Math. Struct. in Comp. Science (2010), *vol.* 20, *pp.* 1019–1050. © Cambridge University Press 2010
doi:10.1017/S0960129510000356

Limits to measurement in experiments governed by algorithms[†]

EDWIN J. BEGGS[‡], JOSÉ FÉLIX COSTA[§] and JOHN V. TUCKER[‡]

[‡]*School of Physical Sciences, Swansea University, Singleton Park, Swansea, SA2 8PP, Wales, United Kingdom*
Email: {E.J.Beggs;J.V.Tucker}@Swansea.ac.uk
[§]*Instituto Superior Técnico, Technical University of Lisbon, 1049-001 Lisboa, Portugal and Centro de Matemática e Aplicações Fundamentais, University of Lisbon, 1649-003 Lisboa, Portugal*
Email: fgc@math.ist.utl.pt

Received 19 January 2010; revised 4 July 2010

We pose the following question: *If a physical experiment were to be completely controlled by an algorithm, what effect would the algorithm have on the physical measurements made possible by the experiment?*
In a programme to study the nature of computation possible by physical systems, and by algorithms coupled with physical systems, we have begun to analyse:

(i) the algorithmic nature of experimental procedures; and
(ii) the idea of using a physical experiment as an oracle to Turing Machines.

To answer the question, we will extend our theory of experimental oracles so that we can use Turing machines to model the experimental procedures that govern the conduct of physical experiments. First, we specify an experiment that measures mass via collisions in Newtonian dynamics and examine its properties in preparation for its use as an oracle. We begin the classification of the computational power of polynomial time Turing machines with this experimental oracle using non-uniform complexity classes. Second, we show that modelling an experimenter and experimental procedure algorithmically imposes a limit on what can be measured using equipment. Indeed, the theorems suggest a new form of uncertainty principle for our knowledge of physical quantities measured in simple physical experiments. We argue that the results established here are representative of a huge class of experiments.

1. Introduction

We consider performing a simple experiment to measure a physical quantity. The quantity, the method, the equipment and the exact experimental procedure that together constitute the experiment are based upon, or belong to, a physical theory – supposing the experiment to be sufficiently simple to involve only one theory. The experiment is carried out by a technician following the experimental procedure. The experimental procedure is a sequence

[†] The authors would like to thank EPSRC for their support under grant EP/C525361/1. The research of José Félix Costa is also supported by FEDER and FCT Plurianual 2007.

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

of instructions that initialise and set parameters, control and observe the equipment, then record, calculate and display the results. A more modern image would introduce a computer to help the technician manage the experiment. Indeed, it seems a small step to imagine the experiment performed *completely* by a computer. We then ask the following question:

If a physical experiment were to be completely controlled by an algorithm, what effect would the algorithm have on the physical measurements made possible by the experiment?

This is not a small step for theorists. Notice that the technician does not belong to the physical theory: the concept plays only an informal and unanalysed role in our understanding of measurement. The role of the technician is to implement a procedure by following a sequence of instructions that are precisely formulated in terms of the physical theory. Now, the sequence of instructions resembles an algorithmic procedure. Therefore, the idea of replacing the informal concept of a technician by the formal concept of an algorithm enables us to *contemplate* a new theoretical analysis of experimentation and measurement, one in which new properties, such as the process of performing an experiment, can be analysed using the methods of computability theory. In this paper we will show how the theory of algorithms can be used to make a deeper analysis of experiments and measurements.

For many years we have been thinking about what can be computed by experimenting with physical systems. In a series of papers (Beggs and Tucker 2006; 2007a; 2007b; Beggs *et al.* 2008a; 2008b; 2009a), we have begun to develop a methodology and mathematical theory to study the nature and limits of computations made possible by:

(i) physical systems in isolation; and
(ii) physical systems combined with algorithms.

With the aims of

(i) understanding the roles of data representations, procedures and equipment
(ii) determining the computational powers of physical systems,

we introduced four principles in Beggs and Tucker (2006; 2007a; 2007b) and, later, a fifth involving the idea of using a physical experiment as an oracle to a Turing machine in Beggs *et al.* (2008b; 2009a). We began the development of a theory of physical oracles by analysing a specific experiment and its use as an oracle to a Turing machine; the experiment measured position or distance using Newtonian dynamics and the oracle was termed the *scatter machine experiment* (SME). In the theory of physical oracles, the Turing machine interacts with an experiment via queries that become quite complicated and need to be managed by a *protocol*. We classified computational power using non-uniform complexity classes: in polynomial time the physical oracle SME allows Turing machines to compute non-uniform complexity classes such as $P/poly$ and $BPP//log*$, depending upon assumptions about the precision of the data. Our early experimental case studies revealed a great deal theoretically.

In an attempt to answer the question, in this paper we extend our methodology by adding a new principle (Principle 6), which defines our approach. Instead of focussing on

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

the power of the oracle to *boost* the computational power of the algorithm, we focus on the aim of

(iii) determining the power of the algorithm to *govern and control* the experiment.

We show that the extended mathematical theory of physical oracles accommodates both aims and can be used for two purposes:

$$\text{Boosting computation} \longleftarrow \text{Turing machine} + \text{protocol} + \text{experiment} \longrightarrow \text{Controlling measurement.}$$

In this paper, we will use a Turing machine to model an experimenter or technician following an experimental procedure. There is a similarity between an experimental procedure and that of an algorithmic procedure since they are both sequences of instructions, though the instructions are defined by radically different theories. Thus, the model of the technician following instructions is in essence *identical* to Turing's model of a human calculator performing a calculation. The extension to the theory of physical oracles leads to a computational model of measurement with significant properties, including:

(a) the time taken to perform experimental measurements is a function of accuracy;
(b) the existence of quantities that cannot be measured by any experimental procedure applied to equipment.

The model uncovers a new form of indeterminacy and suggests an uncertainty principle that imposes a limit to our knowledge of the values of the physical quantities involved in physical experiments in Newtonian mechanics. Our methods suggest that indeterminacy is a general property of experimental measurements.

First, we specify a new experiment that measures an unknown mass in Newtonian dynamics. The experiment uses a procedure involving controlled collisions between test particles and the unknown mass. We call the experiment the *collider machine experiment* (CME). The time taken to perform the experiment is *exponential* in the desired accuracy of the measurement. In this property and its consequences, the new experiment is more representative of physical measurement generally. Timing and scheduling plays a prominent role in the extended theory.

Following the methods of Beggs *et al.* (2008b; 2009a), we begin the classification of the computational power of this experimental oracle using non-uniform complexity classes. However, the lower bound results here differ from those of the experiment in Beggs *et al.* (2008b; 2009a) for reasons that are significant (see the theorems in Section 6 and the discussion in Section 9.2).

Theorem 1.1. Turing machines equipped with the collider machine experiment as an oracle can compute the following complexity classes in polynomial time for each of the following assumptions on how precisely the mass of the test particle is known:

— Infinite precision – $P/log*$.
— Finite unbounded precision – $P/log*$.
— Finite fixed precision – $BPP//log*$ (assuming a uniform distribution for the mass of the test particle).

We will then reverse the direction in which the theory is applied to address the main question. Modelling experimental procedures algorithmically leads to results that

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

challenge a fundamental assumption of the classical physical world, namely, that values of a physical quantity μ belong in experimental procedures as patterns of the form $\mu \pm \Delta\mu$, where theoretically the value $\Delta\mu$ can be made as small as we wish. For a century, this assumption confronted the quantum world wherein measurements cannot be made with errors as small as we wish. We show that the mathematics of computation imposes limits on what can be measured (Theorem 7.13).

Theorem 1.2. There are uncountably many masses μ such that, for every experimental procedure governing the CME, it is only possible to determine finitely many digits of μ, even allowing arbitrary long run times for the procedure.

Although not every mass can be measured, no matter how we vary the procedure, we can show the following result.

Theorem 1.3. There is a universal experimental procedure that measures every mass μ that can be measured by some procedure using the CME.

This is the third in our series of basic papers on physical oracles: the first of them, Beggs *et al.* (2008b), is recommended for background and essential technical material (such as, non-uniform complexity and time constructible functions). In Beggs *et al.* (2008c; 2009b), we argue that algorithms with physical oracles occur naturally in physics. In Beggs *et al.* (2009c), we began to study relativisations of the $P = NP$ problem to physical oracles.

2. Computation and physical systems

In this section we summarise five methodological principles for determining the computational power of physical systems, and then introduce a new principle determining the power of experimental procedures.

2.1. *Methodological principles for experimental computation*

To explore what may be computed through experimentation with physical systems, we must carry out the analysis from first principles that are independent of the theory of algorithms. For conceptual clarity, and mathematical precision and detail, we proposed, in Beggs and Tucker (2006; 2007a), the following four principles and stages for an investigation of any class of experimental computations.

Principle 1 – Defining a physical subtheory: *Define precisely a subtheory T of a physical theory and examine experimental computation by the systems that are valid models of the subtheory T.*

Principle 2 – Classifying computers in a physical theory: *Find systems that are models of T that can through experimental computation implement specific algorithms, calculators, computers, universal computers and hyper-computers.*

Principle 3 – Mapping the border between computer and hyper-computer in physical theory: *Analyse what properties of the subtheory T are the source of computable and non-computable*

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

behaviour and seek necessary and sufficient conditions for the systems to implement precisely the algorithmically computable functions.

Principle 4 – Reviewing and refining the physical theory: *Determine the physical relevance of the systems of interest by reviewing the truth or valid scope of the subtheory. Criticism of the system might require strengthening the subtheory T in different ways, leading to a portfolio of theories and examples.*

Our methodology requires a careful formulation of a physical theory T, which we have discussed at length elsewhere (Beggs and Tucker 2006; 2007a). From the theory T are derived the central concepts of *experimental procedures* and *equipment*. We need to control T to lay bare all the concepts and technicalities to be found in examples and to classify their computational behaviour. The theory T is everywhere, so we are actually studying *T-computability* and *T-computational complexity*. Our approach has been applied in a new discussion of the physical basis of the Church–Turing Thesis in Ziegler (2009).

