Physical Experiments as Oracles

Edwin Beggs∗†  José Félix Costa†‡  John V. Tucker†

1 Introduction

Over the past six years we have developed a methodology, some mathematical theory, and several applications, exploring physical foundations for computability and complexity. Enough of our theory and case studies have been published, sometimes in media remote from theoretical computer science, to warrant a short review of our approach and results1. Our aim of this review is to arouse curiosity, not to satisfy it: we will refer to our published papers for expositions, technical details and references to related work. Although we have read widely, we will be delighted to receive information about research related to our programme. Here we will concentrate on the theory of using a physical experiment as an oracle to an algorithm.

∗Corresponding author.
†School of Physical Sciences, Swansea University, Singleton Park, Swansea, SA3 2HN, Wales, United Kingdom. Email: E.J.Beggs@Swansea.ac.uk and J.V.Tucker@swansea.ac.uk.
‡Department of Mathematics, Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa, Portugal, and Centro de Matemática e Aplicações Fundamentals do Complexo Interdisciplinar, Universidade de Lisboa, Lisbon, Portugal. Email: fgc@math.ist.utl.pt.
1We thank Prof. Rozenberg for the invitation to discuss our work in the Bulletin.
1.1 Our questions

Our research programme began by focussing on two questions:

Consider a physical experiment in which some independent physical quantities \( x_1, x_2, \ldots \) may vary and a series of corresponding behaviours \( y_1, y_2, \ldots \) can be observed. The experiment defines a function \( y_i = f(x_i) \) or, more generally, a relation \( r(x_i, y_i) \) on real numbers, for \( i = 1, 2, \ldots \).

**Question 1.** *How can we use such an experiment to compute a function \( f \), or decide a relation \( r \), on a set of data? How do we measure its performance?*

To answer the question we must think precisely about data representation by physical quantities, experimental equipment and experimental procedures; and we must gain a thorough understanding of one (or more) physical theories. This form of calculation, based directly and exclusively upon a physical system, we called experimental computation. Now, for any particular type of experimental computation we can ask the question:

**Question 2.** *Is experimental computation equivalent to algorithmic computation? If not, what is the difference?*

Specifically, can we determine what functions and sets a particular class of experiments computes? The question is related to all sorts of physical aspects of computing to be found in research on Natural and Unconventional Computing. In Logic, one recalls Kreisel’s questions and speculations in [21].

The papers that focussed on these physical issues are [12, 13, 14, 15, 16, 17].

Building upon our first steps at making physical theories of computing, we considered combining experimental and algorithmic computation. Our third question is an extension of the last:

**Question 3.** *If a physical experiment were to be coupled with algorithms, would new functions and relations become computable or, at least, computable more efficiently?*

To answer this question we pursued the idea of using experiments as oracles to algorithms. This has proved to be interesting and productive, both conceptually and technically. Changing the direction of thought, from computation to physics, we have begun to answer this question:

**Question 4.** *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?*

Our papers on these hybrid issues are [5, 4, 6, 10, 7, 8, 11, 9].

1.2 Our programme

We introduced the idea of experimental computation to capture a large and diverse set of examples, old and new, and to formulate our general approach to
analysing computation as physical process. Examples abound in modelling natural systems and technologies for designing machines. Experimental computation involves choosing a physical system, which may be a part of nature or a machine, and is therefore dependent on specific physical theories; these are needed to specify and reason about what can be computed. Thus, the experiments are idealised and abstract experiments — "gedankenexperimente". We developed a number of general principles and case studies to explore and shape some theory. For completeness, we give the principles in Section 2, but there is no substitute for the looking at their origins and use in [12, 13, 14, 15, 16, 17].

As a case study, an experiment was devised to measure the position of the vertex of a wedge to arbitrary accuracy, by scattering particles that obey some laws of elementary Newtonian dynamics [15]. Let SME denote this scatter machine experiment. The SME was put under a theoretical microscope: the Newtonian theory was specified precisely and theorems were proved that showed that the experiment was able to compute numbers measuring positions that were not computable by algorithms. Indeed, the experiment SME could, in principle, measure any real number to any accuracy. Thus, [15] contains a careful answer to Question 2 above, in the negative: the experiment is capable of measuring wedge positions that are non-computable numbers. We argued why this cannot be ruled out by the physical theory.

