

Characterizing Computable Analysis with Differential Equations (*Preprint from CCA 2008*)

Manuel L. Campagnolo
D.M./I.S.A., Technical University of Lisbon
and SQIG - Instituto de Telecomunicações
Lisbon, Portugal
mlc@math.isa.utl.pt

Kerry Ojakian
Department of Mathematics
SQIG - Instituto de Telecomunicações and IST, Portugal
Lisbon, Portugal
ojakian@math.ist.utl.pt

August 2008

Abstract

The functions of Computable Analysis are defined by enhancing the capacities of normal Turing Machines to deal with real number inputs. We consider characterizations of these functions using function algebras, known as Real Recursive Functions. Bournez and Hainry 2006 [5] used a function algebra to characterize the twice continuously differentiable functions of Computable Analysis, restricted to certain compact domains. In a similar model, Shannon's *General Purpose Analog Computer*, Bournez et. al. 2007 [3] also characterize the functions of Computable Analysis. We combine the results of [5] and Graça et. al. [13], to show that a different function algebra also yields Computable Analysis. We believe that our function algebra is an improvement due to its simple definition and because the operations in our algebra are less obviously designed to mimic the operations in the usual definition of the recursive functions using the primitive recursion and minimization operators.

1 Introduction

Computable Analysis is a well accepted paradigm of computability for real numbers and real functions. Nonetheless, other approaches to computation over the reals have been proposed. Among these there are models that evolve step by step, like BSS-machines [1] or real random access machines, and continuous-time models like Shannon's *General Purpose Analog Computer* (GPAC) [19] [12], continuous neural networks [15] or Moore's real recursive functions [16] (for an up-to-date review of continuous-time models see [2]). Recently, there has been a quest for characterizations of computability of real functions (in this paper, "computable" with no additional reference will always refer to Computable Analysis) based on alternative models, namely on those with a continuous state space that evolve in continuous time. One motivation for this research has been to

understand the computational power of continuous-time systems, and search for an analog of the Turing-Church thesis in the context of real computation.

The first true model of a universal continuous time machine was proposed by Shannon [19]. However, this has been proved to be weaker than Computable Analysis since computable functions like Euler's Γ or Riemann's ζ are not "GPAC generable" [17] [18]. Note that the computability of all GPAC generable functions follows from the results in [13], and the fact that they correspond precisely to the solutions of polynomial differential equations as shown in [12].

To improve the upper bounds on Shannon's GPAC a new notion of limit computation with polynomial differential equations (called GPAC-computability) was proposed in [11] and explored further in [3]. In that paper, it is shown that the real computable functions are precisely the GPAC-computable functions as long as we restrict ourselves to compact domains. Although GPAC-computability relies on a simple dynamics, the parameters of the differential equations can be arbitrary computable reals and the limit operator applies to functions with an arbitrary slow convergence rate (see [8] for some remarks on that model).

A more general approach to computability over the reals with differential equations was proposed in [16]; it is called *Real Recursive Functions*. It is a recursion theory on the reals and provides algebraic characterizations of classes of real functions. The general theory contains many non-computable functions, resulting essentially from the inclusion of a powerful zero finding operator over the real line. Restricting the theory, it is possible to define classes that extend to the reals certain classes of discrete functions like the primitive recursive or the elementary computable functions [6] [7].

Instead of asking which functions over the naturals have extensions in the reals, Bournez and Hainry proposed classes of real recursive functions that correspond precisely to classes of real computable functions. They describe analog characterizations for the real elementary computable functions [4] and for the functions of Computable Analysis, all restricted to \mathcal{C}^2 functions on compact domains [5]. This last characterization contains a set of basic functions and is closed under composition, linear integration, a limit operation and a root finding operation. In this paper we propose a class of real recursive functions which is simpler in some respects, avoiding the use of a root-finding operation, which has the disadvantage of directly mimicking the operation of minimization over the naturals. This is accomplished by strengthening the operation of integration, removing the linearity restriction. A significant point is then to show that this class is not too strong. To accomplish this, we use the results of Graça et. al. [13] to show that even this general integration preserves computability.

2 Formulating the Main Result

We now provide the basic definitions and state the result, leaving the proof for the next section. We start by defining a flexible set of computable functions over the reals, \mathbb{R} , placing no restriction on the domains of the functions (see [14] or [20] for details on Computable Analysis and Type-2 computability). Unless otherwise stated, for us, a function will always have some domain $E \subseteq \mathbb{R}^k$, for some positive integer k , and codomain \mathbb{R} .

Definition 2.1 *We say a function f is in $\mathbf{C}(\mathbb{R})$ iff there is a Type-2 Turing Machine M such that for any \bar{x} in the domain of f , M calculates $f(\bar{x})$ on input \bar{x} ; for \bar{x} not in the domain of f , we ask nothing of M .*

Though the above definition is convenient, useful theorems will consider restrictions of the domains.

