International Journal of Foundations of Computer Science
Vol. 25, No. 4 (2014) 373–389

DOI: 10.1142/S0129054114400012

AN ANALOGUE-DIGITAL CHURCH-TURING THESIS

EDWIN BEGGS

Department of Mathematics, Swansea University, Singleton Park
Swansea, SA2 8PP, Wales, UK
e.j.beggs@swansea.ac.uk

JOSÉ FÉLIX COSTA* and DIOGO POÇAS†

Instituto Superior Técnico, Universidade de Lisboa
Av. Rovisco Pais, 1049 - 001 Lisboa, Portugal
CMAF Centro de Matemática e Aplicações Fundamentais
Universidade de Lisboa
**fgc@math.ist.utl.pt*
† *diogopocas1991@gmail.com*

JOHN V. TUCKER

Department of Computer Science, Swansea University, Singleton Park
Swansea, SA2 8PP, Wales, UK
j.v.tucker@swansea.ac.uk

Received 14 November 2013
Accepted 15 February 2014
Communicated by G. Mauri

We argue that dynamical systems involving discrete and continuous data can be modelled by Turing machines with oracles that are physical processes. Using the theory introduced in Beggs *et al.* [2,3], we consider the scope and limits of polynomial time computations by such systems. We propose a general polynomial time Church-Turing Thesis for feasible computations by analogue-digital systems, having the non-uniform complexity class $BPP//log\star$ as theoretical upper bound. We show why $BPP//log\star$ should be replace $P/poly$, which was proposed by Siegelmann for neural nets [23,24]. Then we examine whether other sources of hypercomputation can be found in analogue-digital systems besides the oracle itself. We prove that the higher polytime limit $P/poly$ can be attained via non-computable analogue-digital interface protocols.

Keywords: Analogue computation; analogue-digital systems; hybrid systems; non-uniform complexity; Church-Turing Thesis.

1. Introduction

Consider a dynamical system involving discrete and continuous data. We suppose that the system can be modelled mathematically by algorithms operating on discrete data in discrete time but with real number parameters. We address the mathematical question:

What is the computational power of algorithms with real number parameters?

There are many examples of such systems, such as neural nets, analogue computers and hybrid systems. But even for familiar systems, the presence of real number parameters can lead to controversial answers.[a]

To address the question in general we will reflect on the raison d'être for real numbers in systems. A system accesses continuous data by some form of measurement. A part of the control structure of the system is able to read the expansion of some real number valued quantity, digit by digit. At any moment the system is in possession of only finitely many approximate measurements of the continuous quantities. Mathematically, at any stage the algorithm has received finitely many rational approximations to its real parameters. We propose that such dynamical systems have the following form:

> *Models of systems with real number parameters perform measurements governed by an algorithmic procedure that are combined with an algorithmic computation of arbitrary complexity.*

We will argue that the measurement of the real number parameter can be modelled by a special kind of oracle to the algorithm, which encodes an *advice function*. The use of advice functions can be found in the

(i) analogue recurrent neural nets (ARNN) of Siegelmann and Sontag (see [25]);
(ii) optical computers of Woods and Naughton (see [27]); and
(iii) mirror systems of Bournez and Cosnard (see [14]).

In the ARNN case, a subsystem of about eleven neurones performs a measurement of the unique non-rational weight of the network, *approximating its value both from above and from below*. Once the measurement is done, up to some precision, the computation resumes, simulated by a system of a thousand rational neurones interconnected with integer and a few rational weights. In the case of the optical computer, the physical parameters are encoded in the image, and the control part of the system operates to successively extract the bits of the real-valued coordinates of the pixels. The mirror systems are an analogue extension of the mirror system of Moore (see [22]).

To tackle the general question we will consider Turing machines with the ability of making measurements. Using the theory introduced in Beggs *et al.* [2,3], we consider measurement as an oracle to the machine. We will use Turing machines operating in polynomial time with advice functions taken from some familiar non-uniform complexity classes.

The interface between the oracle and the machine is an analogue-digital interface. Its protocols are more complex than the standard conventions for oracle queries. First, this oracle must have a *cost function* $T : \mathbb{N} \to \mathbb{N}$ that gives the number

[a]See the controversy over results for neural nets in [16,17,21,23].

of time steps allowed to perform measurements. The common dynamical systems having real parameters, perform measurements that cannot be accomplished in linear time, even in an idealised world. To take the simplest sort of measurement, in a balance scale the pans move with acceleration that depends on the difference of masses placed in them, in such a way that the time needed to detect a mass difference increases *exponentially with the number of bits of precision of the measurement*, no matter how small that difference may be. This measurement has an exponential cost that should be considered in the complexity of the decision problem.[b]

Second, there is the matter of precision: operations and tests can be performed with *infinite precision*, in the sense that the real is taken as a whole entity; or with *unbounded precision*, in the sense that the machine can obtain as many bits of the real number as needed; or with arbitrary finite but *fixed precision*, defined once and for all for the particular equipment in use. In any of these scenarios we are still in an idealised world. Such a model of computation requires a theory of computation with oracles that have a cost (for the measurement or consultation) and, indeed, can be stochastic (for the precision).