To pursue Questions 3 and 4, we imagined using a physical experiment as an oracle to a Turing machine, which on being presented with, say, \( x_i \) as its \( i \)-th question, returns \( y_i \) to the Turing machine. Now, choosing a physical experiment to use as an oracle is a major undertaking. The experiment belongs to physics and involves concepts from the study of experimental computation (equipment, experimental procedure, measurement, observable behaviour, etc.). The connection between the abstract computing device and the physical experiment is defined by a protocol through which each query communicates information to fix the parameters of an experiment and, afterwards, receives an answer from the oracle to the device (e.g, yes or no).

We began to address Question 3 by using the SME as an oracle to a Turing machine and attempting to classify the computational power of the new type of analogue-digital system. Given the strong results in [15], the SME seemed an interesting choice of oracle that should enhance the computational power and efficiency of the Turing machine. We found that physical oracles demanded changes to some basic assumptions. For example:

(a) the protocol consumed resources, especially time, that had to be integrated into the complexity measures;
(b) the setting of initial conditions by queries could be exact or error prone;
(c) the experiments could have non-deterministic behaviour.

In [4, 7], we began the study of physical oracles by formulating certain prop-
erties of protocols and precision, and by determining the computational power of these machines in terms of non-uniform complexity classes (see [3]).

In [8] we studied in depth a second Newtonian experiment to measure inertial mass, called the collider machine experiment, CME. The protocol necessary for the CME proved to be particularly revealing physically. In particular, we realized that we could view a Turing machine with an oracle in a rather different way. Rather than supposing the oracle was there to boost the computational power of the Turing machine, we assumed the Turing machine could be regarded as an idealised experimenter performing experiments in Nature².

Thus, we have two questions and two reasons for interesting ourselves in physical oracles for algorithms: they help us think about the computing power of physical technologies and the laws of physical measurement.

## 2 Computation and physical systems

We summarise six methodological principles for the theoretical investigation of Questions 1-4.

### 2.1 Methodological principles for experimental computation

The idea of experimental computation is to attempt to analyse physical models of computation independently of the theory of algorithms. Physical theories play a fundamental role in understanding experimental computation, which we have discussed at length elsewhere [13, 14]. To seek conceptual clarity, and mathematical precision and detail, we proposed, in [13, 14], 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 behaviour and seek necessary and sufficient

---

¹This is not unlike the role of algorithms in learning theory (e.g., [19]).
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.

To study experimental computation and seek answers to Questions 1 and 2, the key idea is to lay bare all the concepts and technicalities to be found in examples by putting them under a mathematical microscope. Our methodology requires a careful formulation of a physical theory $T$, which can best be done by axiomatisations, ultimately formalised in a logical language, i.e., a formal specification of a fragment of the physical theory. We analyse the computational behaviour of classes of systems that obey the laws of $T$ and so study $T$-computability. Our approach has been applied in a new discussion of the physical basis of the Church-Turing Thesis in Ziegler [23]. A language for Newtonian experimental procedures was described in [16].

2.2 Methodological principles for combining experiments and algorithms

Next, we extend our methodology to consider the interaction between experiments and algorithms, and answer Questions 3 and 4.

First, consider using 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 their rich theory of computational complexity.

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 $i$-th question to the oracle, the machine generates a string that is converted into a rational number $x_i$ and used to set an input parameter $p_i$ to the equipment. The experiment is performed and, after some delay, returns as output a rational number measurement, or qualitative observation, $y_i$, which is converted into a string or state for the Turing machine to process. The Turing machine may pause while waiting for the oracle. In summary:
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 extends the power and efficiency of the algorithmic model.

Secondly, consider the nature of an experimental procedure for making a measurement. The process of calculating initial conditions and performing an experiment step by step and interpreting the results can be expressed as a sequence of commands and rules. We imagine a human experimenter following an experimental procedure made up of basic operations defined by a physical theory \( T \). The analysis can be compared with Turing’s analysis of the human computer, which resulted in the Turing machine. If the \( T \)-commands are coded on the tape then the Turing machine represents an abstract model of the procedure. Of course, today many experiments are fully computer controlled.

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.

In [8, 10] we introduce this principle and the Question 4.

Our experiments are ideal. However, the ontology of our gedanken experiment means that there is no fundamental difference between the ideal experiments defined by physical theory and the Turing machine. Like the Turing machine, they unify the essential features of examples, and map the limit of physical reality.

3 The view from computation theory

Computational complexity theorists have been using, unconsciously, some forms of gedanken experiment, namely, throwing a fair coin. Using Turing machines and diverse accept/reject criteria, they measure computational power, say in polynomial time, by defining complexity classes such as \( P \), \( NP \), \( PP \), \( BPP \), \( R \), \( ZPP \), etc.

To introduce what physical oracles involve technically, let us dismantle the standard black box idea of an oracle.