2.2. *Methodological principles for combining experiments and algorithms*

We will now extend our methodology to consider the *interaction* between experiments and algorithms.

First, we consider the use of an experiment as a component to boost the performance of an algorithm or class of algorithms. In this case, computations involve some form of *protocol* for exchanging data between physical system and algorithm. A simple general way to do this is to choose an algorithmic model and incorporate the experiment as an oracle. There are many algorithmic models, but the advantage of choosing Turing machines is the rich theory of computational complexity that has been developed around them.

Suppose we wish to study the complexity of computations by *Turing machines with experimental oracles*. Given an input string w over the alphabet of the Turing machine, in the course of a finite computation, the machine will generate and receive a finite sequence of oracle queries and answers. Specifically, as the ith question to the oracle, the machine generates a string that is converted into a rational number z_i and used to set an input parameter, which is a physical quantity, p_i to the equipment. The experiment is performed and, after some delay, returns as output a rational number measurement, or qualitative observation, which is converted into a string or state for the Turing machine to process. The Turing machine may pause while waiting for the oracle, which may take quite some time. This time must be measured by a clock belonging to the Turing machine and the protocol. In Beggs *et al.* (2008b), we introduced a fifth principle.

Principle 5. Combining experiments and algorithms: *Use a physical system as an oracle in a model of algorithmic computation, such as Turing machines. Determine whether the subtheory T, the experimental computation and the protocol extend the power and efficiency of the algorithmic model.*

We will now consider experiments and the way they measure physical quantities. At the heart of our conception of an experiment is an *experimental procedure* that is

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

applied to *equipment*. As we proposed in the Introduction, an experimental procedure is a sequence of instructions that act on and observe a physical system, and are to be performed step by step. More precisely, an experimental procedure is a text consisting of instructions, commands and rules derived from the physical theory T. The idea that an experiment is reproducible and subject to debate requires precise specifications and rigorous reasoning, all of which depends upon T. The analogy between experimental procedures and algorithmic procedures and programs was noticed in our earliest investigations – a language for Newtonian experimental procedures was described in Beggs and Tucker (2008).

Next, we imagine a human experimenter or technician performing an experimental procedure, that is, reading the text and following instructions, commands and rules defined by a physical theory T. The image is in all essential respects identical to Turing's analysis of a human computing with pencil and paper, following instructions, commands and rules, which is the fundamental intuition behind the Turing machine. Indeed, the image *generalises* Turing's human computer. If the T-instructions are coded as strings, the Turing machine can serve as an abstract model of the technician following the procedure. Of course, today, many experiments are fully computer controlled. These considerations lead us to introduce a new principle.

Principle 6. Algorithms controlling experiments: *Use a model of algorithmic computation, such as Turing machines, to control a physical system. Determine whether the subtheory T, the experimental procedure and equipment, and the protocol extends or limits the accuracy and efficiency of the physical experiment to make measurements.*

All four components have effects. There are many variations when modelling even simple equipment, and these may require a choice of assumptions in the theory T. Recently, we remodelled the equipment of the SME in Beggs *et al.* (2008b) and obtained dramatically different results on rounding a pointed wedge. In addition, the models of measurement introduce time in a natural quantitative way. The role of time is somewhat neglected in formal theories of measurement after Hempel (Hempel 1952); see Krantz *et al.* (2009).

Our experiments are idealised. They are designed to uncover fundamental ideas and properties of less perfect systems. Often, our results are intended to be a theoretical best case, where introducing more realism would only serve to make the results worse. Like the Turing machine, the experiments capture the essential features of examples, and map the limit of physical reality. Indeed, the ontology of our *gedankenexperimente* means that there is no fundamental difference between the ideal experiments defined by physical theory and the Turing machine.

3. The collider machine experiment

In this section and the one following it we describe an experiment that uses elastic collisions to measure the unknown (inertial) mass of a particle to arbitrary accuracy.

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

3.1. *Theory*

The elastic collision between two particles on a line is dictated by two basic laws of physics: the conservation of *linear momentum* and the conservation of *kinetic energy*. These laws are underpinned by the homogeneity of Newtonian space and time, respectively, and both can be derived from the three laws of Newtonian dynamics, which we take to be the underlying physical theory T (*cf.* Froda (1959)).

3.2. *Experimental equipment and its behaviour*

In the one dimensional collision, the centre of mass of the two particles are in the same line of motion. Let m and μ be the masses of the two particles. We will assume that the particle of 'unknown' mass μ is always at rest before the collision, and that the test particle of mass m is projected along the line towards the particle of unknown mass μ with speed $u = 1.0\ (\pm\ \varepsilon)\ ms^{-1}$, for example, with $0 \leqslant \varepsilon \leqslant 0.1$[†]. After the collision, the particle of mass m acquires the velocity v_m and the particle of mass μ is projected forward with velocity v_μ.

Since the conservation of energy reduces to the conservation of kinetic energy in the case of elastic collisions in the horizontal plane, the collision is described by the equations:

$$mu = mv_m + \mu v_\mu \tag{1}$$

$$\frac{1}{2}mu^2 = \frac{1}{2}mv_m^2 + \frac{1}{2}\mu v_\mu^2, \tag{2}$$

which can be solved for v_m and v_μ:

$$v_m = \frac{m-\mu}{m+\mu}\ u \tag{3}$$

$$v_\mu = \frac{2m}{m+\mu}\ u. \tag{4}$$

From these formulae we see that after a collision:

(a) if $m < \mu$, the test particle will move backwards;
(b) if $m > \mu$, the test particle will move forward; and
(c) if $m = \mu$, the test particle of mass m comes to rest and the particle of unknown mass μ is projected forward with the previous value of the speed of the test particle.

An experiment can be designed to measure the unknown mass μ using test particles of known mass m projected at approximately the same speed u; the procedure is given in the next section.

We will use the convention that the particle of unknown mass is placed at the origin of coordinates and points $P^- \equiv -1\ m$ and $P^+ \equiv +1\ m$ are the *flags* of the experimenter's observations: when the test particle is seen crossing the points P^- or P^+ the experiment

[†] This error margin in the initial speed of the test particle of mass m means that precision in speed does not matter for this experiment.

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

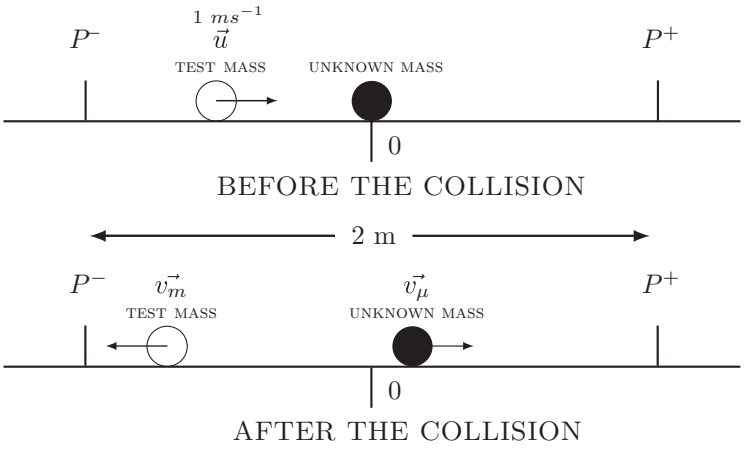


Fig. 1. COLLIDER MACHINE experiment.

terminates. If the test mass crosses the flag P^-, we have $m < \mu$ (as depicted in Figure 1), and if it crosses the flag P^+, we have $m > \mu$.

We now consider some tolerances in the design of this equipment. Amongst the properties that are largely irrelevant, or where errors can be tolerated, are:

— the (finite) distance between the two flags;
— the precision of the placement of the flags;
— the error in placing the particle of the unknown mass at the origin;
— the initial velocity of the test particle (let us say approximately 1 ms^{-1}); and
— the error in the velocity of the unknown mass.

Note that the observed velocities of the particles after the collision, that is, after crossing one or both the flags, are irrelevant.

However, properties and quantities that are relevant include:

— the one-dimensional character;
— the fact that the masses of the unknown particles are in the range $(0, 1)$;
— the fact that the particle of unknown mass μ is at rest; and
— the fact that the collisions are elastic.

Of course, these assumptions may be relaxed with various degrees of difficulty. For example, we may consider how dependent the computation is on the unknown particle being at rest at the start of each experiment, as follows. Conservation of kinetic energy requires

$$mu_m^2 + \mu u_\mu^2 = mv_m^2 + \mu v_\mu^2, \tag{5}$$

where the us and vs are the velocities of the particles before and after the collision, respectively. Conservation of linear momentum determines

$$mu_m + \mu u_\mu = mv_m + \mu v_\mu. \tag{6}$$

From these two equations we obtain

$$v_m = \frac{m+\mu}{2m}\, u_\mu + \frac{m-\mu}{2m}\, v_\mu. \tag{7}$$

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

When the unknown mass is at rest at the start of each experiment, we get

$$v_m = \frac{m-\mu}{2m}\, v_\mu. \tag{8}$$

If $m > \mu$, the proof particle moves forwards, otherwise, if $m < \mu$, the proof particle moves backwards (note that the velocity v_μ of the unknown particle after the collision is always positive, assuming it was at rest initially). Suppose now that the unknown particle can be put almost at rest, possibly with a small error, that is, moving slowly to the right ($u_\mu > 0$). The sign of the velocity of the proof mass still informs the experimenter if the following conditions hold:

— $m > \mu$:
Since both u_μ and v_μ are positive, we conclude that the sign of v_m is positive – no information is lost.

— $m < \mu$:
Since both u_μ and v_μ are positive, we conclude that the sign of v_m is preserved (that is, is negative) if

$$\frac{m+\mu}{2m}\, |u_\mu| - \frac{|\mu-m|}{2m}\, |v_\mu| < 0, \tag{9}$$

that is, if

$$(m+\mu)\, |u_\mu| < |\mu-m|\, |v_\mu|, \tag{10}$$

that is, if

$$|u_\mu| < \frac{\mu-m}{\mu+m}\, |v_\mu|. \tag{11}$$

Now, for $m \sim \mu$ and very slow initial motion u_μ, we have $v_\mu \sim u_m$. In consequence, we should have

$$u_\mu < \frac{\mu-m}{\mu+m}\, u_m, \tag{12}$$

meaning that, for suficiently large u_m, for example, with $u_m/(\mu+m) = k$,

$$u_\mu < k(\mu-m), \tag{13}$$

So we have to place the unknown mass almost at rest, with an accuracy proportional to the difference in the masses. And this is not only interesting in itself, but interesting from the computational point of view, since it introduces another exponential.