A possible objection is that a measurement (cf. [19]) is *never* exact, for however precise, sooner or later it finds the obstacle of the atomic structure — though even quantum theory is infested with real number parameters and concepts. In fact, classical measurement has its own theoretical domain (see [7,15,19,20]) and can only be conceived as an asymptotic procedure. As observed by Geroch and Hartle in [18]: *Regard number w as measurable if there exists a finite set of instructions for performing an experiment such that a technician, given an abundance of unprepared raw materials and an allowed error ε, is able by following those instructions to perform the experiment, yielding ultimately a rational number within ε of w.* It means that measurement — like complexity — can only be conceived asymptotically. Once we fix space or time resources, complexity as we know it disappears.[c]

Any oracle can be encoded in a real number just by concatenating in lexical order all the words of the oracle. A real number is the right way of incorporating an oracle in a system making numerical computations by sums, products etc., such as in the ARNN case or analogue networks. The neural model, the optical computer, the mirror system, etc., perform some measurement in linear time. However, a typical experiment to measure some quantity x (mass, position etc.) is nonlinear. It consists of performing the experiment with a test value z, for which we could test one or both of the comparisons "$z < x$" and "$x < z$". Both comparisons (or two-sided experiments) are considered in [2–4]. In [4], we considered one comparison

[b]In the neural net case, with piecewise linear activation functions, the cost function is like the standard oracle Turing machine: a one-step consultation device, since any further bit has the constant cost of k transitions, for some constant $k \in \mathbb{N}$ (see [25]). This is due to the fact that the activation function is piecewise linear instead of the common analytic sigmoid.

[c]For example, only regular languages can be decided in finite space. We could say that tapes can have as many cells as the number of particles in the observable universe, but such conditions do not lead to an interesting theory.

— threshold measurements — like the measurement of the threshold of a neurone, which can be approximated just from one side, since from the other side the neurone is always firing. Different types of measurements may reveal different complexity classes.

To sum up: thinking about measurement provides intuitions about real number oracles, namely that: (a) they are based on comparisons making approximations; (b) they have a cost, i.e. the oracle answers queries in a time $T : \mathbb{N} \rightarrow \mathbb{N}$, dependent on the size of the query, modelling the fact that successive approximations have a cost that is not necessarily linear in the number of bits of precision; (c) they can contain errors; and (d) they can be stochastic. Although experiments can be replaced by mathematical oracles of some kind, they provide valuable intuitions to better reason about analogue-digital systems. In this paper we propose a method of answering the general question and make a clear new statement of a general analogue-digital Church-Turing Thesis — one that differs from that proposed for the neural net case by Siegelmann in [23,24]. We also discuss for the first time the power of protocols between the digital computer and the physical device.

We will begin by introducing the analogue-digital model in Sec. 2. In Sec. 3 we exemplify with just one analogue-digital machine taken from previous papers and, in Sec. 4, we summarise the computational power of analogue-digital machines with a variety of physical oracles. The analogue-digital Church-Turing Thesis is then stated in Sec. 5. In Sec. 6, we look at time schedules for protocols. Finally, in Sec. 7, we address some open problems and next steps.

2. The Physical Oracle

Our object of study is the *analogue-digital Turing machine*. In [2,3,5,8,11] we characterised these types of machines and the complexity classes decided by them. There are three important components of an analogue-digital machine, which we describe individually:

AD machine = physical experiment + interface + Turing machine.

2.1. *On Turing machines*

Following [2,8,9], the Turing machines are equipped with one *input tape* and several *work tapes* for performing calculations. The control unit of the machine includes at fewest three special states to begin and end the computation: these are called the *initial state*, the *accepting state* and the *rejecting state*, respectively. The Turing machines are either deterministic or probabilistic. (No non-deterministic machine per se will be considered.)

The Turing machines have additional properties: one *query tape*, for the purpose of instructing the physical experiment, and a finite amount of states that are used for interacting with the physical experiment: one of these is called the *query state*, and the others refer to each possible outcome of the experiment. For the time being we consider three outcomes, and thus three additional states: the YES *state*, the NO *state* and the TIMEOUT *state*. Observe that there is a difference between what we call an oracle Turing machine and the usual definition where two additional states are considered (the YES state and the NO state) which represent the possible answers of an oracle (= a set) to a query. However, in our definition, the number of additional states is arbitrary (but finite) and equals the number of possible outcomes.

2.2. *On physical experiments*

In [2,4,5,9,12,13] we have analysed a variety of physical experiments. They have in common (a) some initial conditions that can be tuned to some specific values; (b) a physical process, depending on the initial conditions, which takes a (possibly infinite) amount of time; and (c) a finite set of possible results or outcomes. Thus, we will say that a physical experiment is completely characterised by a set of *initial conditions* $\mathcal{I}$, a set of *outcomes* $\mathcal{R}$, a time function $t_{exp} : \mathcal{I} \to \mathbb{R}$ and an *outcome function* $r : \mathcal{I} \to \mathcal{R}$. A physical theory is needed to specify these components.

2.3. *On the interface*

The interface between the Turing machine (digital) and the physical experiment (analogue) has two main components. First, the *protocol*, which is the sequence of instructions that operates a physical experiment. The protocol should begin by reading a query word from the query tape, and it should end by resuming the computation of the Turing machine in a particular state. Second, a *time schedule*, which is used to specify the time allowed for the experiment. In most cases we will want some way to interrupt a physical experiment that has been going on for too much time.

A time schedule T is a function $T : \mathbb{N} \to \mathbb{N}$. Our standard definition is that it is a *time-constructible* function.[d] But this condition can be changed, e.g., we could require only that T is computable and that $T(n) \geq n$. However, any computable function f can be majorised by some time-constructible function f' (see [1]).

3. On the Experiment

A portfolio of experiments has been described in [5,10,13], as sophisticated as the Rutherford's scattering experiment in a Coulomb field. We will now focus on the *Broken Balance Experiment*. There are two main reasons for doing so.