Definition 2.2 Suppose \mathcal{F} is a set of functions. By its **compact restriction**, denoted $[\mathcal{F}]$, we mean the following set of functions: Consider any function $f(x_1, \dots, x_k)$ in \mathcal{F} and any product of intervals $([a_1, b_1] \times \dots \times [a_k, b_k])$, where a_i, b_i are rationals. If the domain of f includes this product, then the restriction of f to this product is in $[\mathcal{F}]$.

We will basically consider the set $[\mathbf{C}(\mathbb{R})] \cap \mathcal{C}^2$, where for an integer $k \geq 1$, we let \mathcal{C}^k refer to the set of (possibly partial) functions which are k times continuously differentiable with respect to any variable on their domain of definition.

We now turn our attention to function algebras. We use the term **operation** to refer to a function whose inputs and outputs are real functions.

Definition 2.3 Suppose \mathcal{B} is a set of functions (called **basic functions**), and \mathcal{O} is a set of operations. Then $FA[\mathcal{B}; \mathcal{O}]$ is called a **function algebra**, and it denotes the smallest set of functions containing \mathcal{B} and closed under the operations in \mathcal{O} . For ease of readability, we often list the elements of \mathcal{B} or \mathcal{O} simply as a list separated by commas.

Some of the most important operations will be defined using differential equations. In general an **initial value problem (IVP)** is given by a system of equations like the following:

$$\begin{aligned} h_1(\bar{x}, 0) &= f_1(\bar{x}) \\ &\vdots \\ h_k(\bar{x}, 0) &= f_k(\bar{x}) \\ \\ \frac{\partial}{\partial t} h_1(\bar{x}, t) &= g_1(\bar{x}, t, h_1, \dots, h_k) \\ &\vdots \\ \frac{\partial}{\partial t} h_k(\bar{x}, t) &= g_k(\bar{x}, t, h_1, \dots, h_k). \end{aligned}$$

The functions f_i are used to give initial conditions depending on the parameters \bar{x} and the functions g_i are used to describe the differential equations. More succinctly, the above system of equations will be written as:

$$\begin{aligned} \bar{h}(\bar{x}, 0) &= \bar{f}(\bar{x}) \\ \frac{\partial}{\partial t} \bar{h}(\bar{x}, t) &= \bar{g}(\bar{x}, t, \bar{h}), \end{aligned}$$

with the bars on top of the functions indicating a vector of functions. Given functions \bar{f}, \bar{g} , by a solution to the corresponding IVP, we essentially mean the usual notion. However, since \bar{f} and \bar{g} may be partial, we further clarify what we intend. We mean to take functions \bar{h} such that for every \bar{x} , either $\bar{h}(\bar{x}, t)$ is not defined for any t , or is defined for t on some open interval containing 0; in this latter case, we take the interval to be a maximal such interval.

We can now define the operations we will be using (note that in the operations there is an implicit choice of which arguments of the function we choose to use; any choice is allowed):

1. The operation **ODE** takes as input some functions \bar{f} and \bar{g} of appropriate arities, sets up the corresponding IVP discussed above to obtain a solution \bar{h} (as discussed above), and returns h_1 , the first function in the list \bar{h} .

2. LI is the same as ODE, with the restriction that the \bar{g} are linear in the \bar{h} .
3. Let **comp** be the operation which takes two functions and returns the functional composition; a composition of partial functions is defined on the maximal well-defined domain.
4. Let **Inverse** be the operation which takes a function $f(t, \bar{x})$ such that for any \bar{x} , $f(t, \bar{x})$ is a bijection in t , such that $\frac{\partial}{\partial t}f(t, \bar{x}) > 0$ for all t . The operator **Inverse** then returns $\text{Inverse}(f, t)$, the inverse f^{-1} in t , i.e. $f(f^{-1}(t, \bar{x}), \bar{x}) = t = f^{-1}(f(t, \bar{x}), \bar{x})$.
5. The operation **UMU** takes a function $f(t, \bar{x})$ such that:
 - (a) For any \bar{x} , $f(t, \bar{x})$ is increasing (not necessarily strictly) in t , and
 - (b) For any \bar{x} , there is a unique t such that $f(t, \bar{x}) = 0$ (and at that t , $\frac{\partial}{\partial t}f > 0$),
then returns the function $\text{UMU}(f, t) =$ the unique t such that $f(t, \bar{x}) = 0$.