3.3. *Timing the experiment*

Looking closely at the experiment, we find an experimental barrier: the time for the test particle crossing the distance of 1 m after the collision is given by

$$t_m = \frac{1}{u}\left|\frac{m+\mu}{m-\mu}\right| = \frac{1}{u}\left|1+\frac{2\,\mu}{m-\mu}\right|, \tag{14}$$

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

so the time taken to complete the experiment is of the order

$$t_m \propto \left| \frac{1}{m-\mu} \right|. \tag{15}$$

For the values we will take for the masses and initial speed, the time is of the order of

$$\frac{A}{|m-\mu|} \leqslant t_{exp} \leqslant \frac{B}{|m-\mu|}, \tag{16}$$

for some constants A and B.

Suppose we wanted to measure the unknown mass with accuracy $|m-\mu| \leqslant 2^{-n}$. Then, substituting into inequality (16), we need exponential time

$$A.2^n \leqslant t_{exp}. \tag{17}$$

Thus, the time taken goes to infinity as the test particle's mass m approaches the unknown mass μ. If we have $m = \mu$, we will wait forever for the result.

4. Experimental procedures and protocols

To find the unknown mass, which we assume is known to lie in the interval $(0, 1)$, we need an experimental procedure. The experimental procedure we use is based on a bisection method and, ultimately, must be represented by a Turing machine. For the experimental procedure to operate the equipment through oracle queries, we must specify new mechanisms and a protocol.

4.1. *Oracle queries and setting initial conditions*

The Turing machine is connected to the collider experiment CME in the same way as it would be connected to an oracle – where we replace:

— the query state with a *shooting state* (q_s);
— the 'yes' state with a *lesser state* (q_l); and
— the 'no' state with a *greater state* (q_g).

The resulting computational device is called the *(analogue-digital) collider machine experiment*.

In order to carry out an experiment with the equipment, the machine will write a word z in the query tape and enter the shooting state. The word z codes for a dyadic rational mass m of the test particle. In the *shooting state*, the machine prepares and fires a test particle of mass m as detailed above. The experiment continues until the test particle crosses one of the flags $P^{\pm}$, and then returns to a state q_l, for $m < \mu$, or a state q_g, for $m > \mu$, of the Turing machine.

Technically, this word z will be either '1' or a binary word beginning with 0. We will use z ambiguously to denote both a word $z_1 \dots z_n \in \{1\} \cup \{0s : s \in \{0,1\}^*\}$ and the corresponding dyadic rational $\sum_{i=1}^{n} 2^{-i+1} z_i \in [0, 1]$. In this case, we write $|z|$ to denote n, that is, the size of $z_1 \dots z_n$.

We now consider the precision of the experiment. When measuring the output state, the situation is simple: either the test particle of mass m crosses P^- or it crosses P^+ (or, after

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

some timeout, no test particle is detected). Errors in observation do not arise. However, there are different postulates for the precision of the experiment, which are defined in order of decreasing strength as follows.

Definition 4.1. The CME is error-free if the mass of test particle can be set exactly to any given dyadic rational number. The CME is error-prone with arbitrary precision if the mass of the test particle can be set to within a non-zero, but arbitrarily small, dyadic precision. The CME is error-prone with fixed precision if there is a value $\varepsilon > 0$ such that the mass of test particle can be set only to within a given precision ε.

4.2. *The bisection method: basic ideas and constraints*

The Turing machine can choose and fire the test mass for the CME. For simplicity, we shall assume that the masses are set exactly, without error, according to the dyadic rational coded on the query tape; our account applies to the error-prone cases *mutato nomine*.

The bisection method for the CME is roughly as follows. We start with two dyadic rational test masses m_1 $(=\ 0\ Kg)^{\dagger}$ and m_2 $(=\ 1\ Kg)$ such that, after the collision, the test particle moves backwards and forwards, respectively. We know that the unknown mass μ is in the interval (m_1, m_2). Then we try with the mass $m = (m_1 + m_2)/2$. If m is reflected, we know that $m < \mu < m_2$ and we make the assignment $m_1 := m$, otherwise, if m continues to move forward, we know that $m_1 < \mu < m$ and we make the assignment $m_2 := m$. In principle, we will get *a binary digit of the unknown mass per collision.*

However, there are complications. There are potential problems in the case where the unknown mass is dyadic, and there is also the matter of timing. As the method proceeds, we will have dyadic fractions that are described by longer and longer words. It is reasonable that setting up the apparatus for longer words would require more time (for example, the increased time just to read the word). There are physical reasons inherent in the CME determining why some experiments take longer times than others.

As the Turing machine does not know the value of μ, it has no way of determining how long an experiment will take. So the following questions arise:

— How does the programmer of the Turing machine deal with this problem?
— Should the Turing machine wait indefinitely, or should the programmer employ a timer or schedule and abort the experiment after a certain time?
— If we have a 'time out' in an experiment, does the Turing machine program halt, or does it try to 'deal' with the time out?

4.3. *Time schedules and the collider machine protocol*

For the CME there are definite physical restrictions on choosing its protocol: there are constants $K, N > 0$ such that for a query z of length $|z|$, giving a dyadic rational z that

[†] Although this has no physical meaning in the setting of classical mechanics, this does not matter, as will soon become apparent.

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

determines the test mass m, the time taken to return a result is

$$K/|\mu - z| \pm N.$$

The machine returns one result if $\mu > z$, and another if $\mu < z$. If $\mu = z$, we get no result, that is, we wait an infinite amount of time, unless the experiment is timed out by a process in the Turing machine. Such a time limit would mean that no result would be given for z sufficiently close to μ, not just the case where $z = \mu$. We consider two cases:

(1) *There is no time limit on the oracle consultation.* After making a query z, the Turing machine will wait until it receives an interrupt from the collider machine with the result, which is either $\mu > z$ or $\mu < z$. At this time, the calculation will resume. It is important to note that, in this case, the Turing machine has no idea of the elapsed time and that the Turing machine may wait indefinitely.

(2) *There is a time limit on the oracle consultation.* We assume the time limit has the form a function $T(|z|)$, where z is the query. We use this as a timer in the following way. When the query z is made, the Turing machine starts a timer that counts time $T(|z|)$, and then the Turing machine looks for an answer from the collider machine, which may or may not have reached a decision by that time. The possible results are $\mu > z$, $\mu < z$ or *out of time*. Note that the Turing machine must wait for the whole time $T(|z|)$. Also note that if $z = 1/2$, for example, the Turing machine can increase the waiting time just by adding trailing 0s to z, a padding that does not affect its numerical value, but affects its word length $|z|$[†].

It is reasonable to assume that the reason for increasing the word length of the query is to find the mass μ more accurately. But the time taken to find μ to more places increases at least exponentially in the number of places. If we do not allow for an increase in time that is at least exponential in the word length, we have no hope of finding μ to that length.

It is important to note that there are two possible reactions by the Turing machine to an experiment having timed out, and both of these will be considered later in the paper:

(a) The computation is aborted. This will be used in the definition of measurability later.

(b) The computation is continued, with a result of 'time out' being returned by the physical oracle. This will be used in the fixed precision results later.

4.4. *The bisection method: experimental procedure*

The function T specifies a timer or schedule, that is, in each experiment, in order to read the $|m|$-bit of the mass μ, $T(|m|)$ gives the number of time steps that the experimenter is prepared to wait before abandoning the experimental run.

[†] Padding buys time, but a polynomial protocol could not gain sufficient time for the CME by padding: to perform the experiment it would need exponential size padding, which would require exponential time.

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

Bisection(T) – an experimental procedure to read the first n bits of an unknown mass μ with schedule T:

1 **input** n – this is the required precision coded by the number of places to the right of the left leading 0;
2 $m_1 := 0$, $m_2 := 1$, $m := 0$ – these are the initial values with no physical significance (note that $|m_1| = 0$, $|m_2| = 1$ and $|m| = 0$);
3 **while** $|m| \leqslant n$ **do**
  (a) $m := (m_1 + m_2)/2$;
  (b) place the particle of unknown mass $\mu \in (0, 1)$ at the origin;
  (c) project test particle of mass m to collide with particle of unknown mass;
  (d) **if** test particle crosses the flag P^- in time $T(|m|)$, **then** $m_1 := m$; append 1 – this means it is known that $\mu \in (m, m_2)$;
  (e) **if** test particle crosses the flag P^+ in time $T(|m|)$, **then** $m_2 := m$; append 0; – this means it is known that $\mu \in (m_1, m)$;
  (f) **if** no particle crosses the flags in time $T(|m|)$, **then** return "time out";
4 **end while**;
5 **output** dyadic rational denoted by m.

The bisection method is parameterised by the time schedule T given to the experimenter to test for the crossing of one of the flags, and it applies to each type of precision. There is an essential constraint on T, which has fundamental consequences.

Proposition 4.2. At the nth stage of the bisection algorithm, the lower bound on the time of a single experiment with the CME is exponential in n.

Proof. At the stage of the bisection algorithm where we try to determine the nth place of μ, we need to employ a test particle of mass m where $|m - \mu| \leqslant 2^{-n}$. Then equation (17) gives $A.2^n \leqslant t_{exp}$. □

Thus, we have the following consequence.

Proposition 4.3. The protocol that processes queries between a Turing machine and the collider takes time that is at least exponential in the size of the mass of the test particle specified by the queries.

Finally, there are some observations of properties that are experimentally undecidable.

Proposition 4.4. That the test mass coincides with the given unknown mass cannot be established experimentally in finite time by the CME under any experimental procedure.