[d]A function $f : \mathbb{N} \to \mathbb{N}$ is said to be time constructible if there is a deterministic Turing machine $\mathcal{M}$ and a natural number n_0 such that for any input word z of size $|z| = n > n_0$ the machine $\mathcal{M}$ halts after exactly $f(n)$ steps.

The balance experiments are fairly simple to analyse and understand, as they contain the most basic properties of two-sided and threshold experiments. Also, their proof techniques are applicable in the other experiments, so that the main results concerning complexity classes should not be different.

The threshold version of the balance scale, also known as the broken balance scale, consists of a balance scale with two pans (see Fig. 1). In the right pan we have some body with an unknown mass a. To measure a we place test masses z on the left pan of the balance: if $z < a$, then the scale will not move since the rigid block prevents the right pan from moving down; if $z > a$, then the left pan of the scale will move down, which will be detected in some way; if $z = a$, then we assume that the scale will not move since it is in equilibrium.

We assume several features of the apparatus, namely: we can take a to be a real number in the interval $[0, 1]$; the mass z can be set to any dyadic rational in the interval $[0, 1]$; a pressure-sensitive stick is placed below the left side of the balance, such that, when the left pan touches the pressure-sensitive stick, it reacts producing a signal; the mass z can be set so that the system begins in absolute rest; the pressure required to trigger the pressure stick is small enough so that a signal is always produced whenever the left pan of the scale sinks; the friction between the masses and the pans is large enough so that these will not slide away from their original position once the scale is in motion; and the bar on which the masses are placed is made of an homogeneous material, so that the two pans have exactly the same weight.

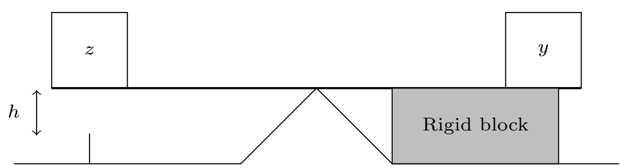

Fig. 1. Schematic representation of the *broken balance experiment.*

Definition 1. *For $a \in [0, 1]$, we denote by $BBE(a)$ the broken balance experiment with unknown mass a, which is defined by the following properties: (a) initial conditions $\mathcal{I} = [0, 1]$ and set of outcomes $\mathcal{R} = \{\text{YES}\}$, (b) experimental time function t_{exp} defined on $(a, 1]$ such that $t_{exp}(z) = (\frac{z+a}{z-a})^{1/2}$ if $z > a$, and (c) outcome function r defined on $(a, 1]$ such that $r(z) = \text{YES}$ if $z > a$. We also denote by BBE the family of experiments $BBE = \{BBE(a) : a \in [0, 1]\}$.*

Notice that the physical time taken grows exponentially with the precision of z as the test mass approaches the unknown mass, as in the two-sided case (without the rigid block), e.g. as in [12]. Moreover, if $z > a$, there are constants $C, D \in \mathbb{N}$

such that

$$\frac{D}{\sqrt{z-a}} < t_{exp}(z,a) < \frac{C}{\sqrt{z-a}} \ . \qquad (1)$$

3.1. *Precision*

Just as in previous investigations (see, e.g., [2,8,11]), we will consider different types of precision, i.e., different communication protocols between the experimenter/Turing machine and the analogue device/oracle. The query word $z \in \{0,1\}^{|z|}$ of length $|z|$ is converted to a dyadic rational in $[0,1)$ by taking zero point (the query word) in binary notation. Depending on the context, the experiment is performed either with infinite, unbounded or finite precision as follows:

(1) *infinite precision*: when the dyadic z is read on the query tape, a test mass $z' = z$ is simultaneously placed in the left pan.
(2) *unbounded precision*: when the dyadic z is read on the query tape, a test mass z' is simultaneously placed in the left pan such that $z - 2^{-|z|} \le z' \le z + 2^{-|z|}$. Here $z' \in \mathbb{R}$ is independently and uniformly distributed in the interval.
(3) *fixed precision* $\epsilon > 0$: when the dyadic z is read on the query tape, a test mass z' is simultaneously placed in the left pan such that $z - \epsilon \le z' \le z + \epsilon$. Here $z' \in \mathbb{R}$ is independently and uniformly distributed in the interval.

We write $\mathcal{M}$ for any such analogue-digital Turing machine.

3.2. *The time schedule*

To the oracle Turing machine model $\mathcal{M}$ we associate a schedule $T : \mathbb{N} \to \mathbb{N}$. On submitting the query z, the Turing machine waits a time $T(|z|)$, and then receives the answer to the query. By default, if no other answer is provided, the answer TIMEOUT is returned. We suppose that $T(\ell)$ is a time constructible function, i.e. that the Turing machine can itself count its own waiting time, a condition we might call *busy waiting*.

The threshold oracles have answers YES or TIMEOUT. For $y \in (0,1)$, the broken balance experiment BBE with unknown mass y is characterised by the following property:

> *For a test mass $z' \in [0,1)$ the experiment, having z' approaching y from above, takes a time provided in Definition 1. If the experiment completes (by touching the pressure sensitive stick), then $z' > y$ and the outcome of the experiment is Mass(z) =* YES. *If the experiment does not complete (i.e. if the experimental time t_{exp} exceeds the time schedule $T(|z|)$), the outcome of the experiment is Mass(z) =* TIMEOUT.

3.3. *Measurement*

We now give the procedure "Mass" for the BBE for some unknown mass y and some time schedule T; it comes in three cases. The algorithm Binary Search of Fig. 2 measures a mass, in the cases of infinite or unbounded precision.[e] The experimental procedure Mass is either deterministic (for the infinite precision case) or stochastic (for the unbounded precision case) and takes the scheduled time $T(\ell)$, where ℓ is the size of the query and T an arbitrary time constructible function.