Consider the classical model of a Turing machine with an oracle. We imagine the oracle to be an unknown external device and we model it as a set \( A \). The machine queries the oracle \( A \) with a string \( w \in \Sigma^* \), and the oracle answers “Is
w ∈ A?” in one computational step. Now, consider a particular type of oracle, the tally oracle. A tally oracle is a subset of Σ∗, with Σ a single letter alphabet, e.g., Σ = {0}. The class of sets decidable in polynomial time by Turing machines consulting tally oracles is the well known class P/poly; see [3] and Section 4.

**Oracle as a real number.** Let us interpret the tally set A as a real number x in the unit interval [0, 1] in its binary expansion: for any positive integer i, if 0i ∈ A, then the i-bit of the binary expansion of x is 1, otherwise it is 0. Suppose the tally set oracle A is replaced a real number x.

**Oracle as a physical quantity.** The number x can be seen as a value of a physical quantity (ignoring units of physical quantities) such as a distance between two points, an electric charge in an electrostatic field, or a mass of a particle. For definiteness, suppose (x ≡) μ is the measure of a mass defined by (a fragment of) Newtonian dynamics T. Suppose the abstract oracle A is replaced with an unknown physical experiment of measuring a mass defined by T.

**Oracle as an experiment measuring mass.** In T the measure of a mass is a collateral effect of combining Newton’s Second and Third Laws of dynamics. We derive the theorem of conservation of linear momentum in T and are able to calculate the value of the mass by an experiment using a proof particle of known mass to collide with the unknown mass. This experiment has centuries of commentaries and discussions, starting with Galileo. But now, as computer scientists hunting for the binary digits of μ, and not simply for a canonical measurement of the kind μ ± Δμ, an experimental procedure must be found to discover the binary bits of μ and use them as oracle to the Turing machine. Suppose the oracle is such an experiment.

**Cost of performing the experiment.** We choose a specific experiment for mass such as the collider machine CME in [8]. To measure the real number μ we have to proceed, step by step, with finite approximations. Thus, besides the unknown mass μ, we use proof particles of dyadic rational masses (easily denoted by finite binary strings). The idea is this: if we project a particle of known mass towards a particle of unknown mass, then the first will be reflected if its mass is less than the unknown mass, and it will be projected forward, together with the particle of unknown mass, if its mass is greater than the unknown mass. There is a bisection procedure which determines, bit by bit, the value of the unknown mass. But here we find another novelty: if we want to read the bits of μ using such a method then
the time needed for a single experiment is

\[ \Delta t = \left| \frac{1}{m - \mu} \right|, \]

where \( m \) is the mass of the proof particle in that single experiment. The time needed for a single experiment to read the bit \( i \) of the mass \( \mu \), using the proof particle of mass \( m \) of size \( i \) (number of its bits) is at best exponential in \( i \).

This experiment \( CME \) (fully described in [8]) tells us that the time needed to consult the oracle is no longer a single step but \textit{depends on the size of the query}. The Turing machine with the experiment as oracle must also be provided with a protocol, i.e., a process that manages the interface and counts the time needed in each consultation of the oracle.

**Precision in performing the experiment.** In the physical world it is not reasonable that a proof particle of mass \( m \) can be set with infinite precision. Does our measurement of \( \mu \), bit by bit, need us to possess equipment with infinite precision? The answer is \textit{No}. Suppose precision is not infinite but finite and unbounded, i.e., we can be as precise as we need. We have shown that we can continue reading the bits of \( \mu \). The passage from infinite to finite unbounded precision, in setting the mass of the proof particle, is no obstacle to the reading of bits: the same complexity classes are defined and the same protocols can be used to communicate between the Turing machine and the physical experiment. However, if we reject unbounded precision in favour of the more realistic \textit{fixed precision} criterion then we have shown that, using stochastic methods, we are still able to read the bits of \( \mu \). The lack of precision in measurement is not an obstacle to the reading of the bits of \( \mu \).

## 4 Reflections on computation

Let us consider Principle 5 in 2.2. Our discussion in Section 3 introduced a new complexity theory of Turing machines where oracles (a) need time to be consulted and (b) operate with various forms of precision.

**Managing oracles.** A protocol manages the interaction with the experimental oracle. First, we have to classify the protocols.

