [Kal43]. As examples, note that multiplication and exponentiation over \mathbb{N} are both in \mathcal{E} , since they can be written as bounded sums and products respectively: $xy = \sum_{z < y} x$ and $x^y = \prod_{z < y} x$. Since \mathcal{E} is closed under composition, for each m the m -times iterated exponential $\exp^{[m]}(x)$ is in \mathcal{E} , where $\exp^{[m+1]}(x) = 2^{\exp^{[m]}(x)}$ and $\exp^{[0]}(x) = x$. In fact, no elementary function can grow faster than $\exp^{[m]}$ for some m , and many of our results will depend on the following bound on their growth [Cut80]:

Proposition 1. *If $f(\mathbf{x}) \in \mathcal{E}$, there is a number m such that, for all \mathbf{x} , $f(\mathbf{x}) \leq \exp^{[m]}(\|\mathbf{x}\|)$, where $\|\mathbf{x}\| = \max_i x_i$.*

The elementary functions are exactly the functions computable in elementary time [Cut80]. The class \mathcal{E} is therefore very large, and many would argue that it contains all *practically* computable functions. It includes, for instance, the connectives of propositional calculus, functions for coding and decoding sequences of natural numbers such as the prime numbers and factorizations, and most of the useful number-theoretic and metamathematical functions [Cut80, Ros84].

However, \mathcal{E} does not contain all partial recursive functions, or even all primitive recursive ones. For instance, Proposition 1 shows that it does not contain the iterated exponential $\exp^{[m]}(x)$ where m is a variable, since any function in \mathcal{E} has an upper bound where m is fixed. To include all primitive recursive functions, we review the *Grzegorzcyk hierarchy* [Grz53, Ros84].

Definition 2 (The Grzegorzcyk hierarchy). Let \mathcal{E}^0 denote the smallest class containing zero, the successor function, and the projections, and which is closed under composition and limited recursion. Let \mathcal{E}^{n+1} be defined similarly, except with the function E_n added to the list of initial functions, where E_n is defined as follows:

$$\begin{aligned}
 E_0(x, y) &= x + y \\
 E_1(x) &= x^2 + 2 \\
 E_{n+1}(0) &= 2 \\
 E_{n+1}(x + 1) &= E_n(E_{n+1}(x)) = E_n^{[x]}(2)
 \end{aligned}$$

where by $f^{[x]}$ we mean f iterated x times.

The functions E_n are, essentially, repeated iterations of the successor function, and each one grows qualitatively faster than the previous one. $E_1(x)$ grows quadratically, and

composing it with itself produces functions that grow as fast as any polynomial. $E_2(x)$ grows roughly as 2^{2^x} , and composing it yields functions as large as $\exp^{[m]}$ for any fixed m . $E_3(x)$ grows roughly as $\exp^{[2x]}(2)$, and so on. (E_0 is defined with two variables for partly technical and partly historical reasons.)

We will use the fact that for $n \geq 3$, we can replace limited recursion in the definition of \mathcal{E}^n with bounded sum and bounded product [Ros84, p.112]:

Proposition 3. *For $n \geq 3$, \mathcal{E}^n is the smallest class containing zero, successor, the projections, and E_{n-1} , which is closed under composition, bounded sum, and bounded product.*

One consequence of this is that the elementary functions are simply the third level of the Grzegorzcyk hierarchy, where E_2 is included as an initial function [Ros84]:

Proposition 4. $\mathcal{E} = \mathcal{E}^3$.

Moreover, the union of all the levels of the Grzegorzcyk hierarchy is simply the class \mathcal{PR} of primitive recursive functions:

Proposition 5. $\mathcal{PR} = \cup_n \mathcal{E}^n$.

It can be shown that the class of primitive recursive functions can be generated using iteration instead of primitive recursion [Odi89, p.72]. This means that iteration cannot be used freely in the Grzegorzcyk hierarchy. Rather, as the definitions suggest, iteration moves a function one level up:

Proposition 6. *If $f \in \mathcal{E}^n$ then the iteration $F(x,t) = f^{[t]}(x) \in \mathcal{E}^{n+1}$.*

We can also generalize Proposition 1 and put a bound on the growth of functions anywhere in the Grzegorzcyk hierarchy:

Proposition 7. *If $n \geq 2$ and $f \in \mathcal{E}^n$ then there is an integer m such that $f(\mathbf{x}) \leq E_{n-1}^{[m]}(\|\mathbf{x}\|)$, where $\|\mathbf{x}\| = \max_i x_i$.*

The two previous lemmas are obviously related. They show that *finite* iteration of E_{n-1} for a fixed number of levels gives a bound on any function in \mathcal{E}^n , but unbounded iteration of E_{n-1} defines E_n and generates precisely \mathcal{E}^{n+1} . In this sense, the Grzegorzcyk hierarchy stratifies the primitive recursive functions according to how many levels of iteration are needed to define them.

All functions in the Grzegorzcyk hierarchy are *subrecursive*, meaning that they are computable in a number of steps which is a less computationally complex function of the input than the function itself [Ros84].

3 Linear differential equations

An ordinary linear differential equation is a differential equation of the form

$$\mathbf{x}'(t) = A(t)\mathbf{x}(t) + \mathbf{b}(t), \quad (1)$$

where $A(t)$ is a $n \times n$ matrix whose entries are functions of t and $\mathbf{b}(t)$ is a vector of functions of t . If $\mathbf{b}(t) = 0$ we say that the system is homogeneous. We can reduce a non-homogeneous system to a homogeneous one by introducing an auxiliary variable x_{n+1} such that $x_{n+1}(t) = 1$ for all t , that is, which satisfies $x_{n+1}(0) = 1$ and $x'_{n+1} = 0$. The new matrix will just be

$$\left[\begin{array}{c|c} A(t) & \mathbf{b}(t) \\ \hline 0 \dots 0 & 0 \end{array} \right].$$

This matrix is not invertible, which makes (1) harder to solve. However, since we don't need to solve the system explicitly, we prefer to consider the homogeneous equation

$$\mathbf{x}' = A(t)\mathbf{x} \quad (2)$$

as the general case in the remainder of the paper.

The fundamental existence theorem for differential equations guarantees the existence and uniqueness of a solution in a certain neighborhood of an initial condition for the system $\mathbf{x}' = f(\mathbf{x})$ when f is Lipschitz. For linear differential equations, we can strengthen this to global existence whenever $A(t)$ is continuous, and establish a bound on \mathbf{x} that depends on $\|A(t)\|$.¹

Proposition 8 ([Arn96]). *If $A(t)$ is defined and continuous on an interval $I = [a, b]$ where $a \leq 0 \leq b$, then the solution of Equation 2 with initial condition $\mathbf{x}(0) = \mathbf{x}_0$ is defined and unique on I . Furthermore, this solution satisfies*

$$\|\mathbf{x}(t)\| \leq \|\mathbf{x}_0\| e^{Ct} \quad (3)$$

where $C = C(a, b)$ is an upper bound on $\|A(t)\|$ on I , that is, $\|A(t)\| \leq C$ for $a \leq t \leq b$.