PROTOCOL IP: "MASS": Infinite Precision Case
Receive as input the description of a dyadic rational z (possibly padded with 0s);
Place a mass z in the left pan;
Wait $T(\vert z\vert)$ units of time;
Check if the pressure stick has sent a signal. If so, return YES, otherwise TIMEOUT.

PROTOCOL UP: "MASS": Unbounded Precision Case
Receive as input the description of a dyadic rational z (possibly padded with 0s);
Place a mass z' in the left pan, where $z' \in (z - 2^{-\vert z\vert}, z + 2^{-\vert z\vert})$;
Wait $T(\vert z\vert)$ units of time;
Check if the pressure stick has sent a signal. If so, return YES, otherwise TIMEOUT.

PROTOCOL FP: "MASS": Finite Precision Case (ϵ)
Receive as input the description of a dyadic rational z (possibly padded with 0s);
Place a mass z' in the left pan, where $z' \in (z - \epsilon, z + \epsilon)$;
Wait $T(\vert z\vert)$ units of time;
Check if the pressure stick has sent a signal. If so, return YES, otherwise TIMEOUT.

ALGORITHM "BINARY SEARCH"

input number $\ell \in \mathbb{N}$; `% Number of places to the right of the leftmost 0`
  $x_0 := 0$; $m := 0$, $x_1 := 1$;
  while $x_1 - x_0 > 2^{-\ell}$ **do begin**
    $m := (x_0 + x_1)/2$;
    $s :=$ Mass$(m\rfloor_\ell)$; `% Procedure Mass takes time` $T(\ell)$
    if $s =$ YES **then** $x_1 := m$ **else** $x_0 := m$;
  end while;
output x_0.

Fig. 2. The three types of protocol: infinite, unbounded and fixed precision. The suffix operation $\rfloor_n$ on a word w, $w\rfloor_n$, denotes the prefix sized n of the ω-word $w0^\omega$, no matter the size of w. Mass$(m\rfloor_\ell)$ denotes the action that triggers the BBE experiment with mass (query word) $m\rfloor_\ell$.

[e]We will not discuss in this paper the case of fixed precision. The reader will find a full description in [5].

4. Complexity Classes

We have characterised several types of oracle and complexity class using physical examples and, subsequently, axiomatic specifications of their interfaces; see Sec. 7. Typically, in our later work, we use a particular physical experiment to motivate, illustrate and prove new results — e.g., the balance scale in the two-sided case and the broken balance scale in the threshold case. However, *most of these results can be formulated and proven assuming that the computation is carried out with an oracle* $\mathcal{O}$ of the form:

> *$\mathcal{O}$ receives a dyadic rational and returns one of a finite number of results; $\mathcal{O}$ may be deterministic or stochastic; $\mathcal{O}$ has a cost of consultation. If y is the unknown value:* (1) *Two-sided oracles are of the form* $\mathcal{O}_y(z) =$ LEFT *if* $z < y$ *and* $\mathcal{O}_y(z) =$ RIGHT *if* $z > y$ *and* (2) *Threshold oracles are of the form* $\mathcal{O}_y(z) =$ YES *if* $z > y$.

The determinism or stochasticity of the oracle is given by the notion of precision considered. The consultation cost is given by the experimental time function of the chosen experiment. Naturally, each experiment considered has an associated physical time; we now consider each oracle type in more detail to infer exactly the required conditions on the time function.

4.1. *Two-sided case*

In proving lower bounds, the only condition required for the experimental time is that it is bounded by $t_{exp}(z, y) \leq C/|z-y|^d$, which is exponential in the precision of the query. This means that for any other class of oracles with exponential cost we can reach in the same way the lower bounds of $P/log\star$,[f] $BPP//log\star$ and $BPP//log\star$,[g,h] for each type of precision, infinite, unbounded and fixed, respectively. For the upper bounds, we use the property that we can simulate two-sided oracle queries of size k with an advice polynomial in k, in polynomial time. In the same way, for any two-sided oracle such that the cost is increasing as the test value approaches the unknown value, we can argue that the upper bounds of $P/poly$, $BPP//poly$ and $BPP//poly$ are common to all classes of two-sided oracles.

[f] Let $\mathcal{B}$ be a class of sets and $\mathcal{F}$ a class of functions. The advice class $\mathcal{B}/\mathcal{F}$ is the class of sets A for which there exists $B \in \mathcal{B}$ and some $f \in \mathcal{F}$ such that, for every word w, $w \in A$ if and only if $\langle w, f(|w|)\rangle \in B$. For the prefix advice class $\mathcal{B}/\mathcal{F}\star$ some (prefix) function $f \in \mathcal{F}$ must exist such that, for all words w of length less than or equal to n, $w \in A$ if and only if $\langle w, f(n)\rangle \in B$. The role of advices in computation theory is fully discussed e.g., in [1], Chapter 5. We use $\log^2$ to denote the class of advice functions such that $|f(n)| \in \mathcal{O}((\log(n))^2)$.

[g] $BPP//\mathcal{F}\star$ is the class of sets A for which a probabilistic Turing machine $\mathcal{M}$, a prefix function $f \in \mathcal{F}\star$, and a constant $\gamma < \frac{1}{2}$ exist such that, for every length n and input w with $|w| \leq n$, $\mathcal{M}$ rejects $\langle w, f(n)\rangle$ with probability at most γ if $w \in A$ and accepts $\langle w, f(n)\rangle$ with probability at most γ if $w \notin A$.