Besides constant functions like “0” we will also use the following basic functions:

- For a positive integer k , let $\theta_k(x) = \begin{cases} 0, & x < 0; \\ x^k, & x \geq 0. \end{cases}$, a \mathcal{C}^{k-1} version of the discontinuous function which indicates whether a number is to the left or right of zero.
- Let **P** be the set of projection functions (e.g. $P_{(2,1)}(x, y) = x, P_{(3,2)}(x, y, z) = y$, etc.).

We now define all the function algebras we will be using.

Definition 2.4 (*Function Algebras*)

- Let $BH_k^{(\infty)}$ be $FA[0, 1, -1, \theta_k, P; \text{comp}, LI, \text{UMU}]$
- Let $BH_k^{(c)}$ be the functions of $BH_k^{(\infty)}$ that can be defined using c or less applications of the operation **UMU**.
- Let $\mathcal{L}_k^{(\infty)}$ be $FA[0, 1, -1, \theta_k, P; \text{comp}, LI, \text{Inverse}]$
- Let $\mathcal{L}_k^{(c)}$ be the functions of $\mathcal{L}_k^{(\infty)}$ that can be defined using c or less applications of the operation **Inverse**.
- Let G_k be $FA[0, 1, -1, \theta_k, P; \text{comp}, \text{ODE}]$

To make the connection to Computable Analysis, we consider a limit operation.

Definition 2.5

- Let LIM^* be the operation which takes a function $f(t, \bar{x})$ and returns $F(\bar{x}) = \lim_{t \rightarrow \infty} f(t, \bar{x})$ if the limit exists, $|F(\bar{x}) - f(t, \bar{x})| \leq 1/t$, for $t \geq 1$, and F is \mathcal{C}^2 .
- Given a class of functions \mathcal{F} , we let $\mathcal{F}(LIM^*)$ denote the closure of \mathcal{F} under the operation LIM^* (for all the classes considered, it will in fact suffice to apply the operation LIM^* just once).

Note that we use the expression LIM^* to distinguish it from the limit LIM that we use in [9], which is the same, except that it has no restriction that the resulting F need be \mathcal{C}^2 .

We now state the main claim, that the compact restriction of our function algebra coincides with the compact computable functions which are \mathcal{C}^2 .

Theorem 2.6 *For $r \geq 3$, $\mathcal{C}^2 \cap [\mathbf{C}(\mathbb{R})] = [\mathbf{G}_r(\text{LIM}^*)]$.*

3 The Proof

The following 4 steps prove theorem 2.6, modulo a few lemmas (whose proofs follow).

1. A series of lemmas (3.2, 3.5, 3.11) will show the following inclusions for any $c \geq 0$ and $k \geq c + 3$:

$$\text{BH}_k^{(c)} \subseteq \mathcal{L}_k^{(c)} \subseteq \mathbf{G}_{k-c} \subseteq \mathbf{C}(\mathbb{R}).$$

2. Lemma 3.1 will show that for some fixed constant “bh”, and for any $k \geq 3$, we have:

$$\mathcal{C}^2 \cap [\mathbf{C}(\mathbb{R})] \subseteq [\text{BH}_k^{(\text{bh})}(\text{LIM}^*)].$$

3. $\mathbf{C}(\mathbb{R})$ is closed under LIM^* (easy to see; discussed in [9] after lemma 13 for the case of elementary computable functions).
4. Putting together the previous three steps we arrive at the following inclusions (for $k \geq \text{bh} + 3$) by closing under limits and considering compact restrictions:

$$\begin{aligned} \mathcal{C}^2 \cap [\mathbf{C}(\mathbb{R})] &\subseteq [\text{BH}_k^{(\text{bh})}(\text{LIM}^*)] \subseteq [\mathcal{L}_k^{(\text{bh})}(\text{LIM}^*)] \subseteq \\ &\subseteq [\mathbf{G}_{k-\text{bh}}(\text{LIM}^*)] \subseteq [\mathbf{C}(\mathbb{R})(\text{LIM}^*)] \cap \mathcal{C}^2 \subseteq [\mathbf{C}(\mathbb{R})] \cap \mathcal{C}^2 \end{aligned}$$

Theorem 2.6 follows by choosing $k = r + \text{bh}$, with $r \geq 3$ in the preceding inclusions.

The main result from [5] that we use is the following lemma (with an altered form discussed in [8]); also note that they did the construction for $k = 3$, but it is similar for larger k , while maintaining the restriction to just \mathcal{C}^2 functions. Furthermore, we include a fixed constant we call bh, allowed since their construction is uniform.

Lemma 3.1 *There is a fixed constant bh, such that for any $k \geq 3$, we have*

$$\mathcal{C}^2 \cap [\mathbf{C}(\mathbb{R})] \subseteq [\text{BH}_k^{(\text{bh})}(\text{LIM}^*)].$$

Note that [5] uses a restricted kind of LI operation, which they use in proving the opposite inclusion to the above lemma; however, we can simply use LI since we will not use their argument to show the opposite inclusion.