Proof. According to Equation 14, as $m \to \mu$ through the bisection method, the time the experimenter has to wait goes to infinity, $t_{exp} \to +\infty$. If the two masses coincide, then the experimenter will never know. □

Proposition 4.5. To know if the unknown mass is a dyadic rational cannot be established experimentally in finite time by the CME under any experimental procedure.

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

4.5. *Computation using CME as an oracle*

Here are the definitions for deciding sets by Turing machines with the help of the CME (recall Beggs *et al.* (2008b)).

Definition 4.6. Let $A \subseteq \Sigma^*$ be a set of words over Σ. We say that an error-free analogue-digital collider machine M *decides* A if there exists a time constructible schedule T for a protocol to operate the oracle CME(μ) such that, for every input $w \in \Sigma^*$, w is accepted if $w \in A$ and rejected when $w \notin A$. We say that M *decides* A *in polynomial time* if M decides A and there is a polynomial p such that, for every $w \in \Sigma^*$, the number of steps of the computation is bounded by $p(|w|)$.

Definition 4.7. Let $A \subseteq \Sigma^*$ be a set of words over Σ. We say that an error-prone analogue-digital scatter machine M *decides* A if there exists a time constructible schedule T for a protocol to operate the oracle CME(μ) and a number $\gamma < 1/2$, such that the error probability of M for any input w is smaller than γ. We say those computations that correctly accept or reject the input are *correct*. We say that M *decides* A *in polynomial time* if M decides A and there is a polynomial p such that, for every input $w \in \Sigma^*$, the number of steps in every computation of M on w is bounded by $p(|w|)$.

5. Preliminaries on non-uniform complexity

In this paper Σ denotes an *alphabet* and Σ^* denotes the *set of finite words* over Σ. A *language* is a subset of Σ^*. We will almost always adopt the binary alphabet $\{0, 1\}$.

The *pairing function* is the well-known map $\langle -, - \rangle : \Sigma^* \times \Sigma^* \to \Sigma^*$, computable in linear time, that allows us to encode two words in a single word over the same alphabet by duplicating bits and inserting a separation symbol '01'.

We now recall the definition of a *non-uniform complexity class*. We use the term *advice function* to mean any total map $f : \mathbb{N} \to \Sigma^*$.

Definition 5.1. Let $\mathscr{B}$ be a class of sets and $\mathscr{F}$ be a class of advice functions. We define the new class $\mathscr{B}/\mathscr{F}$ as the class of sets A such that there exists a set $B \in \mathscr{B}$ and an advice function $f \in \mathscr{F}$ such that, for every word $x \in \Sigma^*$, $x \in A$ if and only if $\langle x, f(|x|) \rangle \in B$.

Suppose we fix the class $\mathscr{B}$ to be the class P of sets decidable by Turing machines in polynomial time. Then we still have one degree of freedom, which is the class of advice functions $\mathscr{F}$ that makes $P/\mathscr{F}$. In this paper, we will work with subpolynomial advice functions, that is, $\mathscr{F}$ is a class of functions with sizes bounded by *polynomials* and computable in polynomial time. Note that the advice functions are not, in general, computable but the associated bounds are computable; for example, if the class is *poly*, then any, possibly non-computable, advice function $f : \mathbb{N} \to \Sigma^*$ is bounded by a (computable) polynomial p such that, for all $n \in \mathbb{N}$, $|f(n)| \leqslant p(n)$.

5.1. *Deterministic classes*

Although the class $\mathscr{F}$ of functions is arbitrary, there is no point in using functions with growth rate greater than exponential. Let *exp* be the set of advice functions bounded

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

in size by functions in the class $2^{O(n)}$. Then P/exp contains all sets. Given this fact, we consider whether either $P/poly$ or P/log (subclasses of P/exp) exhibit some interesting internal structure. The following three main results should be recalled from Balcázar *et al.* (1988, Chapter 5).

Proposition 5.2. There exist sets in $EXPSPACE$ not in $P/poly$.

Here $EXPSPACE$ uses a Turing machine whose working tape, rather than being infinite, has length bounded by an exponential on the size of the input, and there is no bound on time. This proposition is proved by a non-trivial diagonalisation of $P/poly$.

Proposition 5.3. The *Halting Set* $H = \{0^n : \text{Turing machine with code } n \text{ halts on } 0\}$ is in $P/poly$.

We will also need to treat prefix non-uniform complexity classes. For these classes we may only use prefix functions, that is, functions f such that $f(n)$ is always a prefix of $f(n+1)$. The idea behind prefix non-uniform complexity classes is that the advice given for inputs of size n may also be used to decide smaller inputs.

Definition 5.4. Let $\mathscr{B}$ be a class of sets and $\mathscr{F}$ be a class of advice functions. The prefix advice class $\mathscr{B}/\mathscr{F}*$ is the class of sets A for which some $B \in \mathscr{B}$ and some prefix function $f \in \mathscr{F}$ are such that, for every length n and input w with $|w| \leqslant n$, we have $w \in A$ if and only if $\langle w, f(n)\rangle \in B$.

Non-uniform classes are, indeed, relevant because of their impact in characterisation theorems. For example, the class $P/poly$ is the class of sets decidable by the families of polynomial size circuits. They have been used in many theories of analogue computation and hybrid systems (for example, see Siegelmann (1999) and Bournez and Cosnard (1996)).

5.2. *Probabilistic classes*

In the context of probabilistic complexity classes, it is a matter of some controversy whether Definition 5.1 is the appropriate definition of a non-uniform probabilistic class (for example, of $BPP/\mathscr{F}$). Notice that by demanding that there is a set $B \in BPP$ and a function $f \in \mathscr{F}$ (in this order!) such that $w \in A$ if and only if $\langle w, f(|w|)\rangle \in B$, we are demanding a fixed probability $1/2 + \varepsilon$, $0 < \varepsilon < 1/2$ (fixed by the Turing machine chosen to witness that $B \in BPP$) *for any possible correct advice*, instead of the more intuitive idea that the error $\gamma = 1/2 - \varepsilon$ only has to be bounded after choosing the correct advice. This leads to the following definitions for the specific complexity class BPP, which we use throughout this paper.

Definition 5.5. $BPP//poly$ is the class of sets A for which a probabilistic Turing machine TM clocked in polynomial time, a function $f \in poly$ and a constant $0 < \gamma < 1/2$ exist such that TM rejects $\langle w, f(|w|)\rangle$ with probability at most γ if $w \in A$, and accepts $\langle w, f(|w|)\rangle$ with probability at most γ if $w \notin A$.

Definition 5.6. $BPP//log*$ is the class of sets A for which a probabilistic Turing machine TM clocked in polynomial time, a prefix function $f \in log*$ and a constant $0 < \gamma < 1/2$

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

exist such that, for every length n and input w with $|w| \leqslant n$, TM 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 $w \notin A$.

It can be shown that $BPP//poly = BPP/poly$, but it is unknown whether $BPP//log* \subseteq BPP/log*$. Following Balcázar and Hermo (1998), we can assume without loss of generality that, for $P/log*$ and $BPP//log*$, the length of any advice $f(n)$ is exactly $\lfloor a \log n + b \rfloor$ for some $a, b \in \mathbb{N}$ that depend on f. The proof also applies to BPP//log*.

6. The COLLIDER computes $P/log*$ and $BPP//log*$

6.1. *Coding the advice function as the unknown mass value*

The aim is to code advice functions as quantities. We take a binary coding $c(a)$ for words a by first converting a to a string of 0s and 1s using a binary code for every letter in its alphabet Σ (for example, using Unicode or ASCII). For example, we might suppose that the binary form of a was 00101. To find $c(a)$, we replace every 0 in its binary form by 100 and every 1 by 010. Thus our example becomes 100100010100010. If a is empty, then so is $c(a)$. Now we must decide on a coding for log* advice, where, recall, the advice function $f(n)$ can be used to decide all words w with $|w| \leqslant n$. We can recursively define $\mu(f)$ as the limit of $\mu(f)(n)$, using our coding c, starting with

$$\mu(f)(0) \;=\; 0 \cdot c(f(0))$$

and using the following cases:

$$\mu(f)(n+1) \;=\; \begin{cases} \mu(f)(n)\, c(s) & \text{if } n+1 \text{ not a power of 2 and } f(n+1) = f(n)s \\ \mu(f)(n)\, c(s)\, 001 & \text{if } n+1 \text{ a power of 2 and } f(n+1) = f(n)s. \end{cases}$$

The net effect is to add a 001 at the end of the codes for $n = 1, 2, 4, 8 \ldots$

Now take w with $2^{m-1} < |w| \leqslant 2^m$. We read the binary expansion until we have counted a total of $m+1$ 001 triples. Now the extra 001s are deleted, and we can reconstruct $f(2^m)$, which can be used as the advice function for w. But how many binary digits from $\mu(f)$ must we read in order to reconstruct $f(2^m)$? We start with $|c(f(n))| \leqslant L\ \log_2(n) + K$ (for some constants K and L) from the definition of log length. We must read at most $L\,m + K + 3\,m$ digits when we add in the separators, so the number of digits is logarithmic in $|w|$.

Of course, not every element of $(0, 1)$ occurs as a $\mu(f)$. Note that no dyadic rational can occur, as they would have to end in 0^ω or 1^ω, and the triples 000 and 111 do not occur in any $\mu(f)$. We will need a stronger result about which numbers cannot be of the form $\mu(f)$.

Proposition 6.1. Let $f : \mathbb{N} \to \Sigma^*$ be an advice function. Given a dyadic rational $k/2^n \in (0, 1)$, for integer k, $|\mu(f) - k/2^n| > 1/2^{n+5}$, for all f.

Proof. The statement is a reasonably simple consequence of the fact that $\mu(f)$ has a binary expansion consisting only of triples of the form 001 or 100 or 010. □

Of course, this coding technique also gives us the following proposition.

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

Proposition 6.2. Given that the unknown mass μ is of the form $\mu(f)$ for some advice function f, at any step of the *bisection method*, the lower and the upper bounds on the time of a single experiment are exponential in the word length.