[h] Note that in experiments where the lower/upper bounds are $P/poly$ for the infinite precision case, the unbounded comes together because $BPP//poly = P/poly$. In the threshold experiments, however, the unbounded and finite precision cases display identical power.

4.2. *Threshold case*

Just as in the previous case, the only condition relevant for the experimental time is that $t_{exp}(z, y) \leq C/(z-y)^d$, for $z > y$. In the broken balance experiment the value of d was set to $1/2$ but the same proof holds for any $d > 0$. This means that using any exponential cost threshold oracle we reach the lower bounds of $P/log\star$, $BPP//log\star$ and $BPP//log\star$ for the three types of precision. We did not study the upper bounds without restrictions on the time schedule. However, the proofs made for the two-sided oracles can be equally stated in this case. To simulate in polynomial time a threshold oracle query of size k we only need an advice of size polynomial in k, which is given by an approximation of the corresponding boundary number z such that $t_{exp}(z, y) = T(|z|)$. Thus we get in the same way the upper bounds of $P/poly$, $BPP//poly$ and $BPP//poly$.

We also consider the additional restriction of an exponential time schedule, which induces a logarithmic bound on the query sizes, giving us upper bounds of $P/log^2\star$, $BPP//log^2\star$ and $BPP//log^2\star$. These upper bounds, once more, only require that the cost function increases as the test value approaches the unknown value. However, it is possible to refine this bound, with the extra assumption that approximations to the boundary numbers are computable in polynomial time using some advice. For the broken balance case, since the experimental time function is given by $t_{exp}(z, y) = ((z + y)(z - y))^{1/2}$, we can invert this function to get the boundary numbers, so that the advice consists of the digits of the unknown value y. Using this, we obtain the upper bounds of $P/log\star$, $BPP//log\star$ and $BPP//log\star$. We now observe that the same reasoning can be made for the two-sided oracles, and so using two-sided oracles with exponential time cost, under the assumption that there is a procedure to compute in polynomial time the boundary numbers given some advice, we reach the upper bounds of $P/log\star$, $BPP//log\star$ and $BPP//log\star$.

4.3. *Most general assumptions*

That the lower and upper bounds of the analogue-digital machine — and, consequently, of many analogue models of computation with input and output processes — are quite general can be seen by listing and analysing the assumptions in the proofs. Here is a list of all assumptions on the computational cost in the proof of lower bounds:

- Experimental time, i.e. inherent physical time, t_{exp} is bounded by $C/|z - y|^d$ for some constants C and d, where z is an approximation of the unknown value y.
- Experimental time t_{exp} increases as $|z - y|$ decreases.
- Experimental time t_{exp} is differentiable and t'_{exp} fits between $C/|z - y|^d$ and $D/|z - y|^d$ for some constants C, D and d.

Assumptions on the computational cost in the proof of upper bounds are:

- Experimental time t_{exp} increases as $|z - y|$ decreases.
- There is a procedure to simulate in polynomial time queries of size k using $O(k)$ bits of advice.
- There is a procedure to compute in polynomial time the first k bits of numbers z_k that satisfy equations such as $t_{exp}(z_k, 1/2) = T(k)$ (so-called *boundary numbers*) or $t_{exp}(z_k, 1/2) = k$ (so-called *section numbers*) using $O(k)$ bits of advice.

The assumptions allow us to prove that certain interface axioms are satisfied from which we can establish the computational power of systems [11]. They are satisfied by many common choices of experimental time functions, such as: functions of the form $t_{exp}(z) = C/|z - a|^d$, and $t_{exp}(z) = (z + a)^b/|z - a|^d$.

We now observe that the bounds for the two-sided and threshold oracles are essentially the same. In fact, we can state that the power of Turing machines, when coupled with either the two-sided or threshold oracles, is boosted to a class between $P/log\star$ ($BPP//log\star$ using non-infinite precision) and $P/poly$. The class is exactly $P/log\star$ ($BPP//log\star$ using non-infinite precision) if we further assume that the time schedule is exponential. This bound is weaker than other bounds presented in literature: for example, [26] studied an analogue model that boosted the computational power (using polynomial resources) to $P/poly$. There is a reason for the difference in the classes obtained: in the neural networks, it is possible to extract a polynomial amount of information (that is, the bits of the real weights in the network) in polynomial time; however, in our model, since the experimental time functions are exponential, it seems that we can only extract a logarithmic amount of information in polynomial time. There is evidence that exponential cost is the cause for the upper bounds of logarithmic advice. However, in all of the physical experiments that we considered the experimental time function was seen to increase exponentially as the test value approaches the unknown value.

5. Analogue-Digital Church-Turing Thesis

We are led to question and make a first conjecture that this is common to all physical experiments: For all reasonable physical theories $\mathcal{T}$, for all reasonable physical measurements based upon $\mathcal{T}$, the $\mathcal{T}$-time for the physical experiment is at least exponential in the size of precision.

By *exponential* we generally mean a law of time of the form $t_{exp}(n) = 2^{kn}$ for some value of k different from 0 and n given by the number of zeros of precision in $|z - a|$, where a is the unknown to be measured and z the dyadic rational approximation to a. All experiments of measurement in nature have then an exponential cost — we conjecture. Note that by measurement we mean the same as in the analogue models of computation found in the literature: asymptotic measurement with unbounded precision or fixed precision but arbitrary large number of experiments. That is, in polynomial time with computable schedule, it is only possible to extract a logarithmic amount of information from a physical experiment of measurement, i.e. the class of sets decided in polynomial time by analogue-digital machines using oracles arising from physical experiments is contained in $BPP//log\star$. We are led to make a conjecture about the computational capabilities of analogue systems.