Therefore, if $A(t)$ is continuous on \mathbb{R} then solutions of linear differential equations are defined on arbitrarily large domains and have an exponential bound on their growth that depends only on $\|A(t)\|$. The last result holds both for the max norm, $\|\mathbf{x}\| = \max\{|x_1|, \dots, |x_n|\}$, and for the Euclidean norm [Har82]. If we use the max norm, which satisfies $\|A(t)\| \leq n \max_{ij}\{|A_{ij}(t)|\}$ [HJ85], then

$$\|\mathbf{x}(t)\| \leq \|\mathbf{x}_0\| \exp(n \max_{ij}\{|A_{ij}(t)|\}t). \quad (4)$$

when the conditions of Proposition 8 are fulfilled.

¹ By $\|\cdot\|$ we denote both the norm of a vector and the norm of a matrix, with $\|A\| = \sup\{\|A\mathbf{x}\| : \|\mathbf{x}\| = 1\}$.

4 The analog class $\mathcal{L} + \theta_k$ and the elementary functions

In [Moo96,CMC99] a definition of Shannon's *General Purpose Analog Computer* (GPAC) in the framework of the theory of recursive functions on the reals is given and is denoted by \mathcal{G} . It is the set of functions that can be inductively defined from the constants 0, 1, and -1 , projections, and the operations of composition and integration. Integration generates new functions by the following rule: if f and g belong to \mathcal{G} then the function h defined by the initial condition $h(\mathbf{x}, 0) = f(\mathbf{x})$ and the differential equation $\partial_y h(\mathbf{x}, y) = g(\mathbf{x}, y, h(\mathbf{x}, y))$ also belongs to \mathcal{G} , over the largest interval containing 0 on which the solution is finite and unique. Thus the GPAC has the power to solve arbitrary differential equations, constructed recursively from functions it already contains.

We define here a proper subclass of \mathcal{G} which we call \mathcal{L} , by restricting the integration operator to solving time-varying linear differential equations. To make the definition more general, we add a set of 'oracles' or additional basis functions. Let φ be a set of continuous functions defined everywhere on \mathbb{R}^k for some k . Then $\mathcal{L} + \varphi$ is the class of functions of several real variables defined recursively as follows:

Definition 9. A function $h : \mathbb{R}^m \rightarrow \mathbb{R}^n$ belongs to $\mathcal{L} + \varphi$ if its components can be inductively defined from the constants 0, 1, -1 , and π , the projections $U_i(\mathbf{x}) = x_i$, functions in φ , and the following operators:

1. *Composition:* if a p -ary function f and a function g with p components belong to $\mathcal{L} + \varphi$, then $h = f \circ g$, defined as $h(\mathbf{x}) = f(g(\mathbf{x}))$, belongs to $\mathcal{L} + \varphi$.
2. *Linear integration:* if f and g are in $\mathcal{L} + \varphi$ then the function h satisfying the initial condition $h(\mathbf{x}, 0) = f(\mathbf{x})$ and the differential equation

$$\partial_y h(\mathbf{x}, y) = g(\mathbf{x}, y) h(\mathbf{x}, y)$$

belongs to $\mathcal{L} + \varphi$. If h is vector-valued with n components, then f has the same dimension and $g(\mathbf{x}, y)$ is an $n \times n$ matrix whose components belong to $\mathcal{L} + \varphi$. As shorthand, we will write $h = f + \int gh dy$.

Several notes on this definition are in order. First, note that linear integration can only solve differential equations $\partial_y h = gh$ where the right-hand side is linear in h , rather than the arbitrary dependence $\partial_y h = g(h)$ of which the GPAC is capable. Secondly, using the same trick as in Section 3 we can expand our set of variables, and so solve non-homogeneous linear differential equations of the form

$$\partial_y h(\mathbf{x}, y) = g_1(\mathbf{x}, y) h(\mathbf{x}, y) + g_2(\mathbf{x}, y).$$

Finally, the reader will note that we are including π as a fundamental constant. The reason for this will become clear in Proposition 15. Unfortunately, we have not found a way to derive π using this restricted class of differential equations. Perhaps the reader can find a way to do this.

Figures 1 and 2 show schematically the initial functions and the operations that define $\mathcal{L} + \varphi$. In those figures each box represents a function. Nested boxes in figure 2 mean that the outer box is in $\mathcal{L} + \varphi$ if the inner boxes are.

Proposition 10. *All functions in $\mathcal{L} + \varphi$ are defined everywhere.*

Proof. The proof is by induction on the definition of any $\mathcal{L} + \varphi$ function. Basic functions are defined everywhere by definition, composition of total functions is total, and Proposition 8 shows that the solution of a linear differential equation is defined everywhere if the entries of the matrix in the right side of equation 3 are continuous. Furthermore, Equation 4 gives a bound on the solution of the linear differential equation. \square

Let us look at a few examples. Addition, as a function of two variables, is in \mathcal{L} since $x + 0 = U_1(x) = x$ and $\partial_y(x + y) = 1$. Similarly, multiplication can be defined as $x0 = 0$ and $\partial_y(xy) = x$. Exponentiation $\exp(x) = e^x$ is in \mathcal{L} since it can be defined as $\exp(0) = 1$ and $\partial_x \exp(x) = \exp(x)$. We can define $\exp^{[2]}(x) = e^{e^x}$ by using either composition or linear integration, with $\exp^{[2]}(0) = \exp(1) = e$ and $\partial_x \exp^{[2]}(x) = \exp(x) \exp^{[2]}(x)$. (Note that we are now using e rather than 2 as our base for exponentiation.)

Thus the iterated exponential $\exp^{[m]}$ is in \mathcal{L} for any fixed m . However, the function $\exp^{[x]}(0)$, where the number of iterations is a variable, is not in \mathcal{L} or in \mathcal{G} . We prove this in [CMC99], and use it to show that Shannon's GPAC is not closed under iteration. However, if \mathcal{G} is extended with a function which checks inequalities in a differentiable way, then the resulting class $\mathcal{G} + \theta_k$ is closed under iteration, where θ_k is defined as follows.

Let $\theta_k(x) = x^k \theta(x)$ where $\theta(x)$ is the Heaviside step function

$$\theta(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

Each $\theta_k(x)$ can be interpreted as a function which checks inequalities such as $x \geq 0$ in a $(k - 1)$ -times differentiable way. It was also shown in [CMC99] that allowing those functions is equivalent to relaxing slightly the definition of GPAC by considering a two-point boundary value problem instead of just an initial condition.