The rest of the paper will prove the three inclusions of step 1 above, in order. The following lemma proves the first inclusion of step 1.

Lemma 3.2 *For $k \geq 2$ and $c \geq 0$, $\text{BH}_k^{(c)} \subseteq \mathcal{L}_k^{(c)}$.*

Proof

We just need to show that any application of UMU in $\text{BH}_k^{(\infty)}$ can be simulated by a single application of Inverse in $\mathcal{L}_k^{(\infty)}$. Suppose $f(t, \bar{x})$ satisfies the conditions for the application of UMU . This means for any \bar{x} , f is non-decreasing in t and has a unique zero. Let $F(t, \bar{x}) = \theta_k(f)e^t - \theta_k(-f)e^{-t}$, and we can check that F has the same zero as f for any \bar{x} , and also has the property that its derivative is positive at all the t except possibly at its zero, where it may be 0. Now consider the function $G(t, \bar{x}) = f(t, \bar{x}) + F(t, \bar{x})$. It has the same zero as f and has a positive derivative in t , everywhere (using the fact that to apply UMU to f one of the conditions was that its derivative in t be positive at the zero of t). Now to get the root given by $\text{UMU}(f, t)$, we let H be $\text{Inverse}(G, t)$ and return $H(0, \bar{x})$.

■

Now we work towards the second inclusion of step 1. Note that when working in \mathbf{G}_k , the ODE operation can solve differential equations with the initial condition not just for $t = 0$, but for t equal to a constant definable in \mathbf{G}_k . The following often used lemma follows by an inductive proof on \mathbf{G}_k , using the key fact that from $y' = f(y, t)$, if f is \mathcal{C}^k , then so is the solution y ; also recall that the basic function θ_k is \mathcal{C}^{k-1} .

Lemma 3.3 *Any function in \mathbf{G}_k is \mathcal{C}^{k-1} on its domain.*

We often want at least $k \geq 3$, since the guarantee of \mathcal{C}^2 functions allows us to switch the order of partial derivatives.

Lemma 3.4 *For $k \geq 1$, if $f(t, \bar{x}) \in \mathbf{G}_k$ and f satisfies the conditions:*

1. $\frac{\partial}{\partial t}f > 0$
2. $\frac{\partial}{\partial t}f \in \mathbf{G}_k$

then $\text{Inverse}(f, t) \in \mathbf{G}_k$.

Proof

First, we notice that for any function $f(t, \bar{x}) \in \mathbf{G}_k$ we have following property: *For any \bar{x} such that $f(t, \bar{x})$ is defined for some t , there are some constants (independent of t , but not \bar{x}) $\alpha, \beta \in \mathbf{G}_k$ such that $f(\alpha, \bar{x})$ is defined and $f(\alpha, \bar{x}) = \beta$.* The property is easily verified by structural induction on the description of f in \mathbf{G}_k . In addition, \mathbf{G}_k contains the function $h(z) = 1/z$, since it can be defined by the ODE $h(1) = 1$ and $h' = -h^2$.

Now we prove the lemma by applying the Inverse Function Theorem to $f(t, \bar{x})$, where \bar{x} is fixed. By hypothesis we know $\frac{\partial}{\partial t}f \in \mathbf{G}_k$ and $\frac{\partial}{\partial t}f \neq 0$; additionally we know that \mathbf{G}_k contains h and is closed under composition, and thus we can define the differential equation

$$\frac{\partial}{\partial t}f^{-1}(t, \bar{x}) = \frac{1}{\frac{\partial}{\partial t}f(f^{-1}(t, \bar{x}), \bar{x})}$$

in \mathbf{G}_k , with initial condition is given by $f^{-1}(\beta, \bar{x}) = \alpha$. Therefore, $f^{-1}(f(t, \bar{x}), \bar{x}) = t = f(f^{-1}(t, \bar{x}), \bar{x})$, i.e. $f^{-1} = \text{Inverse}(f, t)$.

■

The next lemma shows that c applications of the inverse operation are captured by G_{k-c} , the second inclusion in step 1 of the above proof outline.

Lemma 3.5 For $c \geq 0$ and $k \geq c + 3$, we have $\mathcal{L}_k^{(c)} \subseteq G_{k-c}$.

Proof

We will prove a stronger result:

We show that if $g \in \mathcal{L}_k^{(c)}$, then $g \in G_{k-c}$ and $g' \in G_{k-c-1}$, where g' denotes any partial derivative of g .