Why can we not just use the class log? The collider machine experiment allows us to read only $log(n)$ bits of the unknown mass in polynomial time because of the exponential time delay in the protocol. Assuming that f is an advice function in log, to encode the advice information, for $n = 0$ to $n = p$, we need $f(0), \ldots, f(p)$. A way to encode this information requires the concatenation of all these sequences, which amounts to a number of bits of the order of $n\log(n)$, which is greater than logarithmic size. This problem is avoided with the prefix advice $log*$.

6.2. *The error-free* COLLIDER *can decide* $P/log*$ *in polynomial time*

In this subsection we will show that error-free colliders with an exponential AD-protocol can decide all the sets in $P/log*$ in polynomial time. We leave the more complicated problem of determining an upper bound for the complexity of the sets that these machines can decide in polynomial time (see Beggs *et al.* (2009c)).

Let A be a set in $P/log*$, and, by definition, let $B \in P$, $f \in log*$ be such that

$$w \in A \iff \langle w, f(|w|)\rangle \in B.$$

Now take an AD-collider with the unknown mass set to $\mu(f)$, as described in Subsection 6.1, and exponential time protocol. For a word w, if we can determine the advice $f(|w|)$ in polynomial time in $|w|$, then the Turing machine can determine $\langle w, f(|w|)\rangle$ in polynomial time, so the decision problem $w \in A$ can be solved in polynomial time in $|w|$. In turn, to determine $f(|w|)$, it suffices to show that we can read the first $log(n)$ binary places of the unknown mass $\mu(f)$ in polynomial time in n.

Proposition 6.3. An error-free analogue-digital collider that uses an exponential AD-protocol can determine the first $log(n)$ binary places of the unknown mass $\mu(f)$ in polynomial time in n, where n is the size of the input.

Proof. We use the *bisection method* with the exponential protocol, which requires $k' \times log(n)$ collisions for some fixed constant k', with collision i requiring time $O(2^{k''i}) \subseteq O(2^{k \times log(n)}) = O(n^k)$, with $k = k'' \times k'$. Adding all these times gives a total time polynomial in n. □

Theorem 6.4. An error-free analogue-digital collider with an exponential AD-protocol can decide $P/log*$ in polynomial time.

Proof. This result follows from the discussion earlier in this subsection together with Proposition 6.3. □

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

6.3. *The error-prone* COLLIDER *with arbitrary precision can decide* $P/log*$ *in polynomial time*

Now we come to the error-prone arbitrary precision case, which is solved in almost exactly the same way. The work lies in choosing the errors so that the same process actually works, and for that we need Proposition 6.1.

Proposition 6.5. An error-prone arbitrary precision analogue-digital collider with a strictly exponential AD-protocol can determine the first $log(n)$ binary places of the unknown mass $\mu(f)$ in polynomial time in n, where n is the size of the input.

Proof. We use the *bisection algorithm* again. At the ith stage in the bisection process (involving dyadic rationals with denominators 2^i), we set the error in the mass $k/2^i$ of the test particle to be $1/2^{i+6}$, that is, such that the mass of the test particle lies in the interval $[k/2^i - 1/2^{i+6}, k/2^i + 1/2^{i+6}]$. By Proposition 6.1, the unknown mass cannot be in the error interval about the given dyadic rational, and thus the result of the experiment is the same as if the mass of the test particle was the given dyadic rational $k/2^i$. Also by Proposition 6.1, the distance between the test mass and μ is at least $1/2^{i+6}$, so the experimental time is bounded by an exponential in i. Thus, the first $log(n)$ binary places of the unknown mass $\mu(f)$ can be read in polynomial time in n. □

Theorem 6.6. An error-prone arbitrary precision analogue-digital collider with an exponential AD-protocol can decide $P/log*$ in polynomial time.

Proof. This result follows from the discussion earlier in this subsection together with Proposition 6.5. □

Note that from the point of view of the complexity classes considered so far, there is no difference between *infinite precision* and *unbounded precision*. One may think that infinite precision is needed to achieve the power of such non-uniform complexity classes, but as the analysis in Beggs *et al.* (2008b) and the current paper shows, no such difference exists.

6.4. *The error-prone* COLLIDER *with fixed* a priori *precision can decide* $BPP//log*$ *in polynomial time*

The mass of the test particle is set at some dyadic rational m up to some dyadic fixed precision ε. We will show that such machines may, in polynomial time, make probabilistic guesses of up to a log-number of digits of the value of the unknown mass. We will then conclude that these collider machines decide $BPP//\log*$. It is important to note that these machines are allowed to keep running after an experiment times out – the result 'time out' is returned by the physical oracle.

Proposition 6.7. For any real value $\delta < 1/2$ and prefix function $f \in \log*$, there is an error-prone analogue-digital collider machine with fixed precision that obtains $f(n)$ in polynomial time with a probability of error of at most δ.

Proof. We encode the prefix function $f \in \log*$ as a real number $s \in (0,1)$ using the method of Section 6.1. The problem of obtaining $f(n)$ in polynomial time becomes the

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

problem of reading the first $\lfloor a \log n + b \rfloor$ digits of s in polynomial time (for some constants a, b).

Suppose that ε is the *a priori* fixed error in measuring the mass of the proof particle. We then set the unknown mass of our analogue-digital collider machine at the value $\mu = 1/2 - \varepsilon/2 + s\varepsilon$, so that $\mu \in [1/2 - \varepsilon/2, 1/2 + \varepsilon/2]$. We set a time limit T on each experiment such that if $|m - \mu| \geqslant \varepsilon/4$, we are guaranteed a result in time T. Then there is an $\eta \in (0, \varepsilon/4)$ such that the experiment runs out of time (exceeds the bound T) on the interval $(\mu - \eta, \mu + \eta)$ – the exact value of η is not important, what is important is that the interval is symmetric about μ. (In this proof we will ignore the end points of intervals since they occur as random values of the test mass with probability zero.)

Our method for guessing digits of s begins by commanding the collider to shoot proof particles with mass $m = 1/2$ ($\pm\ \varepsilon$) a number ζ times. Then we get the following results with the stated probabilities:

(1) $m < \mu$, with probability $p = (\mu - \eta + \varepsilon - \frac{1}{2})/(2\varepsilon)$ – this occurs a random number α times.
(2) $m > \mu$, with probability $q = (\frac{1}{2} + \varepsilon - \mu - \eta)/(2\varepsilon)$ – this occurs a random number β times.
(3) Out of time, with probability $r = 2\eta/(2\varepsilon)$ – this occurs a random number γ times.

By our assumption of independent uniform distribution for m in the interval $[1/2 - \varepsilon, 1/2 + \varepsilon]$, the resulting distribution is a trinomial with probability for (α, β, γ) being

$$\frac{p^\alpha\ q^\beta\ r^\gamma\ \zeta!}{\alpha!\ \beta!\ \gamma!}, \quad \alpha + \beta + \gamma = \zeta.$$

Now we consider the random variable $X = 2\alpha + \gamma$, which is a combination chosen to cancel the number η from the mean. This is fairly easily seen to have mean $\bar{X} = \zeta\,(\mu + \varepsilon - \frac{1}{2})/\varepsilon$, but its variance $\mathrm{Var}(X)$ is a little more difficult to find:

$$\begin{aligned} \mathrm{Var}(X) &= 4\,\mathrm{Var}(\alpha) + \mathrm{Var}(\gamma) + 4\,\mathrm{Covar}(\alpha, \gamma) \\ &= 4\,\mathrm{Var}(\alpha) + \mathrm{Var}(\gamma) + 4\,(\mathbb{E}(\alpha\gamma) - \bar{\alpha}\,\bar{\gamma}). \end{aligned}$$

A little calculation yields the expectation

$$\mathbb{E}(\alpha\gamma) = \zeta\,(\zeta - 1)\,p\,r,$$

and substituting in

$$\begin{aligned} \bar{\alpha} &= \zeta\,p \\ \bar{\gamma} &= \zeta\,r \\ \mathrm{Var}(\alpha) &= \zeta\,p(1 - p) \\ \mathrm{Var}(\gamma) &= \zeta\,r(1 - r) \end{aligned}$$

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

gives

$$\begin{aligned}\mathrm{Var}(X) &= 4\,\zeta\,p(1-p) + \zeta\,r(1-r) - 4\,\zeta\,p\,r \\ &= \zeta\,\frac{3\,\varepsilon + 4\,s\,\varepsilon - 4\,s^2\,\varepsilon - 4\,\eta}{4\,\varepsilon} \\ &\leqslant \zeta\,\frac{2\,\varepsilon + 4\,s\,(1-s)\,\varepsilon}{4\,\varepsilon} \leqslant \frac{3\,\zeta}{4}.\end{aligned}$$

Now we use Chebyshev's inequality, which says that for every $\Delta > 0$,

$$\mathbb{P}(|\bar{X} - X| > \Delta) \leqslant \frac{\mathrm{Var}(X)}{\Delta^2}.$$

Putting $\Delta = x\,\zeta$ here gives

$$\mathbb{P}\left(\left|\frac{1}{2} + s - \frac{X}{\zeta}\right| > x\right) \leqslant \frac{\mathrm{Var}(X)}{x^2\,\zeta^2} \leqslant \frac{3}{4\,x^2\,\zeta}.$$

To read the kth binary place of s, according to the coding, it is sufficient to find s accurate to a value of $1/2^{k+5}$, and then the probability of error is

$$\mathbb{P}\left(\left|\frac{1}{2} + s - \frac{X}{\zeta}\right| > 1/2^{k+5}\right) \leqslant \frac{3\;2^{2k+10}}{4\,\zeta}.$$

To do this within probability of error δ, we need a number of experiments

$$\zeta > \frac{3\;2^{2k+10}}{4\,\delta}.$$

As k is logarithmic in n, the result is polynomial time in n. □

Proposition 6.7 will guarantee us that for every fixed error ε we can find an unknown mass that will allow a CME to extract the desired information. It does not state that we can make use of any unknown mass independently of the fixed error ε. It can be shown, however, that if ε is a dyadic rational, then we may guess $O(\log(n))$ digits of the unknown mass in polynomial time.

Theorem 6.8. An error-prone analogue-digital collider with an exponential AD-protocol can decide $BPP//\log*$ in polynomial time.