In this section we will consider $\mathcal{L} + \theta_k$ and we will prove that for any $k > 2$ this class is an analog characterization of the elementary functions. We will start by showing that all functions in $\mathcal{L} + \theta_k$ have growth bounded by a finitely iterated exponential, $\exp^{[m]}$ for some m . This is analogous to the bound on elementary functions in Proposition 1.

Proposition 11. *Let h be a function in $\mathcal{L} + \theta_k$ of arity m . Then there is a constant d and constants A, B, C, D such that, for all $\mathbf{x} \in \mathbb{R}^m$,*

$$\begin{aligned} \|h(\mathbf{x})\| &\leq A \exp^{[d]}(B\|\mathbf{x}\|) \\ \|\partial_{x_i} h(\mathbf{x})\| &\leq C \exp^{[d]}(D\|\mathbf{x}\|) \text{ for all } i = 1, \dots, m \end{aligned}$$

where $\|\mathbf{x}\| = \max_i |x_i|$. We will call d the degree of h or $\deg h$.

Proof. We proceed by induction. The constants 0, 1, -1 , π , the projections, and θ_k all have degree 1. For composition, it is easy to see that $\deg(f \circ g) \leq \deg f + \deg g$. For linear integration Proposition 8 shows that $\deg h \leq \max(\deg f, \deg g + 1)$. As a matter of fact, we easily exceed sums and products by adjusting the constants A, B, C and D , so $\deg(f + g)$ and $\deg(fg)$ are both at most $\max(\deg f, \deg g)$.

To bound the derivative, we will inductively assume that $\partial_x h$ is bounded by the same degree as h is. Then for composition, if $h = f \circ g$ the chain rule gives $\partial_x h = (\partial_g f \circ g)(\partial_x g)$, so $\deg h \leq \max(\deg(\partial_g f \circ g), \deg \partial_x g) \leq \deg f + \deg g$ as before. For linear integration, if $h = f + \int gh dy$, then $\partial_y h = gh$ is bounded with the same degree as h . Taking the derivative with respect to one of the other variables gives $\partial_x h = \partial_x f + \int_0^y (g \partial_x h + h \partial_x g) dy$, which is again bounded by $\max(\deg f, \deg g + 1)$ by Proposition 8.

The above induction is easy to generalize to the case of several variables rather than one, where the degree of a vector-valued function is the maximum of the degrees of its components. \square

Propositions 1 and 11 establish the same kind of bounds for \mathcal{E} and $\mathcal{L} + \theta_k$. But the relation between those two classes can be shown to be much tighter: namely, all functions in $\mathcal{L} + \theta_k$ can be approximated by elementary functions, and all elementary functions are contained in $\mathcal{L} + \theta_k$.

Since \mathcal{E} is defined over the natural numbers and $\mathcal{L} + \theta_k$ is defined over the reals, we first need to set some conventions. We say that a function over the reals is elementary if it fulfills Grzegorzczuk and Lacombe's definition of computable continuous real function [Grz55, Grz57, Lac55, PER89] and if the corresponding functional is elementary. First, we write $S \rightsquigarrow a$ if an integer sequence $S = \{S_i\}$ satisfies $|S_i/(i+1) - a| < 1/(i+1)$ for all $i \in \mathbb{N}$. Then a continuous real function $f : \mathbb{R} \rightarrow \mathbb{R}$ is *elementarily computable* if there is an elementary functional $\Phi : \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \rightarrow \mathbb{N}$ which, for all $a \in \mathbb{R}$ and for all sequences $\phi \rightsquigarrow a$, satisfies $\Phi(\phi) \rightsquigarrow f(a)$. By $\Phi(\phi)_i$ we denote the value of Φ on i , when Φ accesses the sequence ϕ , and $\Phi(\phi) = \{\Phi(\phi)_i\}$ denotes the sequence $\{\Phi(\phi)_0, \Phi(\phi)_1, \Phi(\phi)_2, \dots\}$. Note that, since $S_i/(i+1)$ can be a non-integer rational, the definition above is such that sequences that are defined over \mathbb{N} can converge to non-integer numbers. The definition for vector-valued functions and functions of several variables is similar.

Conversely, we say that $\mathcal{L} + \theta_k$ contains a function $f : \mathbb{N} \rightarrow \mathbb{N}$ if $\mathcal{L} + \theta_k$ contains some extension of f to the reals, and similarly for functions of several variables. These two conventions allow us to compare analog and digital complexity classes.

Proposition 12. *If f belongs to $\mathcal{L} + \theta_k$ for any $k > 2$ then f is elementarily computable.*

Proof. Once again, the proof will be done by induction. The constants 0, 1 and -1 are clearly elementarily computable (e.c.). If f is e.c. then $U_i(f)$ is simply one of its components and so is as well, and $\theta_k(x)$ is e.c. since $x \mp y$ and polynomials are in \mathcal{E} .

For simplicity, we prove that composition of e.c. functions is also e.c. just for real functions of one variable. The proof is similar for the general case. If g is e.c. then there is a functional Φ_g such that $\Phi_g(\phi) \rightsquigarrow g(a)$ for any sequence $\phi \rightsquigarrow a$, and if f is e.c. then there is a functional Φ_f such that $\Phi_f(\varphi) \rightsquigarrow f(a)$ for any sequence $\varphi \rightsquigarrow a$. Setting $\varphi = \Phi_g(\phi)$ we obtain $|\Phi_f(\Phi_g(\phi))_j/(j+1) - f(g(a))| < 1/(j+1)$ for all j , so $\Phi_f(\Phi_g(\phi)) \rightsquigarrow f(g(a))$. Since the composition of two elementary functionals is elementary, we are done (see [Grz55] for more details).

Finally, we have to show that if f and g are e.c. then h such that $h(\mathbf{x}, 0) = f(\mathbf{x})$ and $\partial_y h(\mathbf{x}, y) = g(\mathbf{x}, y)h(\mathbf{x}, y)$ is also. This means that we have to show that there is an elementary functional Φ such that $\Phi(\phi) \rightsquigarrow h(\mathbf{x}, y)$ for all sequences $\phi = (\phi_{\mathbf{x}}, \phi) \rightsquigarrow (\mathbf{x}, y)$.

Let us suppose that $h \in \mathcal{L} + \theta_k$ is twice continuously differentiable, which is guaranteed if $k > 2$. Fixing \mathbf{x} and expanding h we obtain

$$h(\tau_{i+1}) = h(\tau_i) + (\tau_{i+1} - \tau_i) h(\tau_i) g(\tau_i) + (\tau_{i+1} - \tau_i)^2 h''(\xi)/2$$

for some ξ where $\tau_i < \xi < \tau_{i+1}$. Since g is e.c. there is an elementary functional Φ_g such that $\Phi_g(\phi) \rightsquigarrow g(\tau_i)$ for all $\phi \rightsquigarrow \tau_i$. To obtain an estimate of the value of g on τ_i , we set $\gamma_i = \Phi_g(\phi)_n/(n+1)$. The accuracy of this estimate depends on n since $|\gamma_i - g(\tau_i)| < 1/(n+1)$.