We first notice that $G_k \subseteq G_{k-1}$ since $\theta_k(t) = t \theta_{k-1}(t)$. The proof will be done by induction first in c and then in the description of the function. We skip the base case ($c = 0$) since it can easily be proved using the techniques below and we proceed to prove the induction step. Let's then suppose that if $f \in \mathcal{L}_k^{(c-1)}$, then $f \in G_{k-c+1}$ and $f' \in G_{k-c}$, and consider some function $g \in \mathcal{L}_k^{(c)}$. We now proceed by structural induction on $\mathcal{L}_k^{(c)}$:

- (1) g is a basic function of $\mathcal{L}_k^{(c)}$ and therefore is in G_{k-c} . All basic functions have derivatives in $G_{k-c} \subseteq G_{k-c-1}$ except for θ_k whose derivative is $k \theta_{k-1}$ and belongs to G_{k-c-1} .
- (2) For composition, let $g = u \circ v$, where $u, v \in G_{k-c}$. Therefore $g \in G_{k-c}$ and since $(u \circ v)' = (u' \circ v) \cdot v'$, all the derivatives of g are in G_{k-c-1} by inductive hypothesis.
- (3) Suppose that $g(t, \bar{x})$ is defined by linear differentiation with respect to t , i.e.

$$g(0, \bar{x}) = u(\bar{x}), \quad \frac{\partial}{\partial t} g(t, \bar{x}) = v(t, \bar{x}) g(t, \bar{x})$$

for some $u, v \in G_{k-c}$. Then $g \in G_{k-c}$. We consider the two types of derivatives: $\frac{\partial}{\partial t} g$ and $\frac{\partial}{\partial x_i} g$. It is clear that $\frac{\partial}{\partial t} g \in G_{k-c} \subseteq G_{k-c-1}$. For $\frac{\partial g}{\partial x_i}$ we calculate:

$$\frac{\partial g}{\partial x_i}(0, \bar{x}) = \frac{\partial u}{\partial x_i}(\bar{x}) \quad \text{and} \quad \frac{\partial}{\partial t} \frac{\partial g}{\partial x_i} = \frac{\partial}{\partial x_i} \frac{\partial g}{\partial t} = \frac{\partial v}{\partial x_i} g + v \frac{\partial g}{\partial x_i}$$

since the derivation variables t and x_i can be switched (justified because $g \in G_{k-c}$ and $k \geq c + 3$, so we apply lemma 3.3 to conclude that g is \mathcal{C}^2). By hypothesis, $\frac{\partial u}{\partial x_i}, \frac{\partial v}{\partial x_i} \in G_{k-c-1}$ and $v \in G_{k-c-1}$ and thus we obtain a differential equation that defines $\frac{\partial g}{\partial x_i}$ in G_{k-c-1} .

- (4) Suppose that $g = \text{Inverse}(f, t)$, where $f \in \mathcal{L}_k^{(c-1)}$; note that to apply **Inverse**, we are assuming f is a bijection in t , such that $\frac{\partial f}{\partial t} > 0$. By inductive hypothesis $f \in G_{k-c+1} \subseteq G_{k-c}$ and $f' \in G_{k-c}$. By Lemma 3.4 we conclude that $g \in G_{k-c}$ also. Let's look at the two types of derivatives of g : $\frac{\partial g}{\partial t}$ and $\frac{\partial g}{\partial x_i}$. By the Inverse Function Theorem,

$$\frac{\partial}{\partial t} g(t, \bar{x}) = \frac{1}{\frac{\partial}{\partial t} f(g(t, \bar{x}), \bar{x})}$$

This shows that $\frac{\partial g}{\partial t}$ is in $\mathbf{G}_{k-c} \subseteq \mathbf{G}_{k-c-1}$. To prove the claim for $\frac{\partial g}{\partial x_i}$ we derivate $\frac{\partial g}{\partial t}$ with respect to x_i and switch the order of the derivatives. This leads to a linear differential equation in $\frac{\partial g}{\partial x_i}$ which involves f , g and $\frac{\partial f}{\partial t}$ in $\mathbf{G}_{k-c} \subseteq \mathbf{G}_{k-c-1}$ and the second derivatives $\frac{\partial^2 f}{\partial t^2}$ and $\frac{\partial^2 f}{\partial t \partial x_i}$. Using similar techniques, it can be shown that the second derivatives of f are in \mathbf{G}_{k-c-1} . As a result, $\frac{\partial g}{\partial x_i}$ is also in \mathbf{G}_{k-c-1} . ■

We will now need to introduce some theory from [13] in order to prove the last inclusion in step 1 (recall that it was $\mathbf{G}_k \subseteq \mathbf{C}(\mathbb{R})$).