Analogue-digital Church-Turing Thesis. *No possible abstract analogue-digital device can have more computational capabilities in polynomial time than* $BPP//log\star$.

The complexity class BPP originates with coupling a Turing machine to an independent fair coin toss oracle. According to our current understanding of physics, such devices are constructible using radioactive decay (with a little wastage). However from our point of view here, we need to understand the uncertainty arising from a measurement process with errors. Given such an error prone experiment, we can try to get a better answer by averaging the same experimental setup over a large number of trials. We assume that each time such a measurement is set up in what we perceive to be an identical fashion, that the probabilities of the outcomes are the same, and that the result of each repeat of the experiment is independent from that of the other repetitions of the experiment. This means that we can model the outcomes of the repeated experiments using our coin toss oracle, giving a result involving BPP.

6. Under What Conditions can we Boost $BPP//log\star$?

We have studied analogue-digital machines operating in polynomial time with time schedules in $\Omega(2^{k/2})$ to comply with the intrinsic physical time of the experiment, obtaining the lower and upper bounds of $P/log\star$ in the case of infinite precision. Now we will study the dependence of computational capabilities on the time schedule assumptions. We consider three classes of time schedule: time-constructible functions, computable increasing total functions and increasing total functions. In the latter case, the computational power of the analogue-digital machine rises once more to $P/poly$.

We will work with the BBE machine of Sec. 3. We will remove all incomputability and uncertainty that may be present in the setup by taking the unknown mass a to be exactly 1/2 and considering only infinite precision protocols.

Definition 2. *Let f be an increasing total function. We denote by $AP(f)$ the class of sets decidable in polynomial time by an analogue-digital machine using the physical oracle with time schedule f, infinite precision and unknown $a = 1/2$. If $\mathcal{F}$ is a class of increasing total functions then let $AP(\mathcal{F}) = \bigcup_{f\in\mathcal{F}} AP(f)$.*

Let IN denote the class of *all* increasing total functions, CI denote the class of all computable, increasing total functions and TC denote the class of all time constructible functions. It is obvious that $TC \subset CI \subset IN$ and thus

$AP(TC) \subseteq AP(CI) \subseteq AP(IN)$. We now give alternative descriptions of these classes in terms of non-uniform complexity classes.

Proposition 3. *$AP(IN) = P/poly$.*

Proof: First we assume that $A \in P/poly$. Polynomial advice Turing machines are polynomial time equivalent to tally oracle Turing machines (see [1]), i.e. $P/poly = \bigcup_{S \text{ tally}} P(S)$. Therefore we may assume that A is decidable in polynomial time by a Turing machine $\mathcal{M}$ with some tally set S as oracle. We consider an analogue-digital machine operating with infinite precision, unknown 1/2 and time schedule

$$T(k) = \begin{cases} 2k+1 & \text{if } 0^k \in S \\ 2k & \text{if } 0^k \notin S \end{cases} .$$

If a sequence of dyadic rational numbers z_k exists, such that $|z_k| = k$ and $2k < T_{exp}(z_k, 1/2) < 2k+1$, then we can decide whether $0^k \in S$ by querying the oracle with z_k. Since the previous inequalities imply that

$$\frac{1}{2} + \frac{1}{(2k+1)^2+1} < z_k < \frac{1}{2} + \frac{1}{(2k)^2+1} \ ,$$

for large enough k, the difference between the lower and upper bounds on z_k is greater than 2×2^{-k}. Thus, for large enough k, the dyadic rational (notation $\downharpoonleft$ is explained in the caption of Fig. 2) $z_k = (\frac{1}{2} + \frac{1}{(2k)^2+1})\downharpoonleft_k$ is such that $2k < t_{exp}(z_k, 1/2) < 2k+1$. (We can assume without loss of generality that $0^k \notin S$ for the small values of k.) For an input word x of size n, the analogue-digital machine $\mathcal{M}'$ simulates $\mathcal{M}$ for that input and, when reaching a query state, the machine $\mathcal{M}'$ counts the number of 0s, let us say k, in the query tape. For small values of k the machine moves to the state NO. For large values of k the machine performs the protocol call $\text{Mass}_{IP}(z_k)$ of Fig. 2, where $z_k = (\frac{1}{2} + \frac{1}{(2k)^2+1})\downharpoonleft_k$. If the answer is YES, then $t_{exp}(z_k, 1/2)) < T(k)$, $T(k) = 2k+1$ and $0^k \in S$; in this case the machine $\mathcal{M}'$ enters the state YES. Otherwise $T(k) < t_{exp}(z_k, 1/2)$, $T(k) = 2k$ and $0^k \notin S$; in this case the machine enters the state NO (both states being considered regular states). It is obvious that the machine decides the same set A in polynomial time since the simulation of $\mathcal{M}$ runs in polynomial time and all calls to the oracle S are simulated by the experiment in polynomial time. It follows that $A \in AP(IN)$.