The discrete approximation of h is done by Euler's method, and we will show that we can make the error sufficiently small with only an elementary number of discrete steps. We define a function ψ by

$$\psi_0 = \Phi_f(\phi_{\mathbf{x}})_m/(m+1) \quad \text{and} \quad \psi_{i+1} = \psi_i + \lambda \psi_i \gamma_i \quad (5)$$

where $\lambda = \tau_{i+1} - \tau_i$ is the step size of the discretization and m is an elementary function we will define below of the number of steps of the numerical approximation. Φ will denote the elementary functional that estimates $h(\mathbf{x}, y)$. We define Φ , for each fixed \mathbf{x} and any sequence $\phi \rightsquigarrow y$, by $\Phi(\phi)_l = (l+1)\psi_{N(l)}$, where $N(l)$ is a suitably increasing elementary function and where $\psi_{N(l)}$ is obtained using a discretization step $\lambda_l = \phi_l/N(l)$ in (5). This means that Φ adjusts n , m and λ as a function of l such that $\Phi(\phi)_l/(l+1) = \psi_{N(l)}$ satisfies $|\Phi(\phi)_l/(l+1) - h(\mathbf{x}, y)| < 1/(l+1)$. To prove that this can be done in elementary time, we first need to set a bound on

$$h''(\mathbf{x}, t) = \partial_t h'(\mathbf{x}, t) = \partial_t (g(\mathbf{x}, t) h(\mathbf{x}, t)) = h(\mathbf{x}, t) \partial_t g(\mathbf{x}, t) + g^2(\mathbf{x}, t) h(\mathbf{x}, t).$$

From Proposition 11 we know that $\|h''(\mathbf{x}, t)\|$ is then bounded by $A \exp^{[d]}(B\|\mathbf{x}, t\|)$ for some A and B , where $d \leq \max(\deg f, \deg g)$. We will call this bound β , and in the remainder of the proof we will fix \mathbf{x} and write $h(t)$ and $\beta(t)$ for $h(\mathbf{x}, t)$ and $\beta(\mathbf{x}, t)$ respectively.

The discretization error is

$$\epsilon_i = \psi_i - h(\tau_i) = \psi_i + \lambda \psi_i \gamma_i - h(\tau_i) - \lambda h(\tau_i) g(\tau_i) - \lambda^2 h''(\xi)/2$$

and satisfies

$$|\epsilon_{i+1}| \leq |\epsilon_i| ((1 + \lambda g(\tau_i)) + \lambda \psi_i (\gamma_i - g(\tau_i)) + \lambda^2 \beta(\tau_{i+1})/2).$$

Furthermore, $\epsilon_0 = \psi_0 - h(0)$, and because f is e.c.,

$$|\epsilon_0| = |\Phi_f(\phi_{\mathbf{x}})_m/(m+1) - f(\mathbf{x})| < 1/(m+1)$$

where Φ_f is the elementary functional that computes f . A little tedious algebra shows then that

$$|\epsilon_{N(l)}| \leq \left(\frac{N(l)}{m+1} + \frac{y^2 \beta(y)}{N(l)} \right) \left(1 + \frac{y \beta(y)}{N(l)} \frac{n+2}{n+1} \right)^{N(l)} \leq \left(\frac{N(l)}{m+1} + \frac{y^2 \beta(y)}{N(l)} \right) e^{y \beta(y) \frac{n+2}{n+1}}.$$

Therefore, given β , which is elementary, it suffices to fix any n and set

$$N(l) = l y^2 \beta(y) e^{y \beta(y) \frac{n+2}{n+1}} \quad \text{and} \quad m = m(l) = l N(l) e^{y \beta(y)}$$

to guarantee that $|\epsilon_{N(l)}| = |\psi_{N(l)} - h(\mathbf{x}, y)| < 1/(l+1)$ as required. Since all those functions are elementary, Φ can be computed in elementary time and we're done. \square

As a corollary, any function in $\mathcal{L} + \theta_k$ that sends integers to integers is elementary on the integers:

Corollary 13. *If a function $f \in \mathcal{L} + \theta_k$ is an extension of a function $\tilde{f} : \mathbb{N} \rightarrow \mathbb{N}$, then \tilde{f} is elementary.*

Proof. Proposition 12 shows us how to successively approximate $f(x)$ to within an error ϵ in an amount of time elementary in $1/\epsilon$ and x . If f is an integer, we just have to approximate it to error less than $1/2$ to know its value exactly. \square

Next we will prove the converse of this, i.e. that $\mathcal{L} + \theta_k$ contains all elementary functions, or rather, extensions of them to the reals. We will first prove two lemmas.

Lemma 14. *For any $k > 0$, $\mathcal{L} + \theta_k$ contains \sin , \cos , \exp , the constant q for any rational q , and extensions to the reals of successor, addition, and cut-off subtraction.*

Proof. We can obtain any integer by repeatedly adding 1 or -1 . For rational constants, by repeatedly integrating 1 we can obtain the function $f(z) = z^k/k!$ and thus $f(1) = 1/k!$ for any k . We can multiply this by an integer to obtain any rational q .

We showed above that $\mathcal{L} + \theta_k$ includes addition, and the successor function is just addition by 1. For subtraction, we have $x - 0 = x$ and $\partial_y(x - y) = -1$.

For cut-off subtraction $x \dot{-} y$, we first define a function $s(z)$ such that $s(z) = 0$ when $z \leq 0$ and $s(z) = 1$ when $z \geq 1$, for all $z \in \mathbb{Z}$. This can be done in $\mathcal{L} + \theta_k$ by setting $s(0) = 0$ and $\partial_z s(z) = c_k \theta_k(z(1-z))$, where $c_k = 1/\int_0^1 z^k(1-z)^k dz$ is a rational constant depending on k . Then $x \dot{-} y = (x - y) s(x - y)$.

Finally, $h(t) = (\cos(t), \sin(t))$ is defined by

$$\begin{bmatrix} h'_1 \\ h'_2 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix},$$

with $h_1(0) = 1$ and $h_2(0) = 0$, and \exp was proved above. \square

We now show that $\mathcal{L} + \theta_k$ has the same closure properties as \mathcal{E} , namely the ability to form bounded sums and products.

Lemma 15. *Let f be a function on \mathbb{N} and let g be the function on \mathbb{N} defined from f by bounded sum or bounded product. If f has an extension to the reals in $\mathcal{L} + \theta_k$ then g does also.*

Proof. For simplicity, we give the proof for functions of one variable. We will abuse notation by identifying f and g with their extensions to the reals.