Definition 3.6 *A subset $E \subseteq \mathbb{R}^k$ is **r.e.-open** if it is open and there are computable sequences of rationals $\{a_n\}$ and $\{r_n\}$ such that $E = \bigcup_{n=0}^{\infty} B(a_n, r_n)$, where $B(a, r)$ is the open ball in \mathbb{R}^k with center a and radius r .*

In general, an r.e.-open subset can be exhibited in many ways. At a certain point we will want to make sure that the r.e.-open set is exhibited in the following “robust” manner.

Definition 3.7

- *A **rational ball** is an open ball with a rational number radius, which has a vector of rationals as its center.*
- *An r.e.-open set is given **robustly** if for any ball B in its representation, and any rational ball $A \subseteq B$, A is in the representation.*

Given an r.e. representation of a set, we can devise another r.e. representation which is robust by enumerating the original representation, while regularly pausing in order to put in appropriate rational sub-balls of the balls already enumerated. Since we are dealing with computable rationals which we know exactly, this new enumeration is r.e.

Lemma 3.8 *If a set is r.e.-open, then this can be exhibited robustly.*

The next lemma will be used in the proof of the inductive step of lemma 3.11.

Lemma 3.9 *Given two computable functions with r.e.-open domains, their composition is computable with an r.e.-open domain.*

Proof

It is well known that the composition of real computable functions is real computable. We consider the r.e.-open condition. Suppose f and g are computable unary functions, both with r.e.-open domains, \mathcal{F} and \mathcal{G} , respectively (restricting to unary functions simplifies our discussion to the consideration of intervals rather than balls). By lemma 3.8 we assume that \mathcal{G} is exhibited robustly; additionally, we can assume that if the interval G is in \mathcal{G} , then the closure \bar{G} is in the domain of g . Now to give an r.e. listing of the domain of $f \circ g$ we proceed as follows. We simultaneously list the intervals for \mathcal{F} and \mathcal{G} . For an open interval G of \mathcal{G} , we consider its closure \bar{G} , which is in the domain of g . The minimum, m , and the maximum, M , of g on \bar{G} are computable. If the interval

$[m, M]$ is contained in one of the intervals in the listing of \mathcal{F} , then we include G in our listing of the domain of $f \circ g$. We continue this process, eventually comparing all intervals. There is the technical issue of how the algorithm decides a containment of the form: $[m, M] \subseteq (a, b)$. In the situation we are concerned with, the a and b are exact rational numbers, but the m and M are real numbers that we do not have exactly, but can compute to any desired precision. Since the procedure we are concerned with is a computable enumeration, we can continually revisit an interval $[m, M]$, so when we say to consider the containment $[m, M] \subseteq (a, b)$, we mean to revisit the question with increasingly more accurate m and M , till we can decide the containment. As long as $m \neq a$ and $M \neq b$ we can eventually decide the inclusion, and it will turn out not to matter that we can never decide the other inclusions.

Now we show that the algorithm works correctly. First note that we are not making the domain of $f \circ g$ too big, since by construction, we only include an interval in our listing when we are sure that $f \circ g$ is defined for all values in the interval. Second, for any x in the domain of $f \circ g$, we can show that our algorithm includes an interval containing x . For x in the domain of $f \circ g$, that means x is in the domain of g and $g(x)$ is in the domain of f . Since $g(x)$ is in the domain of f , there is some open interval $J = (a, b)$ of \mathcal{F} containing $g(x)$. Let $J' = (a', b')$ be any open interval containing $g(x)$, not necessarily in \mathcal{F} , such that $a < a' < b' < b$. By continuity of g , there is an open interval I containing x and such that $g(I) \subseteq J'$. Some open interval of \mathcal{G} must contain x , and so by the robustness of \mathcal{G} , there is a rational interval R in \mathcal{G} such that $x \in R \subseteq I$. At some point the algorithm considers the inclusion $g(\bar{R}) \subseteq J$. We know $g(\bar{R}) \subseteq \bar{J}'$, thus not only do we have $g(\bar{R}) \subseteq J$, but we know that the endpoints of the interval J are not the same as those of $g(\bar{R})$, so we can eventually decide the inclusion, putting R , which contains x , into our listing.

■

The next lemma follows easily from the hard work of [13] (the proof below points to the relevant parts of that paper. Note that in that paper, though not always stated, by a “computable function” they mean to include the restriction to an r.e.-open or “r.e.-closed” domain; we do not concern ourselves with “r.e.-closed”).

Lemma 3.10 *If f is a \mathcal{C}^1 computable function with a computable gradient function, then the solution to the initial value problem $y' = f(t, y)$, $y(0) = a$ (a computable) is computable and the domain of y is r.e.-open.*

Proof

Since f and its gradient function are computable, theorem 2.7 of [13] shows that f is “effectively Lipschitz” (in fact theorem 2.7 just refers to derivatives, but, as discussed with D. S. Graça, it should refer to the gradient and then a similar proof works to prove it). We do not concern ourselves with the definition of this notion of Lipschitz, but simply note that theorem 3.1 of [13] shows that if f is \mathcal{C}^1 and effectively Lipschitz, then y is computable on its maximal interval of definition containing the initial condition; furthermore, this interval is r.e.-open.