The Turing machine will generate a (dyadic) rational number \( x_i \) which can be used to set an experimental parameter \( p_i \) of SME in one of two ways: we call the machine \textit{error-free} if \( p_i = x_i \); and \textit{error-prone} if we can ensure that \( p_i \in [x_i - \varepsilon, x_i + \varepsilon] \) for an error margin \( \varepsilon > 0 \). In the error-prone case, we further
differentiate between fixed accuracy, where $\varepsilon > 0$ is fixed for the particular SME, and arbitrary accuracy, where $\varepsilon > 0$ can be made arbitrarily small.

The protocol consumes and measures the time (and, possibly, other physical resources) needed to settle the parameters of the experimental equipment and to perform the experiment. The time is dependent on the size of the Turing machine’s query, which denotes the values of the parameters.

These properties introduce these different types of protocols: polynomial time error-free, exponential time error-free and polynomial time error-prone with arbitrary or fixed precision. The SME possess a natural polynomial protocol whilst that of the CME is exponential.

Finally, we note that with the SME a particle hitting the vertex of the wedge may scatter randomly. Thus, the Turing machines are divided further by their deterministic and non-deterministic oracles.

**Classification.** Taking some inspiration from the technical work of Siegelmann and Sontag [22], we use non-uniform complexity classes of the form $B/F$ and $B//F$, where $B$ is the class of computations and $F$ is the advice class. Examples of interest for $B$ are $P$ and $BPP$; examples for $F$ are $\text{poly}$ and $\log^*$. The power of the machines will correspond to different choices of $B$ and $F$. A first result for the SME was this:

**Theorem.** The class of sets which are decidable in polynomial time by error-free deterministic analogue-digital scatter machines with a polynomial protocol is exactly $P/poly$.

For the error-prone machines we proved lower bounds in [4] and, later, upper bounds in [7], which enabled us to complete the classification of the power of the three kinds of scatter machines. For example, for the error-prone machines, we prove:

**Theorem.** The class of sets which are decidable in polynomial time by error-prone arbitrary precision deterministic analogue-digital scatter machines with a polynomial protocol is exactly $P/poly$.

Thus, there is no difference in computational power between using exact queries and approximate queries having arbitrary precision. This is not the case for fixed finite precision, which computes less for the scatter experiment SME.

The computational power reduction of reading capabilities of physical equipment through the two experiments we analysed so far in [5, 4, 6, 10] is summarized in the following table.
5 Reflections on physics

Let us consider Principle 6 in 2.2. The idea of using algorithms to completely control physical experiments seems to us to be both simple and radical, and the task of making a theory to explore its meaning and consequences an intriguing challenge. For example, one aim would be:

To create a theory about the nature of physical measurements.

Certainly, the experience of thinking about experimental computation is invaluable; we encounter problems of specifying physical theories, analysing and modelling experimental procedures, equipment, observers and experimenters. The subject is a potent mixture of the philosophical and technical, offering work for philosophers, logicians, computer scientists and physicists. Theoretical studies of measurement seem to be rather diverse.

Let us consider the fascinating paper [18], by the physicists Geroch and Hartle, published in 1986 to celebrate the 75th birthday of John Archibald Wheeler. We will outline some of their main ideas and speculations, which we will reinterpret and formalise using our theory.

Geroch and Hartle start by considering the concept of measurable number in contrast to the concept of computable number:

We propose, in parallel with the notion of a computable number in mathematics, that of a measurable number in a physical theory. The question of whether there exists an algorithm for implementing a theory may then be formulated more precisely as the question of whether the measurable numbers of the theory are computable.

Then they add some considerations on numbers being measurable and/or computable:

We argue that the measurable numbers are in fact computable in the familiar theories of physics, but there is no reason why this need be the case in order that a theory have predictive power. Indeed, in some recent formulations of quantum gravity as a sum over histories, there are candidates for numbers that are measurable but not computable.

They introduce the notion of observer and physicist for the purpose of measuring physical variables:
Regard number $w$ as measurable if there exists a finite set of instructions for performing an experiment such that a technician, given an abundance of unprepared raw materials and an allowed error $\varepsilon$, is able by following those instructions to perform the experiment, yielding ultimately a rational number within $\varepsilon$ of $w$.

The set of instructions which Geroch and Hartle refers to, together with some memory to take account of intermediate calculations, we can be interpreted as a Turing machine. The Turing machine represents the physicist, or the observer, or the experimenter. Thus, we propose the assumption:

**Postulate 1** The physicist is modelled by a Turing machine. The measuring process is controlled by an algorithm that runs on the machine, generating the atomic instructions to be performed at each step of the experimental procedure.

This postulate says that the experimenter cannot escape the logic of following rules as formalised by computability theory. The logic of experimental procedures can be captured by a Turing machine.