We first define a step function F which matches f on the integers, and whose values are constant on the interval $[j, j + 1/2]$ for integer j . F can be defined as $F(t) = f(s(t))$, where $s(t)$ is a function such that $s(0) = 0$ and $s'(t) = c_k \theta_k(-\sin 2\pi t)$. Here $c_k = 1/\int_0^{1/2} \sin^k 2\pi t dt$ is a constant depending only on k . Since c_k is rational for k even and a rational multiple of π for k odd, s is definable in $\mathcal{L} + \theta_k$. (Now our reasons for including π in the definition of $\mathcal{L} + \theta_k$ become clear.) Then $s(t) = j$, and $F(s(t)) = f(j)$, whenever $t \in [j, j + 1/2]$ for integer j .

The bounded sum of f is easily defined in $\mathcal{L} + \theta_k$ by linear integration. Simply write $g(0) = 0$ and $g'(t) = c_k F(t) \theta_k(\sin 2\pi t)$. Then $g(t) = \sum_{z < n} f(z)$ whenever $t \in [n - 1/2, n]$.

Defining the bounded product of f in $\mathcal{L} + \theta_k$ is more difficult. Let us first set some notation. Let f_j denote $f(j)$ for $j \in N$, which is also equal to $F(t)$ for $t \in [j, j + 1/2]$. Then $g_n = \prod_{j < n} f_j = f_0 \dots f_{n-1}$ is the bounded product we wish to define. The idea of the proof is to approximate the iteration $g_{j+1} = g_j f_j$ using synchronized clock functions as in [Bra95, Moo96, CMC99]. However, since the model we propose here only allows linear integration, the simulated functions cannot coincide exactly with the bounded product. Nevertheless, we can define a sufficiently close approximation because f and g have bounded growth by Proposition 11. Then since f and g have integer values, the accumulated error resulting from this approximation can be removed with a suitable continuous step function ϕ similar to s above. Specifically, if $\phi(0) = 0$ and $\phi'(t) = c_k \theta_k(-\cos 2\pi t)$, then $\phi(t) = j$ if $t \in [j - 1/4, j + 1/4]$ for all integer j , so ϕ returns the integer closest to t as long as the error is $1/4$ or less.

Now define a two-component function $\mathbf{y}(\tau, t)$ where $y_1(\tau, 0) = y_2(\tau, 0) = 1$ and

$$\begin{aligned} \partial_t y_1 &= (y_2 F(t) - y_1) c_k \theta_k(\sin 2\pi t) \beta(\tau) \\ \partial_t y_2 &= (y_1 - y_2) c_k \theta_k(-\sin 2\pi t) \beta(\tau) \end{aligned} \quad (6)$$

Then we claim that $g_n = \phi(y_1(n, n))$. Here $\beta(\tau)$ is an increasing function of τ . We will see that if β grows fast enough, then by setting $\tau = n$ we can make the approximation error $|y_1(n, n) - g_n|$ as small as we like, and then remove it with ϕ .

As in [Moo96, CMC99] the idea is that on alternate intervals we hold either y_1 or y_2 constant and update the other one. For integer j , it is easy to see that when $t \in [j, j + 1/2]$ the term $\theta_k(-\sin 2\pi t)$ holds y_2 constant, and y_1 moves toward $y_2 f(j)$. Quantitatively, solving (6) for $t \in [j, j + 1/2]$ gives $y_1(n, t) = y_2(n, j) f_j + (y_1(n, j) - y_2(n, j) f_j) \exp(-S(t) \beta(\tau))$ where $S(j) = 0$ and $S'(t) = c_k \sin^k 2\pi t$, so $S(j + 1/2) = 1$. Similarly, when $t \in [j + 1/2, j + 1]$, y_1 is held constant and y_2 moves toward y_1 . This gives us the recursion

$$\begin{aligned} y_1(n, j + 1) &= y_2(n, j) f_j + \epsilon (y_1(n, j) - y_2(n, j) f_j) \\ y_2(n, j + 1) &= y_1(n, j + 1) + \epsilon (y_2(n, j) - y_1(n, j + 1)), \end{aligned} \quad (7)$$

where $\epsilon = \exp(-\beta(\tau))$. Note that if ϵ is sufficiently small then $y_1(n, j + 1) \approx y_2(n, j) f_j$ and $y_2(n, j + 1) \approx y_1(n, j + 1)$ and, therefore, $y_1(n, j + 1) \approx y_1(n, j) f_j$ as intended.

Now let $\beta(\tau) = A \exp^{[m]}(B\tau)$ and $\tau = n$, where $f(n)$ and $g(n)$ are bound from above by $A \exp^{[m]}(Bn)$ as in Proposition 11. In the Appendix we show that

$$\begin{aligned} |y_1(n, n) - g_n| &\leq \epsilon 2^n (n + 1) \beta^{n+1}(n) \\ &= e^{-\exp^{[m]}(n)} 2^n (n + 1) (\exp^{[m]}(n))^{n+1} \\ &< 4e^2 / e^{2e} \approx 0.13 \end{aligned}$$

for all $m, n \geq 1$. Since this is less than $1/4$, we can round the value of y_1 to the nearest integer using ϕ , and we're done. \square

An interesting question is whether $\mathcal{L} + \theta_k$ is closed under bounded product for functions with real, rather than integer, values. Our conjecture is that it is not, but we have no proof of this.

Using the previous two lemmas it is easy now to prove that

Proposition 16. *If f is an elementary function, then $\mathcal{L} + \theta_k$ contains an extension of f to the reals.*

Proof. Basic elementary functions belong to $\mathcal{L} + \theta_k$ as proved in lemma 14. The set of integer functions possessing extensions in $\mathcal{L} + \theta_k$ is closed under composition by definition, and under bounded sum and bounded product by Proposition 15. Therefore, it contains all of \mathcal{E} . \square

Taken together, Propositions 12 and 16 show that the analog class $\mathcal{L} + \theta_k$ corresponds to the elementary functions in a natural way.

5 $\mathcal{G}_n + \theta_k$ and the Grzegorzcyk hierarchy

In this section we show that we can extend the results of the previous section to the higher levels of the Grzegorzcyk hierarchy, \mathcal{E}^n for $n \geq 3$. Let us define a hierarchy of recursive functions on the reals. Each level is denoted by $\mathcal{G}_n + \theta_k$ for some $n \geq 3$. The first level is $\mathcal{G}_3 + \theta_k = \mathcal{L} + \theta_k$, and each following level is defined either by adding a new basis function, or by allowing the system to solve non-linear differential equations, but only when the resulting function is bounded by the iteration of some function in the previous level.