■

Lemma 3.11 For $k \geq 3$, $G_k \subseteq \mathbf{C}(\mathbb{R})$.

Proof

We proceed by induction, showing the following holds for any function of G_k :

★ Any function and any partial derivative of it is computable, and furthermore have r.e.-open domains.

Note that in general a computable function need not have a computable derivative, nor is G_k closed under differentiation, so we include the condition on the derivative in \star . Now we discuss the three cases in the induction.

(1) For basic functions \star is true.

(2) For composition, consider functions $g(x)$ and $f(u, y)$ which are computable, with computable partial derivative functions, where all these functions have r.e.-open domains. We first note that the composition $h(x, y) = f(g(x), y)$ is computable with an r.e.-open domain, by lemma 3.9. For the derivatives note that:

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

$$\frac{\partial}{\partial x} h = \frac{\partial}{\partial u} f(g(x), y) \frac{\partial g}{\partial x}$$

Applying lemma 3.9 again, we see that both derivatives are computable with r.e.-open domains, because we have expressed them as compositions and products of computable functions with r.e.-open domains (by inductive hypothesis).

(3) For the ODE operation consider the initial value problem:

$$y(0, x) = g(x), \quad \frac{\partial}{\partial t} y(t, x) = f(t, y, x)$$

Note that to simplify the discussion we have assumed just a single equation and a single x (relaxing these assumptions we would just use the multi-variate chain rule below and get a larger linear system in the end). Inductively we assume that f and g and their partial derivative functions in t or y are computable with r.e.-open domains (even if not in the class); thus, due to the definition of the gradient in terms of its partial derivatives, we know the gradient is computable. So we can apply lemma 3.10 to say that y is computable with an r.e.-open domain. Now consider the derivatives of y . The derivative in t is just f , which is computable with an r.e.-open domain. For the derivative in x we start with the defining equation for y , and calculate:

$$\frac{\partial}{\partial x \partial t} y = \frac{\partial}{\partial x} f(t, y, x)$$

$$\frac{\partial}{\partial t} \left(\frac{\partial y}{\partial x} \right) = \frac{\partial f}{\partial y} \left(\frac{\partial y}{\partial x} \right) + \frac{\partial f}{\partial x},$$

where the change in the order of differentiation from $\partial x \partial t$ to $\partial t \partial x$ is justified because y is \mathcal{C}^2 . The last differential equation defines $\frac{\partial y}{\partial x}$ as a *linear* differential equation using the functions $\frac{\partial f}{\partial y}$ and $\frac{\partial f}{\partial x}$. Note that $\frac{\partial f}{\partial y} = \frac{\partial}{\partial y} f(t, y)$ should be understood as: Differentiate f with respect to its second argument and then compose the function y at this place. By inductive hypothesis, the derivatives $\frac{\partial}{\partial y} f$ and $\frac{\partial}{\partial x} f$ are computable. We already know that y is computable, so $\frac{\partial f}{\partial y}$ composed with y is computable. Since $\frac{\partial y}{\partial x}$ is defined by a linear differential equation using computable functions, it is computable. Furthermore, the domain of the solution $\frac{\partial}{\partial x} y$ is the same as $\frac{\partial}{\partial y} f$, since the differential equation is linear. Since the domain of $\frac{\partial}{\partial y} f$ is r.e.-open, so is the domain of $\frac{\partial}{\partial x} y$.

■

4 Future Work, Questions, and Conjectures

We now discuss possible improvements of our results. Three specific ways to improve the results are as follows: Removing the restriction to \mathcal{C}^2 functions, allowing more flexible domains (perhaps non-compact), and simplifying the function algebras. On the first point, as in our paper [9] which removed this restriction, we believe this is just an artifact of the current proof, and expect that this restriction can be removed. Concerning the second point, the use of compact domains is used in a significant way in the proof of [5] (i.e. they use this constraint to show a computable \mathcal{C}^2 function has a computable derivative; this is in general false for unbounded domains). We do not conjecture anything here, but will ask a question along these lines. On the third point, in line with our paper [8], we believe that we can dispense with the function θ_k , in essence showing that it suffices to work with an approximation of it that can be built with other functions in the respective function algebras. We are thus led to the following conjecture and question.