A point not considered in [18] is that not all measurements are possible. Assuming the physicist to be a Turing machine, then the limits of computation of the Turing machine imply limits on measurements; and, therefore, on the nature of physical experiments.

The accuracy $\varepsilon$ is to be understood as arbitrarily small. As we will see, our work makes the concept of measurable as precise as the concept of computable, which was not the intention of [18]:

“Measurable” is analogous to, although of course much less precise than, “computable”. The technician is analogous to the computer, the instructions to the computer program, the “abundance of unprepared raw materials” to the infinite number of memory locations, initially blank. Indeed, one can think of the measurable numbers as those that are “computable” using an analog, rather than digital, computer.

Geroch and Hartle stress need for a theory to specify a gedanken experiment as follows:

*The notion “measurable” involves a mix of natural phenomena and the theory by which we describe those phenomena. Imagine that one had access to experiments in the physical world, but lacked any physical theory whatsoever. Then no number $w$ could be shown to be measurable, for, to demonstrate experimentally that a given instruction set shows $w$ measurable would require repeating the experiment an infinite number of times, for a succession of $\varepsilon$s approaching zero. One could not even demonstrate that a given instruction set shows measurability of any number at all, for it could turn out that, as $\varepsilon$ is made smaller, the resulting sequence of experimentally determined rationals simply fails to converge. It is only a theory that can guarantee otherwise.*

Now, how does the Turing machine communicate with Nature? We believe that this interaction is captured by the concept of the continuing evolution of a
physical experiment acting as an oracle.

**Postulate 2** The measurement process is taken to be an oracle to a Turing machine. The interaction is achieved through a protocol. After each consultation, the oracle may provide one bit of the measurement. This bit is also information that provides the necessary information to the machine to proceed with the desired computation and experimental procedure.

These considerations are enough for them to prove theorems such as: Every computable number is measurable. Then the authors ask the following question:

> We now ask whether, conversely, every measurable number is computable — or, in more detail, whether current physical theories are such that their measurable numbers are computable. This question must be asked with care.

Actually, the question received a very careful answer in our [15]: the experiment SME demonstrates that there numbers that are measurable in Newtonian dynamics but that are not computable.

Much more can be said in answer to this question. According to a our framework all depends upon the physical theory chosen. For Newtonian mechanics we have shown that for some experiments quantities are always measurable (see [15]) whilst for others there are quantities that are not always measurable (see [10, 8]). Our technical results can be used to show that the task of measuring quantities in physics, let us say in polynomial time, can be classified by well known complexity classes.

Principle 6 and the postulates leads to a deeper understanding of experimenters and experiments which impose a theoretical and absolute limit on the measurability of a physical quantity. Indeed, we are exploring the conjecture that

Any quantity in physics may not be always measurable.

The ideas of protocol and precision discussed earlier play a role in the model. In [20, 1], inter alia, we find early explicit statements on measurability, accuracy, and time complexity of physical quantities. For instance, Bachelard starts digressing about this fact in his Philosophy of No ([1]) and, in [2], he states that improving accuracy of an experiment increases exponentially the time needed.

In the oracle model, we are not interested in the time taken to build the experimental apparatus from the unprepared raw materials a la Geroch and Hartle, nor in the time taken to tune the instruments to some precision. We are interested in the run time of the physical experiment.

### 6 Conclusion

Ours is one among a number of research programmes involving physical systems and computability. Our objectives are foundational rather than technological.
In [6, 11] we argue that algorithms with physical oracles occur naturally in Physics. In [9] we have studied relativisations of the $P = NP$ problem to physical oracles.

We have considered several standard experiments in Physics measuring distance, inertial mass, resistance, temperature, the ratio $e/m$ of an elementary particle in the classic or quantum Coulombian field, and the Brewster angle in optics. In all these experiments we found that physical experiments have an exponential time protocol. Our results are intended to be a theoretical best case, where introducing more realism would only serve to make the result worse. The project of finding physical systems which allow us to measure some physical quantity ever more accurately and efficiently — e.g., allowing us to halve the error without doubling the time taken (or multiplying by some other fixed factor) — we conjecture to be condemned to failure.

Our work with Principle 6 leads us believe that measurability in Physics is subject to laws which are co-lateral effects of the limits of computability and computational complexity. Our model 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 essentially computational reasons. These unmeasurables allow the definition of quanta of energy for the classical physical world, accompanied by uncertainty principles, courtesy of the theory of computation.

Edwin Beggs, José Félix Costa and John Tucker would like to thank EPSRC for their support under grant EP/C525361/1.

References