Definition 17 (The hierarchy $\mathcal{G}_n + \theta_k$). Let $\mathcal{G}_3 + \theta_k = \mathcal{L} + \theta_k$ be the smallest class containing the constants 0, 1, -1 , and π , projections $U_i(\mathbf{x}) = x_i$ and θ_k , which is closed under composition and linear integration.

For $n \geq 3$, $\mathcal{G}_{n+1} + \theta_k$ is defined as the set containing $\mathcal{G}_n + \theta_k$ which is closed under composition, linear integration, and the following kind of *limited integration*. If $f, g \in \mathcal{G}_{n+1} + \theta_k$, then so is the solution $y(x, t)$ of the following first-order differential equation and initial conditions:

$$\begin{aligned} y(x, 0) &= f(x) \\ g(x, t, y, y') &= 0 \end{aligned}$$

but only on the conditions that the solution is unique and defined everywhere, and that both y and y' are bounded, $\|y(x)\|, \|y'(x)\| \leq G(|x|)$ where $G(t)$ is a monotone non-decreasing extension to the reals of the iteration $g^{[t]}(x)$ for some $g \in \mathcal{G}_n + \theta_k$ and some $x \in \mathbb{R}$. Functions of several variables \mathbf{x} or vector-valued functions \mathbf{y} are defined similarly.

We will now show that $\mathcal{G}_n + \theta_k$ corresponds to the n th level of the Grzegorzcyk hierarchy in the same way that $\mathcal{L} + \theta_k$ corresponds to the elementary functions. First, we will show in analogy to Proposition 6 that iteration carries us up the levels of the $\mathcal{G}_n + \theta_k$ hierarchy just as it does for the \mathcal{E}^n :

Proposition 18. *For any monotone non-decreasing $f \in \mathcal{G}_n + \theta_k$ there is a monotone non-decreasing extension of the iteration $F(x, t) = f^{[t]}(x)$ in $\mathcal{G}_{n+1} + \theta_k$.*

Proof. $F(x, t)$ is given by y_1 in the following system of equations, where the variable of integration is t and x is given in the initial conditions $y_1(0) = y_2(0) = x$:

$$\begin{aligned} y_1' |\cos \pi t|^{k+1} + 2\pi(y_1 - f(y_2)) \theta_k(\sin 2\pi t) &= 0 \\ y_2' |\sin \pi t|^{k+1} + 2\pi(y_2 - y_1) \theta_k(-\sin 2\pi t) &= 0 \end{aligned} \quad (8)$$

Note that $|x|^k$ can be defined in $\mathcal{L} + \theta_k$ as $|x|^k = \theta_k(x) + \theta_k(-x)$. Then $y_1(t) = f^{[t]}(x)$ belongs to $\mathcal{G}_{n+1} + \theta_k$ since it is a solution of a differential equation definable in $\mathcal{G}_n + \theta_k$ and it is bounded by the iteration of a function in $\mathcal{G}_n + \theta_k$, which is f itself. The dynamics are similar to Equation 6 for iterated multiplication, in which y_1 and y_2 are held constant for alternating intervals. The main difference is that the terms $|\cos \pi t|^{k+1}$ and $|\sin \pi t|^{k+1}$ on the left ensure that y_1 converges exactly to $f(y_2)$, and y_2 exactly to y_1 , by the end of the interval $[n, n + 1]$. Details are given in [CMC99]. \square

Now define a series of functions $\exp_2(x) = e^x$ and $\exp_{n+1}(x) = \exp_n^{[x]}(1)$ for $x \in \mathbb{N}$. Since $\exp_2 \in \mathcal{L} + \theta_k$, Proposition 18 shows that $\mathcal{G}_n + \theta_k$ contains extensions to the reals of \exp_{n-1} for all $n \geq 3$. Similarly, recall the definition of the functions E_n from Definition 2. Since E_2 is elementary, an extension of it belongs to $\mathcal{L} + \theta_k$ by Proposition 16, and since E_{n+1} is defined from E_n by iteration, $\mathcal{G}_n + \theta_k$ also contains extensions to the reals of E_{n-1} for all $n \geq 3$. For simplicity, we will also use the notation $\exp_n(x)$ and $E_n(x)$ for monotone extensions of \exp_n and E_n to $x \in \mathbb{R}$.

In fact, finite compositions of \exp_{n-1} and E_{n-1} put an upper bound on the growth of functions in $\mathcal{G}_n + \theta_k$, so in analogy to Proposition 11 we have the following:

Proposition 19. *Let h be a function of arity m in $\mathcal{G}_n + \theta_k$ for any $n \geq 3$. Then there is a constant d and constants A, B, C, D such that, for all $x \in \mathbb{R}^m$,*

$$\begin{aligned} \|h(\mathbf{x})\| &\leq A \exp_{n-1}^{[d]}(B\|\mathbf{x}\|) \\ \|\partial_{x_i} h(\mathbf{x})\| &\leq C \exp_{n-1}^{[d]}(D\|\mathbf{x}\|) \text{ for all } i = 1, \dots, m \end{aligned}$$

where $\|\mathbf{x}\| = \max_i |x_i|$. Moreover, the same is true (with different constants) if \exp_{n-1} is replaced by E_{n-1} .

Proof. The proof is by an easy induction. Let us assume it is true for $\mathcal{G}_n + \theta_k$, with Proposition 11 giving the base case $n = 3$. If we again call d the degree, then for any $n \geq 3$ the degree of a sum or product of two functions is at most the maximum of their degrees, for composition we have $\deg(f \circ g) \leq \deg f + \deg g$, and for linear integration $h = f + \int gh dy$ we have $\deg h \leq \max(\deg f, \deg g + 1)$, all as before. Thus composition and linear integration inductively satisfy the bounds on y and y' , and our definition of limited integration of non-linear differential equations enforces these bounds as well. The proof for E_n is similar. \square

Now, as before, to compare an analog class to a digital one we will say a real function is computable in \mathcal{E}^n if it can be approximated by a series of rationals with a functional in \mathcal{E}^n , and we will say that an integer function is in an analog class if some extension of it to the reals is. Then:

Proposition 20. *The following correspondences exist between $\mathcal{G}_n + \theta_k$ and the levels of the Grzegorzcyk hierarchy, \mathcal{E}^n for all $n \geq 3$:*

1. *Any function in $\mathcal{G}_n + \theta_k$ is computable in \mathcal{E}^n .*
2. *If $f \in \mathcal{G}_n + \theta_k$ is an extension to the reals of some \tilde{f} on \mathbb{N} , then $\tilde{f} \in \mathcal{E}^n$.*
3. *Conversely, if $f \in \mathcal{E}^n$ then some extension of it to the reals is in $\mathcal{G}_n + \theta_k$.*

Proof. To prove that functions in $\mathcal{G}_n + \theta_k$ can be computed with functionals in \mathcal{E}^n , we follow the proof of Proposition 12. However, when we use Euler's method for numerical integration, we now apply the bound of Proposition 19 and set $m(l)$ and $N(l)$ to grow as $AE_{n-1}^{[d]}(Bl)$ for a certain d . Since this is in \mathcal{E}^n , so is the functional Φ . As in Corollary 13, if f takes integer values on the integers we just have to approximate it to within an error less than $1/2$.