Conversely, let $A \in AP(IN)$. Then A is decidable by an analogue-digital machine $\mathcal{M}$ in polynomial time with unknown mass 1/2 and infinite precision. Thus there is a polynomial $p(n)$ that bounds the maximum size of any possible query in the computation for an input word of size n. Consider the advice function f such that

$$f(n) = z_1\downharpoonleft_1 \# z_2\downharpoonleft_2 \# \cdots \# z_{p(n)}\downharpoonleft_{p(n)} \ ,$$

where z_i is the boundary number such that $t_{exp}(z_i, 1/2) = T(i)$. Then $f \in \text{poly}$ and it can be used to simulate any oracle query of size less than or equal to $p(n)$. Thus, we can devise a machine deciding A in polynomial time using f as advice. Simply simulate $\mathcal{M}$ for the same input and replace oracle calls with a comparison between the query word and the appropriate $z_i\rfloor_i$. It follows that $A \in P/poly$. □

Proposition 4. *$AP(CI) = P/poly \cap REC$, where REC is the class of recursive sets.*

Proof: Let $A \in P/poly \cap REC$. Since $P/poly = \cup_{S \text{ tally}} P(S)$, we may assume that A is decidable in polynomial time by a Turing machine using as advice some tally set S. Since A is recursive, we can also assume that S is recursive. Now consider once more the time schedule T such that

$$T(k) = \begin{cases} 2k+1 & \text{if } 0^k \in S \\ 2k & \text{if } 0^k \notin S \end{cases} .$$

We can repeat the same reasoning as we did in Proposition 3 and conclude that A is decidable in polynomial time by an analogue-digital machine using the oracle with unknown 1/2, infinite precision and time schedule T. Moreover, since S is recursive we conclude that T is a total computable function. Thus $A \in AP(CI)$.

Let $A \in AP(CI)$. Since $AP(CI) \subseteq AP(IN)$, we conclude, by Proposition 3, that $A \in P/poly$. We will show that $A \in REC$. Consider an analogue-digital machine $\mathcal{M}$ with unknown 1/2 and time schedule T that decides A in polynomial time, where T is a computable, increasing total function. Then, for any k, we can compute the boundary numbers $z_k\rfloor_k$, where z_k is the number such that $t_{exp}(z_k, 1/2) = T(k)$ (see Definition 1). To decide A, the Turing machine just has to simulate $\mathcal{M}$ on the same input and, whenever in a query state, compute the appropriate $z_i\rfloor_i$ and compare this value with the query word. It follows that $A \in REC$, and so we conclude that $A \in P/poly \cap REC$. □

Proposition 5. *$AP(TC) = P$.*

Proof: If $A \in P$, then it is trivially decidable by an analogue-digital machine in polynomial time that does not make any oracle consultation.

If $A \in AP(TC)$, then A is decidable by an analogue-digital machine $\mathcal{M}$ in polynomial time using the oracle with unknown 1/2 and time-constructible schedule T. We show how to simulate any oracle query of size k in polynomial time. The Turing machine computes first $T(k)$, then computes the boundary number $z_k\rfloor_k$ from the equation $t_{exp}(z_k, 1/2) = T(k)$ (e.g., using the equality $z_k = 1/2 + 1/(T(k)^2 - 1)$ from Definition 1 or another equivalent equation, as discussed in Sec. 4.3), and compares this value with the query word z of size k. Since $\mathcal{M}$ runs in polynomial time, there is a polynomial p such that $T(k) \leq p(n)$. Since T is time-constructible, $T(k)$ can be computed in polynomial time $\mathcal{O}(p(n))$. The boundary number approximation $z_k\rfloor_k$ and the comparison $z =? z_k\rfloor_k$ can then be done in polynomial time. Since each oracle query can be simulated in polynomial time, together with the analogue-digital machine $\mathcal{M}$, we conclude that $A \in P$. □

7. Conclusion

We think that our Turing machine with physical oracle model captures (i) the computational scope and limits of computation by analogue-digital systems; (ii) the relationship between measurement and computation, intrinsic to processing real numbers; and (iii) the scope and limits of what can be measured (such as in [8]).

7.1. *Idealisation*

Reactions towards a gedankenexperiment, such as measuring mass (as in Sec. 3), can express dissatisfaction at the fact that such idealised devices cannot be built perfectly. Unfortunately, there seems to be a diffuse philosophical bias that considers the Turing machine to be an object of a kind different from theoretical models of experiments. Clearly, both the abstract physical experiment and the Turing machine are idealised for use in forms of gedankenexperiments. To implement a Turing machine the engineer would need either unbounded space and an unlimited physical support structure, or unbounded precision in some finite space to code for the contents of the tape. However, just as the experiment can be set up to some degree of precision, in the same way the Turing machine can be implemented up to some degree accuracy. Both objects, the Turing machine and the measurement device, are of the same ideal nature and, hence, we argue that the models allow us to study the power of adding real numbers to computing devices and the limits of what can be measured.

Type of Oracle		Infinite	Unbounded	Finite
Two-sided	lower bound	$P/log\star$	$BPP//log\star$	$BPP//log\star$
	upper bound	$P/poly$	$P/poly$	$P/poly$
	upper bound (w/ exponential T)	−−	−−	−−
Threshold	lower bound	$P/log\star$	$BPP//log\star$	$BPP//log\star$
	upper bound	−−	−−	−−
	upper bound (w/ exponential T)	$P/log\star$	$BPP//log\star$	$BPP//log\star$

Fig. 3. Results from investigations of several two-sided and threshold experiments.

7.2. *Comparison*

In [5] we introduced methods to study the computational power of threshold systems such as the neurone or the photoelectric cell, for which quantities can only be measured either from below or from above. We showed that Turing machines equipped with *threshold oracles* in polynomial time have a computational power below $BPP//log\star$, no matter whether the precision is infinite, unbounded or fixed.

We expect that analogue-digital systems in general cannot transcend such computational power and that this computational power is to analogue-digital systems as the Church-Turing Thesis is to human computation. Our result weakens the claims for other models of physical systems (see, e.g., $P/poly$ in [23,24]). In Fig. 3, we summarise the power of two-sided and threshold oracles in polynomial time.

Using the methods of threshold oracles it is, however, possible to prove the same upper bounds for the two-sided case, assuming exponential schedule. We don't know if other upper bounds can be established without assumptions on the time schedule.