Conjecture Supposing \mathcal{F}_k is one of the function algebras we have considered in this paper, let $a\mathcal{F}$ be the same function algebra, but *without* the basic function θ_k (in such a case we also leave off the now useless subscript k). Then

$$[\mathbf{C}(\mathbb{R})] = [a\mathbf{BH}^{(\infty)}(\mathbf{LIM})] = [a\mathcal{L}^{(\infty)}(\mathbf{LIM})] = [a\mathbf{G}(\mathbf{LIM})].$$

Question Can the restriction to compact domains be removed, or can such a restriction be proved necessary in our context?

In our paper [8], we show how to use our “*method of approximation*” to remove the function θ_k for a different function algebra. While it seems that the same approach might work here, there are some challenges. As a final note, we point to a new paper [10] with the claim that theorem 3.1. of [13] can be improved by replacing the assumption that f be effective Lipschitz by the assumption that f be continuous and the solution to the differential equation is unique. This could mean an improvement of our result along with a simpler proof.

5 Acknowledgements

This work was partially supported by *Fundação para a Ciência e a Tecnologia* and EU FEDER POCTI/POCI, and grant SFRH / BPD / 16936 / 2004.

References

- [1] Blum, L., M. Shub and S. Smale, *On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines*, Bulletin of the American Mathematical Society **21** (1989), pp. 1–46.
- [2] Bournez, O. and M. L. Campagnolo, *A survey on continuous time computations*, in: S. Cooper, B. Löwe and A. Sorbi, editors, *New Computational Paradigms. Changing Conceptions of What is Computable*, Springer-Verlag, New York, 2008 pp. 383–423.
- [3] Bournez, O., M. L. Campagnolo, D. S. Graça and E. Hainry, *Polynomial differential equations compute all real computable functions on computable compact intervals*, Journal of Complexity **23** (2007), pp. 317–335.
- [4] Bournez, O. and E. Hainry, *Elementarily computable functions over the real numbers and \mathbb{R} -sub-recursive functions*, Theoretical Computer Science **348** (2005), pp. 130–147.
- [5] Bournez, O. and E. Hainry, *Recursive analysis characterized as a class of real recursive functions*, Fundamenta Informaticae **74** (2006), pp. 409–433.
- [6] Campagnolo, M. L., C. Moore and J. F. Costa, *Iteration, inequalities, and differentiability in analog computers*, Journal of Complexity **16** (2000), pp. 642–660.
- [7] Campagnolo, M. L., C. Moore and J. F. Costa, *An analog characterization of the Grzegorzcyk hierarchy*, Journal of Complexity **18** (2002), pp. 977–1000.
- [8] Campagnolo, M. L. and K. Ojakian, *Using approximation to relate computational classes over the reals*, in: J. Durand-Lose and M. Margenstern, editors, *MCU 2007*, Lecture Notes in Computer Science **4664** (2007), pp. 39–61.
- [9] Campagnolo, M. L. and K. Ojakian, *The elementary computable functions over the real numbers: applying two new techniques*, Archives for Mathematical Logic **46** (2008), pp. 593–627.
- [10] Collins, P. and D. S. Graça, *Effective computability of solutions of ordinary differential equations - the thousand monkeys approach*, in: *Proceedings of the Fifth International Conference on Computability and Complexity in Analysis (CCA 2008)*, 2008, in print.
- [11] Graça, D. S., *Some recent developments on Shannon’s general purpose analog computer*, Mathematical Logic Quarterly **50** (2004), pp. 473–485.
- [12] Graça, D. S. and J. F. Costa, *Analog computers and recursive functions over the reals*, Journal of Complexity **19** (2003), pp. 644–664.

- [13] Graça, D. S., N. Zhong and J. Buescu, *Computability, noncomputability and undecidability of maximal intervals of IVPs*, Transactions of the American Mathematical Society (2007), to appear.
- [14] Ko, K.-I., “Complexity Theory of Real Functions,” Birkhäuser, 1991.
- [15] Maass, W., *Networks of spiking neurons: the third generation of neural network models*, Neural Networks **10** (1997), pp. 1659–1671.
- [16] Moore, C., *Recursion theory on the reals and continuous-time computation*, Theoretical Computer Science **162** (1996), pp. 23–44.
- [17] Pour-El, M. B., *Abstract computability and its relation to the general purpose analog computer (some connections between logic, differential equations and analog computers)*, Transactions of the American Mathematical Society **199** (1974), pp. 1–28.
- [18] Rubel, L. A., *A survey of transcendently transcendental functions*, American Mathematical Monthly **96** (1989), pp. 777–788.
- [19] Shannon, C., *Mathematical theory of the differential analyzer*, Journal of Mathematical Physics **20** (1941), pp. 337–354.
- [20] Weihrauch, K., “Computable Analysis: An Introduction,” Springer-Verlag, 2000.