Conversely, the remarks above show that $\mathcal{G}_n + \theta_k$ contains an extension to the reals of E_{n-1} , and Lemma 14 shows that it contains the other initial functions of \mathcal{E}^n as well. Furthermore, it is closed under bounded sum and bounded product for integer-valued functions. The proof of Lemma 15 proceeds as before, except that using Proposition 19 again, $\beta(\tau)$ is now $AE_{n-1}^{[m]}(B\tau)$. Since an extension of this to the reals can be defined in $\mathcal{G}_n + \theta_k$, so can the linear differential equation (6). Finally, for $n \geq 3$ we can replace limited recursion in the definition of \mathcal{E}^n with bounded sum and bounded product according to Proposition 3. \square

Notice that Proposition 20 implies that $\cup_n \mathcal{G}_n + \theta_k$ includes all primitive recursive functions since, as mentioned in Proposition 5, $\cup_n \mathcal{E}^n$ is the class \mathcal{PR} of primitive recursive functions.

Finally, we note that if we had defined $\mathcal{G}_n + \theta_k$ by simply including real extensions of E_{n-1} or \exp_{n-1} as basis functions, instead of adding the limited integration operator, Proposition 20 would still hold. While this definition produces a smaller set of functions on the reals, it produces the same set of integer functions with real extensions.

6 Conclusion

We have defined a new version of Shannon's General Purpose Analog Computer \mathcal{G} in which the integration operator is restricted in a natural way — to solving linear differential equations. When we add the ability to measure inequalities in a differentiable way, the resulting system $\mathcal{L} + \theta_k$ corresponds exactly to the elementary functions \mathcal{E} . Furthermore, we have defined a hierarchy of analog classes $\mathcal{G}_n + \theta_k$ using an integration operator whose solutions have restricted growth, and we have shown that this hierarchy corresponds, level by level, to the Grzegorzcyk hierarchy \mathcal{E}^n for $n \geq 3$. When combined with the earlier result [CMC99] that $\mathcal{G} + \theta_k$ contains the primitive recursive functions, this suggests that subclasses of the recursive functions correspond nicely to natural subclasses of analog computers.

Several open questions suggest themselves:

1. Can we do without π in the definition of $\mathcal{L} + \theta_k$? Note that we do not need to include it in the definition of $\mathcal{G}_n + \theta_k$ for $n > 3$, since we can define $f(x) = 1/(x^2 + 1)$ from

limited integration as $f(0) = 1$ and $f' + 2xf^2 = 0$, then define $g(x) = \tan^{-1} x$ from linear integration as $g(0) = 0$ and $g' = f$, and finally set $\pi = 4 \tan^{-1} 1$. However, we have been unable to find a way to define π from linear integration alone.

2. Is $\mathcal{L} + \theta_k$ closed under bounded product for real-valued functions, and not just integer-valued ones? We think this is unlikely, since it would require some form of iteration like that in Equation 8 where y_1 and y_2 converge to the desired values exactly. We see no way to do this without highly non-linear terms. If $\mathcal{L} + \theta_k$ is not closed under real-valued bounded products then we could ask what class would result from that additional operation. While the set of integer functions which have real extensions in the class would remain the same, the set of functions on the reals would be larger.
3. By adding to our basis a function that grows faster than any primitive recursive function, such as the Ackermann function, we can obtain transfinite levels of the extended Grzegorzczuk hierarchy [Ros84]. It would be interesting to find natural analog operators that can generate such functions through diagonalization.
4. How robust are these systems in the presence of noise? Since it is based on linear differential equations, $\mathcal{L} + \theta_k$ may exhibit a fair amount of robustness to perturbations. We hope to quantify this, and explore whether this makes these models more robust than other continuous-time analog models, which are highly non-linear.

It is interesting that linear integration alone gives extensions to the reals of all elementary functions, since these are all the functions that can be computed by any practically conceivable digital device. In terms of dynamical systems, $\mathcal{L} + \theta_k$ corresponds to cascades of finite depth, each level of which depends linearly on its own variables and the output of the level before it. We find it surprising that such systems, as opposed to highly non-linear ones, have so much computational power.

Finally, we note that while including θ_k as an oracle makes these functions non-analytic, by increasing k they can be made as smooth as we like. Therefore, we claim that these are acceptable models of real physical phenomena, and may be more realistic in certain cases than either discrete or hybrid systems.

Acknowledgements. We thank Jean-Sylvestre Gakwaya, Norman Danner, Robert Israel, Kathleen Merrill, Spootie Moore, Bernard Moret, and Molly Rose for helpful discussions. This work was partially supported by FCT PRAXIS XXI/BD/18304/98 and FLAD 754/98.

Appendix: completion of the proof of Proposition 15. For simplicity, we will set the constants in the bound on f and g to $A = 2$ and $B = 1$ since $A \exp^{[m]}(B\tau)$ can always be bounded by $2 \exp^{[m']}(B\tau)$ with m' large enough. Also to simplify the notation, we will denote $y_1(n, j)$ and $y_2(n, j)$ by $y_1(j)$ and $y_2(j)$, respectively. We will prove that $|y_1(n) - g_n| \leq 4e^2/e^{2e}$ for all $m, n \leq 1$. We will proceed by induction on j for $j \leq n$. Recall that $y_1(0) = y_2(0) = 1$, $\epsilon = e^{-\beta}$ and $\beta = \exp^{[m]}(n)$. If $n = 0$, Equation 7 shows that $|y_1(1) - f_0| \leq \epsilon(1 - f_0) \leq \epsilon\beta$ and that $|y_2(1) - y_1(1)| \leq \epsilon|y_2(0) - y_1(1)| \leq \epsilon|1 - \epsilon\beta - f_0| \leq \epsilon\beta$. We will now show that if

$$|y_1(j) - g_j| \leq 2^j(j+1)\epsilon\beta^{j+1} \tag{9}$$

and

$$|y_2(j) - y_1(j)| \leq 2^j(j+1)\epsilon\beta^{j+1} \tag{10}$$

then

$$|y_1(j+1) - g_{j+1}| \leq 2^{j+1}(j+2)\epsilon\beta^{j+2} \quad (11)$$

and

$$|y_2(j+1) - y_1(j+1)| \leq 2^{j+1}(j+2)\epsilon\beta^{j+2}, \quad (12)$$

for all $j \leq n-1$, when $f_j, g_j \leq \beta$.