7.3. *Some next steps*

These arguments and results are work in progress. We have investigated the core ideas and found many diverting questions and intriguing theorems. Immediate next steps are to tackle the third, most subtle, form of experiment that measures quantities that vanish ([6]); to analyse the role of time and, in particular, that of precision in timing; and to nail down axiomatic specifications of broad classes of experiments. This done we expect that the above case for $BPP//log\star$ will be stronger and easier to understand.

Acknowledgment

The research of José Félix Costa and Diogo Poças was supported by Fundação para a Ciência e Tecnologia, PEst — OE/MAT/UI0209/2011.

References

[1] J. L. Balcázar, J. Días and J. Gabarró, *Structural Complexity I*, 2nd edn. (Springer-Verlag, 1995).

[2] E. Beggs, J. F. Costa, B. Loff and J. V. Tucker, Computational complexity with experiments as oracles, *Proceedings of the Royal Society, Series A* (*Mathematical, Physical and Engineering Sciences*) **464**(2098) (2008) 2777–2801.

[3] E. Beggs, J. F. Costa, B. Loff and J. V. Tucker, Computational complexity with experiments as oracles II. Upper bounds, *Proceedings of the Royal Society, Series A* (*Mathematical, Physical and Engineering Sciences*) **465**(2105) (2009) 1453–1465.

[4] E. Beggs, J. F. Costa, D. Poças and J. V. Tucker, On the power of threshold measurements as oracles, *Unconventional Computation and Natural Computation* (*UCNC* 2013), eds. G. Mauri, A. Dennunzio, L. Manzoni and A. E. Porreca, *Lecture Notes in Computer Science* **7956** (Springer, 2013), pp. 6–18.

[5] E. Beggs, J. F. Costa, D. Poças and J. V. Tucker, Oracles that measure thresholds: The Turing machine and the broken balance, *Journal of Logic and Computation* **23**(6) (2013) 1155–1181.

[6] E. Beggs, J. F. Costa, D. Poças and J. V. Tucker, Computation with oracles that measure vanishing quantities, submitted (2014).

[7] E. Beggs, J. F. Costa and J. V. Tucker, Computational models of measurement and Hempel's axiomatization, *Causality, Meaningful Complexity and Knowledge Construction*, ed. A. Carsetti, *Theory and Decision Library A* **46** (Springer, 2010), pp. 155–184.

[8] E. Beggs, J. F. Costa and J. V. Tucker, Limits to measurement in experiments governed by algorithms, *Mathematical Structures in Computer Science* **20**(06) (2010) 1019–1050.

[9] E. Beggs, J. F. Costa and J. V. Tucker, Physical oracles: The Turing machine and the Wheatstone bridge, *Studia Logica* **95**(1–2) (2010) 279–300.

[10] E. Beggs, J. F. Costa and J. V. Tucker, The Turing machine and the uncertainty in the measurement process, *Physics and Computation, P&C* 2010, ed. H. Guerra

(CMATI – Centre for Applied Mathematics and Information Technology, University of Azores, 2010), pp. 62–72.

[11] E. Beggs, J. F. Costa and J. V. Tucker, Axiomatising physical experiments as oracles to algorithms, *Philosophical Transactions of the Royal Society, Series A* (*Mathematical, Physical and Engineering Sciences*) **370**(12) (2012) 3359–3384.

[12] E. Beggs, J. F. Costa and J. V. Tucker, The impact of models of a physical oracle on computational power, *Mathematical Structures in Computer Science* **22**(5) (2012) 853–879.

[13] E. Beggs, J. F. Costa and J. V. Tucker, Three forms of physical measurement and their computability, submitted (2013).

[14] O. Bournez and M. Cosnard, On the computational power of dynamical systems and hybrid systems, *Theoretical Computer Science* **168**(2) (1996) 417–459.

[15] R. Carnap, *Philosophical Foundations of Physics* (Basic Books, 1966).

[16] M. Davis, The myth of hypercomputation, In *Alan Turing: the life and legacy of a great thinker*, ed. C. Teuscher (Springer, 2006), pp. 195–212.

[17] M. Davis, Why there is no such discipline as hypercomputation, *Applied Mathematics and Computation* **178**(1) (2006) 4–7.

[18] R. Geroch and J. B. Hartle, Computability and physical theories. *Foundations of Physics* **16**(6) (1986) 533–550.

[19] C. G. Hempel, Fundamentals of concept formation in empirical science, *International Encyclopedia of Unified Science* **2**(7) (1952).

[20] D. H. Krantz, P. Suppes, R. Duncan Luce and A. Tversky, *Foundations of Measurement* (Dover, 2009).

[21] B. Loff and J. F. Costa, Five views of hypercomputation, *International Journal of Unconventional Computing* **5**(3–4) (2009) 193–207.

[22] C. Moore, Unpredictability and undecidability in dynamical systems, *Physical Review Letters* **64**(20) (1990) 2354–2357.

[23] H. T. Siegelmann, Computation beyond the Turing limit, *Science* (1995) 545–548.

[24] H. T. Siegelmann. *Neural Networks and Analog Computation: Beyond the Turing Limit* (Birkhäuser, 1999).

[25] H. T. Siegelmann and E. D. Sontag, Analog computation via neural networks. *Theoretical Computer Science* **131**(2) (1994) 331–360.

[26] H. T. Siegelmann and E. D. Sontag, On the computational power of neural networks, *J. Comp. Syst. Sciences* **50**(1) (1995) 132–150.

[27] D. Woods and T. J. Naughton, An optical model of computation, *Theoretical Computer Science* **334**(1–3) (2005) 227–258.