Proof. The statement is a consequence of the definition of the class $BPP//\log*$, taking into consideration the result of Proposition 6.7 – see Beggs *et al.* (2008b) for details. There is one minor modification to take account of the possibility of a 'time out' result in an experiment. In the proof that we can (up to a given probability of failure) simulate a sequence of independent coin tosses, we need two mutually exclusive events whose union has probability one. To achieve this, we use the test particle crossing a given flag as one event, and the union of crossing the other flag and 'time out' as the other event. That is, we amalgamate two of the possibilities listed in the proof of Proposition 6.7. □

Measurements based on this more realistic assumption about precision lead to an apparent increase in power from $P/log*$ to $BPP//\log*$. This is because of the introduction of probabilities. (It is apparent because $P \subseteq BPP$, but we do not know if

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

$P = BPP$.) Since the class $BPP//\log *$ contains non-computable sets, the machines can decide super-Turing languages.

7. Measurability by the CME

Fundamentally, our experiment CME tries to find an unknown mass μ by a sequence of comparisons with known masses m, which are rational multiples of some standard mass M. The aim is, for any given n, to find the first n binary places of the number μ/M. In the rest of the paper, we assume for convenience that $M = 1$. The time $T(|m|)$ taken for one run of the experiment using known mass m is governed by an inequality $T(|m|) \geqslant K/|m - \mu|$.

There are good and bad values of μ that place limitations on our experimental method. In the worst case, μ has dyadic rational values. Here, any run of the experiment with $m = \mu$ will fail to give a result, and we will be left with μ being strictly inside a dyadic interval. But no matter how small the interval is, we will only be able to determine a fixed number of binary places. For example, if $\mu = 0{\cdot}001$ exactly, we will never be able to distinguish between $0{\cdot}001$ and $0{\cdot}00011\ldots10$, where the 1s are repeated sufficiently many times before reaching a 0.

Let us formulate precisely what we mean when we say a mass is measurable.

Definition 7.1. A mass μ is said to be *measurable* by the CME if there exists a Turing machine M, equipped with a computable schedule T, such that it prints the first n bits of μ on the output tape in less than $T(n)$ time steps without timing out in any query. Similarly, it is *feasibly measurable* if T can be chosen to be time constructible (recall Balcázar *et al.* (1988)).

Note the importance of time constructibility here – it allows the timing to be done by the Turing machine itself in real time, rather than by employing another device as a timer. At another extreme, were we to allow T to be non-computable, then any non-dyadic rational would be measurable.

What can be measured depends on the experimental procedure chosen to run the experiment. The following concept will prove useful.

Definition 7.2. A Turing machine M is said to be a *universal measuring procedure* for the CME if, for every measurable mass μ, there exists a computable schedule T, such that M equipped with T measures μ.

7.1. *Most masses are measurable*

Here we examine some measurable masses, and show that, in the sense of measure theory, almost all masses are measurable.

Lemma 7.3. Suppose $\mu \in [0, 1]$, and that the time taken to determine if $|m - \mu| < \epsilon$ is K/ϵ. We define a series of algorithms E_r, for $r \in \mathbb{N}$, for the Turing machine using the

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

CME as oracle, as follows:

(a) for all $0 \leqslant p \leqslant 2^r$, fire a particle of mass $p/2^r$, using waiting time $K\,2^{2r+1}$; and thus
(b) the experimental calls in E_r take a total time $K\,2^{3r+1}$.

Then the measure of the set where algorithm E_r fails to find μ to r binary places is $\leqslant 2^{-r}$.

Proof. A set $F_r \subset [0,1]$, of length $\leqslant 2^{-r}$, is defined by

$$F_r = \bigcup_{0 \leqslant p \leqslant 2^r} \left[\frac{p}{2^r} - \frac{1}{2^{2r+1}}, \frac{p}{2^r} + \frac{1}{2^{2r+1}} \right] \cap [0,1].$$

If x is not in the set F_r, the algorithm E_r will determine the first r binary places of x. □

Corollary 7.4. There are programs P_k (for integer $k \geqslant 1$), with specified waiting times, such that there is a measure one set in $[0,1]$ such that for all μ in this set, there is a k such that, for all $n \geqslant 0$, we have P_k will successfully determine the first n binary digits of μ.

Proof. To find the first n places of μ, P_k uses the algorithm E_{n+k} as described in Proposition 7.3. The set where P_k may fail to find the first n places is F_{n+k}, so the set where P_k may fail for some n is $\cup_{n \geqslant 1} F_{n+k}$, and this set has measure $\leqslant 2^{-k}$. The set where, for all k, there is an n such that P_k may fail for that n is the measure zero set $\cap_{k \geqslant 1} \cup_{n \geqslant 1} F_{n+k}$. □

To emphasise the result of Corollary 7.4, if we choose $\mu \in [0,1]$ at random (with a uniform probability distribution), then μ will be measurable with probability one.

The following result is rather trivial, but it is important to point it out.

Proposition 7.5. There are programs N_k (with integer $k \geqslant 1$), with specified waiting times (say T_k), such that for any non-dyadic $\mu \in [0,1]$ and any $n \geqslant 0$, there is a k such that program N_k will find the first n binary places of μ.

Proof. The program N_k runs the bisection procedure with a waiting time of T_k for each experiment. Provided $\mu \in [0,1]$ is not dyadic, there is some accuracy of measurement that will determine its first n binary places. □

It is important to note that Propositions 7.4 and 7.5 prove the existence of programs to find the first n places; there is no idea of being able to find in advance which programs they are. Note, too, that the infinite sequence T_1, T_2, ..., T_k, ..., where T_k is the time needed to find experimentally the k-bit of μ is not recursively enumerable. The Proposition 7.5 can be read in the following way: *if we know the unknown mass μ in advance – as an oracle – then we can define a schedule of computation times for each bit of the mass.*

Now we consider numbers that are easy to find using the bisection method. Recall that a real number is algebraic of order k if it is a root of a polynomial of order k with integer coefficients. The following result, due to Liouville, is well known.

Proposition 7.6. If x is an algebraic number of order k, then, for all non-zero integers a and b such that $x \neq a/b$, there is a computable number $R(x) > 0$ such that

$$\left| x - \frac{a}{b} \right| \geqslant \frac{R(x)}{b^k}.$$

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

Proposition 7.7. If $\mu \in [0,1]$ is an algebraic number, and not a dyadic rational, then there is a procedure to find μ such that the time schedule $T(n)$ to find the nth bit of μ is $\alpha\, n\, 2^{kn}$ for some computable constant α.

Proof. Using the bisection procedure, we need $O(n)$ experiments, each taking time proportional to 2^{kn}. □

7.2. *A characterisation of measurable masses*

We can characterise masses that can be measured quite precisely – non-dyadic masses $\mu \in [0,1]$ can be written in the pattern form, where u_k gives the number of digits in the kth group:

$$\mu = 0{\cdot}\underbrace{1\ldots1}_{u_1}\underbrace{0\ldots0}_{u_2}\underbrace{1\ldots1}_{u_3}\underbrace{0\ldots0}_{u_4}\underbrace{1\ldots1}_{u_5}\underbrace{0\ldots0}_{u_6}\ldots \quad \text{where } u_1 \geqslant 0,\ u_i \geqslant 1\ (i \geqslant 2). \tag{18}$$

Proposition 7.8. For the CME with unknown mass μ (which is not a dyadic rational), written according to the pattern (18):

(1) If μ is measurable by any program, then the sequence u_k is bounded by a computable function.
(2) If the sequence u_k is bounded by a computable function, then μ is measurable by the bisection method.

Proof. First note that the digit at the end of the block labelled by u_k is in the a_kth position, where $a_k = u_1 + \cdots + u_k$. To make the method obvious, we use an example with $u_1 = 3$, $u_2 = 2$, $u_3 = 4$, $u_4 = 3$, and so on.

$$\mu = 0{\cdot}11100111100011\ldots$$

To determine all digits up to the $a_3 = 9$th digit, any program must have successfully run the experiment with test masses in the intervals $[\mu^-, \mu)$ and $(\mu, \mu^+]$, where $\mu^\pm$ are the a_3 digit dyadic rationals, differing only in the last position:

$$\mu^- = 0{\cdot}111001110\ , \quad \mu^+ = 0{\cdot}111001111.$$

So we have the inequalities

$$2^{-a_3} \leqslant |\mu - \mu^-| \leqslant 2^{1-a_3}\ , \quad 2^{-a_4-1} \leqslant |\mu - \mu^+| \leqslant 2^{-a_4}.$$

(1) For the first a_k digits of μ to be determined, we must perform at least one experiment of duration at least $2^{a_{k+1}}K$, where K is a constant. If μ is measurable, there must be a computable function T such that $T(a_k) \geqslant 2^{a_{k+1}}K$, and from this we determine the following formula, from which a computable bound for the sequence u_k can be derived:

$$2^{u_{k+1}} \ \leqslant\ 2^{-a_k}\, T(a_k)/K.$$

(2) Note that in the previous discussion, the numbers $\mu^\pm$ are the last two numbers queried in the binomial method for finding the first a_3 digits of μ. In general, to find the first

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

a_k digits of μ by the binomial method, we need a_k experiments, each of duration at most $2^{a_{k+1}+1}K$. If the sequence u_n is computable, this gives a computable schedule for finding μ. □

Corollary 7.9. The Turing machine equipped with the bisection algorithm is a universal measuring procedure (see Definition 7.2).

7.3. *Results on non-measurable masses*

Now let us consider some further numbers that are difficult to find using our method. For convenience, we recall the specification of a non-computable function beaver : $\mathbb{N} \to \mathbb{N}$ called the *busy beaver function* (*cf.* Cooper (2004)).

Definition 7.10. Let f, $g : \mathbb{N} \to \mathbb{N}$ be total functions. We say that g *dominates* f if there exists a natural number p, called an order, such that, for all n such that $n > p$, we have $g(n) > f(n)$. If $\mathscr{F}$ is a set of such functions, we say that g *dominates* $\mathscr{F}$ if g dominates f for all $f \in \mathscr{F}$.