First, note that from (9), (10) and the triangle inequality, we obtain

$$|y_2(j) - g_j| \leq 2^{j+1}(j+1)\epsilon\beta^{j+1}. \quad (13)$$

To prove (11) from (9) and (10) we use the recursion of Equation 7 and the bounds on f_j and g_j . From (7) we have

$$y_1(j+1) = y_2(j)f_j + \epsilon(y_1(j) - y_2(j)f_{j+1}). \quad (14)$$

From (9) we can write $y_1(j) = g_j + \xi$, where $|\xi| \leq 2^j(j+1)\epsilon\beta^{j+1}$ and, from (13), $y_2(j) = g_j + \zeta$, where $|\zeta| \leq 2^{j+1}(j+1)\epsilon\beta^{j+1}$. Then (14) can be rewritten as

$$y_1(j+1) = (g_j + \xi)f_j + \epsilon(g_j + \xi - (g_j + \zeta)f_{j+1}),$$

which is

$$y_1(j+1) - g_j f_j = \xi f_j + \epsilon g_j + \epsilon \xi - \epsilon g_j f_{j+1} - \epsilon \zeta f_{j+1}.$$

Therefore,

$$\begin{aligned} |y_1(j+1) - g_{j+1}| &\leq 2^j(j+1)\epsilon\beta^{j+1}\beta + \epsilon(\beta + 2^j(j+1)\epsilon\beta^{j+1} + \beta^2 + 2^{j+1}(j+1)\epsilon\beta^{j+1}\beta) \\ &= 2^j(j+1)\epsilon\beta^{j+2} + \epsilon\beta + 2^j(j+1)\epsilon^2\beta^{j+1} + \epsilon\beta^2 + 2^{j+1}(j+1)\epsilon^2\beta^{j+2}. \end{aligned}$$

Since ϵ is small and β is large then the first term dominates the others and, consequently, we have $|y_1(j+1) - g_{j+1}| \leq 2^{j+1}(j+2)\epsilon\beta^{j+2}$, as claimed. The proof that (9) and (10) imply (12) is similar.

Finally, we briefly show that $2^n(n+1)\epsilon\beta^{n+1}$, which is always positive, is smaller than $4e^2/e^{2e}$ for $n, m \geq 1$. When $m = n = 1$ we obtain the above value. It is easy to see that

$$\frac{2^n(n+1)[\exp^{[m]}(n)]^{n+1}}{e^{2\exp^{[m]}(n)}}$$

decreases when m and n increase. In fact,

$$\frac{2^n(n+1)[\exp^{[m]}(n)]^{n+1}}{e^{2\exp^{[m]}(n)}} \leq \frac{e^{2n+(n+1)\exp^{[m-1]}(n)}}{e^{2\exp^{[m]}(n)}},$$

and the upper exponent grows slower than the lower exponent. \square

References

- [Arn96] V. I. Arnold. *Equations Différentielles Ordinaires*. Editions Mir, 5 ème edition, 1996.
- [Bou99] O. Bournez. Achilles and the tortoise climbing up the hyper-arithmetical hierarchy. *Theoretical Computer Science*, 210(1):21–71, 1999.

- [Bow96] M.D. Bowles. U. S. technological enthusiasm and the British technological skepticism in the age of the analog brain. *IEEE Annals of the History of Computing*, 18(4):5–15, 1996.
- [Bra95] M. S. Branicky. Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theoretical Computer Science*, 138(1), 1995.
- [BSS89] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bull. Amer. Math. Soc.*, 21:1–46, 1989.
- [CMC99] M.L. Campagnolo, C. Moore, and J.F. Costa. Iteration, inequalities, and differentiability in analog computers. Working paper 99–07–043, Santa Fe Institute, July 1999.
- [Cut80] N. J. Cutland. *Computability: an introduction to recursive function theory*. Cambridge University Press, 1980.
- [Grz53] A. Grzegorzcyk. Some classes of recursive functions. *Rosprawy Mate.*, IV, 1953.
- [Grz55] A. Grzegorzcyk. Computable functionals. *Fund. Math.*, 42:168–202, 1955.
- [Grz57] A. Grzegorzcyk. On the definition of computable real continuous functions. *Fund. Math.*, 44:61–71, 1957.
- [Har82] P. Hartman. *Ordinary Differential Equations*. Birkhauser, 2nd edition, 1982.
- [HJ85] R. A. Horn and C. J. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
- [Kal43] L. Kalmar. Egyzzerü példa eldönthetetlen aritmetikai problémára. *Mate és Fizikai Lapok*, 50:1–23, 1943.
- [Lac55] D. Lacombe. Extension de la notion de fonction récursive aux fonctions d’une ou plusieurs variables réelles I. *C. R. Acad. Sci. Paris*, 240:2478–2480, 1955.
- [Mee93] K. Meer. Real number models under various sets of operations. *Journal of Complexity*, 9:366–372, 1993.
- [Moo90] C. Moore. Unpredictability and undecidability in dynamical systems. *Physical Review Letters*, 64:2354–2357, 1990.
- [Moo96] C. Moore. Recursion theory on the reals and continuous-time computation. *Theoretical Computer Science*, 162:23–44, 1996.
- [Moo98] C. Moore. Dynamical recognizers: real-time language recognition by analog computers. *Theoretical Computer Science*, 201:99–136, 1998.
- [Odi89] P. Odifreddi. *Classical Recursion Theory*. Elsevier, 1989.
- [Orp97a] P. Orponen. On the computational power of continuous time neural networks. In *Proc. SOFSEM’97, the 24th Seminar on Current Trends in Theory and Practice of Informatics*, Lecture Notes in Computer Science, pages 86–103. Springer-Verlag, 1997.
- [Orp97b] P. Orponen. A survey of continuous-time computation theory. In D.-Z. Du and K.-I Ko, editors, *Advances in Algorithms, Languages, and Complexity*, pages 209–224. Kluwer Academic Publishers, Dordrecht, 1997.
- [PE74] M. B. Pour-El. Abstract computability and its relation to the general purpose analog computer. *Transactions of A.M.S.*, 199:1–28, 1974.
- [PER89] M. B. Pour-El and J. I. Richards. *Computability in Analysis and Physics*. Springer-Verlag, 1989.
- [Ros84] H. E. Rose. *Subrecursion: functions and hierarchies*. Clarendon Press, 1984.
- [SF98] H. T. Siegelmann and S. Fishman. Analog computation with dynamical systems. *Physica D*, 120:214–235, 1998.
- [Sha41] C. Shannon. Mathematical theory of the differential analyser. *J. Math. Phys. MIT*, 20:337–354, 1941.
- [Sie98] H. T. Siegelmann. *Neural Networks and Analog Computation: Beyond the Turing Limit*. Birkhauser, 1998.
- [Tho76] W. Thomson. On an instrument for calculating the integral of the product of two given functions. *Proc. Royal Society of London*, 24:266–268, 1876.