Definition 7.11. Let *beaver* : $\mathbb{N} \to \mathbb{N}$ be the total function defined by:

— *beaver*(0) = 0 by convention;
— *beaver*(n) is the maximum output for input 0 among all Turing machines with n states that halt on input 0.

The *beaver* is a totally defined function because for all n there exists a Turing machine with n states that halts and, consequently, can produce a string of 1s on the output tape that can be interpreted in unary or binary, according to convention.

This function is due to Tibor Radó (Rado 1962), and was one of the first well-defined non-computable total functions. Unsurprisingly, the function is complicated and the growth of the function is considered an open problem (Brady 1994).

Theorem 7.12. The function *beaver* dominates all total computable functions.

Note that for every total computable function f there in an order p_f, depending on the function f, such that from $p_f + 1$ onward, the busy beaver grows faster than f.

Theorem 7.13. There are uncountably many values $\mu \in [0, 1]$ that are not measurable by the CME.

Proof. We take all μ, defined by Pattern 18, made from all possible choices of the following values of each k:

$$u_k = \begin{cases} \text{beaver}(k) \\ \text{beaver}(k) + 1. \end{cases}$$

Because of the choice, there are uncountably many such μ. By Proposition 7.8, all of them are non-measurable. □

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

Can we decide if a mass is measurable? To be more precise, we could imagine using a program on a Turing machine using the CME as oracle to decide this. However, we have the following negative result.

Proposition 7.14. There is no program running on a Turing machine using the CME as oracle that can decide in finite time if a mass is measurable.

Proof. In the given amount of time the program runs for, the CME can only find the mass μ within a given open interval. Any mass within this open interval would produce the same result for the program. However, this interval contains (after some initial segment of μ) both masses with infinite endings of their decimal expansions of the form coded earlier (and therefore effectively measurable) and those with busy beaver endings (and therefore not effectively measurable). □

8. An uncertainty principle in classical mechanics

Using an algorithmic theory of measurement, we have shown that for an experiment in classical dynamics, what is measurable depends on:

(1) *Time*: To buy accuracy you have to pay with time, and the budget for time is controlled by schedules.
(2) *Equipment*: The apparatus for the CME, and the physical theory that governs it, imposes its own limits on how much time is required for a given experiment.
(3) *Procedure*: For the CME there is a universal experimental procedure (the bisection method), which can obtain all the results that can be measured by all experimental procedures.

The limitation on the apparatus in (2) can be phrased as a sort of uncertainty principle, using equation (14) to make an inequality with $\Delta\mu = |m - \mu|$ being the uncertainty in mass, and Δt being the time necessary to perform an experiment, where u is the input velocity and r is the distance from the unknown mass to the flags:

$$\frac{\text{lower bound on masses} \times r}{u} \leqslant \Delta\mu \times \Delta t \leqslant \frac{2 \text{ upper bound on masses} \times \text{r}}{u}.$$

$\Delta\mu \times \Delta t$ is a product of the type $\Delta E \times \Delta t$, where E denotes *energy*, which is quite well known in both classical and quantum physics (Hamming 1989), and is not considered a purely quantum relation.

What happens when we combine this trade-off with the computable schedules of the experimental procedure, which bound the time? Do we find notions of the limits of measurement or quanta? Yes, in a way. Imagine a resource allocation problem that many readers will be familiar with, that of a research council buying time from a research group. With no schedule, the council keeps on paying for a result that will be delivered 'eventually'. With a schedule, there is a table of what results will be delivered by various deadlines, and failure to meet a deadline means that the grant is terminated.

As far as our CME measuring mass is concerned, we may as well assume for simplicity that we are running the bisection method, since we have shown that it is universal. We have shown that there are physical masses μ for which every possible computable schedule

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

will eventually fail. A particular computable schedule will fail at these masses, and more. That failure will consist of not being able to determine the nth digit in time $T(n)$ (after determining the previous $n-1$ digits). But then, using that schedule, we would not be able to distinguish any masses between the mass given by the determined $n-1$ digits followed by 0, and the mass given by the $n-1$ digits followed by 1. Thus, for that mass, and for the amount of effort that society (in the form of the research council) is prepared to devote to finding it, there is an effective 'quantum' or limit of observability. That limit can either be expressed as a limit on the number of places, or in terms of an interval. Of course, everything, in particular a notion of quantum, depends on μ and T. It is quite possible that the experimental procedure will continue indefinitely, as the CME for that mass and schedule keeps meeting every deadline.

9. Conclusions

We are developing a methodology and mathematical theory to examine how data is represented and computed using physical systems. Our primary tasks are to study computation by

(i) physical systems in isolation; and
(ii) physical systems combined with algorithms.

Our objectives are foundational rather than technological. The main ideas and results of this paper play an influential role in our research programme. The methodological principles of Section 2, especially Principles 5 and 6, allow us to pursue questions of interest in computation, physics and philosophy. We believe the CME exemplifies technical ideas and properties that have very wide application.

In summary, the CME focusses attention on the idea that the communication between the Turing machine and an external physical device is complicated and that the concept of a *protocol* is important and essential. In particular, physical oracles seem to require time to consult that depends exponentially upon accuracy (see the Conjecture 9.1). Applying Principle 5, we showed that Turing machines and protocols boost computational power beyond the Turing barrier (Section 6). Applying the new Principle 6, we showed that Turing machines and protocols reduce the power of experiments to measure the classical continuum (Sections 7 and 8). In the rest of this section, we will reflect further on time, our earlier experiment SME and uncertainty.

9.1. *Time and a conjecture*

In his essays Bachelard (1940; 1934), Bachelard stresses the fact that accuracy in a measurement in Physics is related to time: more accurate measurements consume more time. In our setting, the protocol must cope with the time:

(a) to settle the parameters of the experimental equipment; and
(b) to accomplish the experiment to the desired accuracy.

The Turing machine's query denotes the actual values of the parameters and the desired accuracy is given by the size of the query. Now, while (a) may depend upon the

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

experimenter, (b) depends only on the physical theory T specifying the experiment: the protocol's schedule is a measure of the time needed, or allowed, to perform the experiment and retrieve the result. In the collider experiment, the time needed for a collision experiment is exponential in the size of the Turing machine query. This is a new and remarkable fact about the protocol because it seems to be common; indeed, exponential time seems to be the norm.

The new features of the CME, which contrast with our earlier work on the SME, led us to consider several standard experiments in Physics, measuring distance, inertial mass, resistance, temperature, the ratio e/m of an elementary particle in the Coulombian field applying classic or quantum methods, and the Brewster angle in optics. In all these experiments we found that the *the time needed for an experiment is exponential in the size of the Turing machine query*. We intend to consider these experiments in detail in later papers, such as Beggs *et al.* (2010), and so provide more evidence for this statement. For the moment we just note that it is extremely difficult to get round this exponential property without resorting to experiments effectively governed by some discontinuous functions (or with discontinuous derivatives), and that many of these experiments would be considered to be 'unrealistic'.

In inequality (16), the lower bound for the experimental time of the CME can be considered as less controversial than the upper bound. In our fragment of physical theory, we have a somewhat idealised world and it is likely that trying to make the theory more 'realistic' would increase the experimental time. In other words, the lower bound of Inequality 16 is likely to remain, though adding 'realism' might cast doubt on the upper bound.

We shall make a conjecture about the behaviour of Turing machines using physical oracles, for which we plan to publish more evidence in due course. It is based on the idea that the lower bound to experimental time similar to that in Inequality 16 is a universal feature of 'realistic' theories.

Conjecture 9.1. For 'realistic' physical theories, using an experiment as a physical oracle requires an exponential time protocol.

We have begun the refinement and formalisation of this conjecture and the exploration of its consequences as a general mathematical property. If the conjecture is true generally, the project of finding physical systems that allow us to measure some physical quantity ever more accurately *and* ever more efficiently – for example, allowing us to halve the error without doubling the time taken – is condemned to failure.

This exponential lower bound limits the rate that the Turing machine can 'extract' information from the physical oracle and so limits any computational power that may be added to the Turing machine as a result of being connected to the physical oracle.

We should add a comment on the notion of 'time' we have used here. We discuss the time taken to perform the experiment, and this is measured in 'clock cycles' of the Turing machine. The counting of the clock cycles is not itself a part of the physics of the experiment – if the experiment required some form of clock, that would be listed under the experimental apparatus. However, we do assume that the clock cycles are measured or counted at a constant rate in experimental time (so, for example, our results would not

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

apply in a relativistic setting where the Turing machine and experiment suffered different time dilation effects).

9.2. *Comparing the CME and SME*

The scatter machine experiment SME was introduced in Beggs and Tucker (2007b) and studied as an example of experimental computation using the principles in Section 2. We showed that the experiment could measure or compute non-computable numbers in $[0, 1]$. Among its principal features are the bisection method and non-deterministic discontinuous physical behaviour. The theory of oracles began in Beggs *et al.* (2008b), where we introduced the ideas of:

(a) infinite, unlimited and finite precision; and
(b) protocols.

Again, the theme was computational power: we introduced Principle 5 and characterised the power using non-uniform complexity classes. However, there are some important differences in our results for the SME and CME.

In the scatter machine, the time needed to perform a single experiment is constant. The protocol for the scatter machine we used was polynomial, but this is quite an arbitrary assumption to cover the time taken to set up the cannon to the desired accuracy, fire it and observe the result: it assumes that setting up an experiment to a higher accuracy takes more time. However, once the cannon position is set, the experiment is concluded in a few seconds.

A second difference is in the computational power of the two experiments as oracles. The results in Beggs *et al.* (2008b; 2009a) and the current paper are compared in the following table:

Computational Power

Experimental oracle	*precision*	*complexity class lower bound*
scatter machine experiment	infinite	$P/poly$
	unbounded	$BPP//poly = P/poly$
	fixed	$BPP//log*$
collider machine experiment	infinite	$P/log*$
	unbounded	$P/log*$
	fixed	$BPP//log*$

The transition from the SME to the CME implies a loss of computational power. In the infinite precision and unbounded precision cases, the computational power, in polynomial time, falls from $P/poly$ to $P/log*$. However, in the finite precision case, where the protocols have no influence, the computational power, in polynomial time, is the same $BPP//log*$. More subtly, in the CME the computational power rises from the infinite and unbounded precision $P/log*$ to finite precision $BPP//log*$ because the protocol is exponential, and this does not affect the stochastic nature of the finite precision

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

calculations. In the SME, the computational power falls from the infinite and unbounded precision $P/poly$ to finite precision $BPP//log*$ because the protocol is polynomial, and this is a stronger condition.

Now, the SME involves discontinuous behaviour in the velocity of scattered particles as a function of the unknown position, which leads to non-determinism at the vertex of the wedge (at the vertex, the velocity jumps from a finite non-zero value to the negative of that value), while the CME is continuous (the velocity of a test particle falls to zero as its mass approaches the unknown mass from below and then increases again as the mass deviates from the unknown mass from above, but with opposite sign).

Does the SME falsify Conjecture 9.1? The scatter machine has an upper bound that is not of the form in inequality (16) – in fact, we could take a constant upper bound. Recently, however, we have revisited the SME and found that introducing more 'realistic' assumptions into the theory used – rounding the vertex, for example – not only removes this constant upper bound, but institutes a lower bound having the form of inequality (16). This extended SME has an exponential protocol and the same computational power as CME. See Beggs *et al.* (2009e) for a full account including an extended comparison table. This leads us to the following conjecture.

Conjecture 9.2. For 'realistic' physical theories, using an experiment as a physical oracle with an exponential time protocol boosts the power of Turing machines to $P/log*$ for infinite and unbounded precision and to $BPP//log*$ for fixed precision.

Thus, our experiences with a portfolio of physical oracles suggests that the CME introduces limiting results of wide relevance to the physical sciences.

9.3. *On measurement*

In *The Science of Mechanics*, Ernst Mach observed that:

'The laws of impact were the occasion of the enunciation of the most important principles of mechanics, and furnished also the first examples of the application of such principles.'

It seems that the same laws, by governing the CME, introduce some new properties of the concept of measurement in mechanics. This type of experiment to measure mass is at the heart of mechanics – a generalisation of the collider experiment can be used to measure the mass of a star or of a planet, measures that cannot be done with the balance scale.

Our measurement of *inertial mass* is *fundamental*, not *derived*: according to Hempel (Hempel 1952)):

'By derived measurement we understand the determination of a metric scale by means of criteria which presuppose at least one previous scale measurement. It will prove helpful to distinguish between derived measurement by stipulation and derived measurement by law. The former consists in defining a "new" quantity by means of others, which are already available; it is illustrated by the definition of the average speed of a point during a certain period of time as the quotient of the distance covered and the length of the period of time. Derived measurement by law, on the other hand, does not introduce a "new" quantity but rather an alternative way of measuring one that has been previously introduced.'

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

We do not use a scale of distance, or a scale of time. The algorithm only makes a fundamental direct measurement of mass (see Beggs *et al.* (2009d) for a deep discussion of this subject).

Eddington's comments on the fine-structure constant in Eddington (1933) are also relevant to our context of derived measurements:

'There has been much discussion whether the true value is 137.0 or 137.3; both values claim to be derived from observation. The latter, called the "spectroscopic value", is preferred by many physicists. It is, however, misleading to call these determinations *observational values*, for the observations are only a substratum; the spectroscopic value in particular is based on a rather complex theory and is certainly not to be treated as a "hard fact" of observation.'

In fact, in most situations, measurement is made by comparisons between observables, so what we describe applies not only to the measurement of *mass*, but widely in the physical sciences.

The idea of uncertainty is associated with *Heisenberg's indeterminacy principle* and is the subject of an enormous philosophical discussion about quantum mechanics. Uncharacteristically, Popper struggles in Popper (1950a; 950b) with the notion of indeterminacy in classical and quantum mechanics. The nature of our uncertainty is different, but its philosophical implications are similar.

Principle 6 is intriguing and may prove influential. Turing's analysis of people representing information symbolically and following a fixed procedure is an example of an anthropomorphic principle underpinning a scientific theory: models of computability rooted in human action are relevant to computing technologies of the past, present or future. It seems to us to be a beautiful idea to model the experimenter following an experimental procedure as a new form of program based on a physical theory T that can be coded as a Turing machine; and this idea resonates with the prominent and essential role of computers in performing experiments. It leads us to believe that measurability in Physics is subject to *laws* that are the effects of the limits of computability and computational complexity. Our algorithmic model of experiments imposes limitations on the physics we used to describe it. Not all masses can be known, not because of the limitations in measurements due to experimental errors, but because of essentially internal logical limitations of the theory. The mathematics of computation theory does not allow the reading of bits of physical quantities beyond a certain limit. Quantities cannot be measured with infinite precision, not because of the limitations of the physical apparatus but, more deeply, because of computational reasons. These unmeasurabilities allow for the definition of quanta of energy in the classical physical world.

We believe that the computational model of experimental measurement, here represented by the collider, demonstrates the existence of new and fundamental epistemic constraints in physics.

References

Bachelard, G. (1934) *Le nouvel esprit scientifique*. (English translation: Goldhammer, A. (1985) *The New Scientific Spirit*, Beacon Press.)

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

Bachelard, G. (1940) *La Philosophy du Non. Pour une philosophie du nouvel esprit scientifique*, Presses Universitaires de France.

Balcázar, J. L. and Hermo, M. (1998) The structure of logarithmic advice complexity classes. *Theoretical Computer Science* **207** (1) 217–244.

Balcázar, J. L., Días, J. and Gabarró, J. (1988) *Structural Complexity I*, Springer-Verlag (2nd Edition 1995).

Balcázar, J. L., Días, J. and Gabarró, J. (1990) *Structural Complexity II*, Springer-Verlag.

Beggs, E. and Tucker, J. V. (2006) Embedding infinitely parallel computation in Newtonian kinematics. *Applied Mathematics and Computation* **178** (1) 25–43.

Beggs, E. and Tucker, J. V. (2007a) Can Newtonian systems, bounded in space, time, mass and energy compute all functions? *Theoretical Computer Science* **371** (1) 4–19.

Beggs, E. and Tucker, J. V. (2007b) Experimental computation of real numbers by Newtonian machines. *Proceedings of the Royal Society, Series A (Mathematical, Physical and Engineering Sciences)* **463** (2082) 1541–1561.

Beggs, E. and Tucker, J. V. (2008) Programming experimental procedures for Newtonian kinematic machines. In: Beckmann, A., Dimitracopoulos, C. and Löwe, B. (eds.) Proceedings of the 4th conference on Computability in Europe: Logic and Theory of Algorithms. *Springer-Verlag Lecture Notes in Computer Science* **5028** 52–66.

Beggs, E., Costa, J. F., Loff, B. and Tucker, J. V. (2008a) On the Complexity of Measurement in Classical Physics. Theory and Applications of Models of Computation (TAMC 2008), Xi'an, China. *Springer-Verlag Lecture Notes in Computer Science* **4978** 20–30.

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

Beggs, E., Costa, J. F., Loff, B. and Tucker, J. V. (2008c) Oracles and Advice as Measurements. In: Calude, C. S., Costa, J. F., Freund, R., Oswald, M. and Rozenberg, G. (eds.) Proceedings UC 2008 – 7th International Conference on Unconventional Computation. *Springer-Verlag Lecture Notes in Computer Science* **5204** 33–50.

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

Beggs, E., Costa, J. F. and Tucker, J. V. (2009b) Unifying science through computation: Reflections on computability and physics. *New Approaches to the Unity of Science, Vol. II: Special Sciences and the Unity of Science – Logic, Epistemology, and the Unity of Science*, Springer-Verlag (in press).

Beggs, E., Costa, J. F. and Tucker, J. V. (2009c) Comparing complexity classes relative to physical oracles (technical report).

Beggs, E., Costa, J. F. and Tucker, J. V. (2009d) Computational Models of Measurement and Hempel's Axiomatization. In: Carsetti, A. (ed.) Causality, Meaningful Complexity and Embodied Cognition. *Theory and Decision Library A* **46**, Springer-Verlag 155–184.

Beggs, E., Costa, J. F. and Tucker, J. V. (2009e) The impact of limits of computation on a physical experiment (technical report).

Beggs, E., Costa, J. F. and Tucker, J. V. (2010) Physical oracles: The Turing machine and the Wheatstone bridge. In: Aerts, D., Smets, S. and Van Bendegem, J. P. (eds.) The contributions of Logic to the Foundations of Physics. *Studia Logica* **95** 217–292.

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

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press

Brady, A. H. (1994) The busy beaver game and the meaning of life. In: Herken, R. (ed.) *The Universal Turing Machine: A Half-Century Survey*, 2nd Edition, Springer-Verlag 237–254.

Cooper, S. B. (2004) *Computability Theory*, Chapman and Hall.

Eddington, A. S. (1933) *The Expanding Universe*, Cambridge University Press.

Froda, A. (1959) La finitude en mécanique classique, ses axiomes et leurs implications. In: Henkin, L., Suppes, P. and Tarski, A. (eds.) *The Axiomatic Method, with Special Reference to Geometry and Physics*, Studies in Logic and the Foundations of Mathematics, North-Holland.

Hamming, R. W. (1989) *Digital Filters*, (2nd Edition) Prentice-Hall.

Hempel, C. G. (1952) Fundamentals of Concept Formation in Empirical Science. *International Encyclopedia of Unified Science*, **2** (7) University of Chicago Press.

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

Popper, K. R. (1950) Indeterminism in quantum physics and in classical physics. *The British Journal for the Philosophy of Science* **1** (2) 117–133.

Popper, K. R. (1950) Indeterminism in quantum physics and in classical physics (Part II). *The British Journal for the Philosophy of Science* **1** (3) 173–195.

Radó, T. (1962) On non-computable functions. *Bell System Tech. J.* **41** (3) 877–884.

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

Ziegler, M. (2009) Physically-relativized Church–Turing Hypotheses: Physical foundations of computing and complexity theory of computational physics. *Applied Mathematics and Computation* **215** (4) 1431–1447.

https://doi.org/10.1017/S0960129510000356 Published online by Cambridge University Press