Experimental computation of real numbers by Newtonian machines

By E. J. Beggs[1],[*] and J. V. Tucker[2]

[1]*Department of Mathematics, and* [2]*Department of Computer Science, University of Wales Swansea, Singleton Park, Swansea SA3 2HN, UK*

Following a methodology we have proposed for analysing the nature of experimental computation, we prove that there is a three-dimensional Newtonian machine which given *any* point $x \in [0, 1]$ can generate an infinite sequence $[p_n, q_n]$, for $n = 1, 2, \ldots$, of rational number interval approximations, that converges to x as $n \to \infty$. The machine is a system for scattering and collecting particles. The theorem implies that every point $x \in [0, 1]$ is computable by a simple Newtonian kinematic system that is bounded in space and mass and for which the calculation of the nth approximation of x takes place in $O(n)$ time with $O(n)$ energy. We describe variants of the scatter machine which explain why our machine is non-deterministic.

Keywords: physical models of computation; experimental computation;
Newtonian machines; computable real numbers; non-computable physical systems

1. Introduction

Computability theory, founded by Church, Turing and Kleene in 1936, is a deep theory about the functions computable by algorithms on discrete data. At its heart is the concept of an *algorithmic procedure* operating on finite representations of data. Algorithmic procedures have been analysed mathematically in great detail, using theoretical models of machines and programming languages. Computability theory has been extended to continuous data via approximations. Over the past two decades, a stable theory of computable functions on continuous data has begun to take shape. The subject has been guided by foundational questions and applications to mathematics and computer science.

Almost independently over the past two decades, the ideas that *information and computation are physical processes* have emerged from different corners of logic, physics and computer science and have begun to exert influence on computability theory. In mathematical logic, Kreisel drew attention to the question 'Is physical behaviour computable?' in papers such as Kreisel (1974*a*), which stimulated research in computable analysis, including Pour-El (1974) and Pour-El & Richards (1979, 1981, 1983, 1989). In physics, work of Feynman on the physical basis of computation have led to subjects such as quantum information processing (Nielsen & Chuang 2000; Hey 2002). In computer science, hybrid control systems have led to new approaches to analogue computation

* Author for correspondence (e.j.beggs@swansea.ac.uk).

such as Brockett (1991, 1994). Despite a large disparate literature, it is hard to answer the following questions.

What are the numbers and functions computable by experiments with physical systems?

How do they compare with the numbers and functions computable by algorithms?

For instance, the search for physical systems, called hyper-computers, that can compute more than the Turing machine has been controversial and inconclusive. One reason is that the idea of hyper-computation is subtle, as is shown in Loff & Costa (in press). Another reason is that, in contrast to algorithmic computation, the idea of experimental computation is not understood. A theory of what it means to compute by experiment is needed. At the heart of such a theory are the concepts of *experimental procedure* and *equipment*, both of which must be based firmly on the precise formulation of particular physical theories (Beggs & Tucker 2006, 2007).

Here, we a fix a small subtheory T of Newtonian kinematics and present a simple example of a Newtonian machine based on T that performs a form of hyper-computation. Specifically, we consider an experimental procedure that computes a real number. The computation of physical quantities described by real numbers was first discussed in Geroch & Hartle (1986), with physical constants in mind. We prove the following theorem.

Theorem 1.1. *There exists a three-dimensional Newtonian kinematic machine which can compute the position of any point on a unit line segment to arbitrary accuracy; given any point $x \in [0, 1]$ the machine can generate an infinite sequence $[p_n, q_n]$ of rational number interval approximations of x such that for each n, we have*

$$x \in [p_n, q_n] \quad and \quad |p_n - q_n| < \frac{1}{2^n}.$$

The machine is a system for scattering, collecting and observing particles. The entire system is bounded in the following sense.

Space. The system can be contained within an arbitrary shallow box.
Mass. The total mass of the system is bounded.
Time. The calculation of the nth approximation takes place in linear time $O(n)$.
Energy. The calculation of the nth approximation uses linear $O(n)$ energy.

Corollary 1.1. *Every point in the unit interval $[0, 1]$ is computable by experiment using machines based on simple subtheories of Newtonian kinematics.*

Our scatter machine shows that the existence of hyper-computers is provable in a physical subtheory T of Newtonian kinematics. A crucial point for the nature of this hyper-computation is the extent to which the assumption *any point on a one-dimensional scale can be the position of an object*, and its variations, depend on the subtheory T. We discuss reasonable general mathematical and physical assumptions on positioning, including the probability of choosing a point that is a non-computable number.

The scattering machines can be designed using a number of variations and settings; these constitute refinements of the Newtonian kinematic theory T. Interestingly, the basic machine is *non-deterministic* in a fundamental way: the theory T does

not specify what happens to a particle when it meets a sharp edge. To explain this non-determinism, we give an account of the behaviour of different shapes of particle striking sharp edges and show that they require inconsistent deterministic extensions of the basic theory T. If we use spherical particles and triangular particles with size $s > 0$, then we have deterministic scatter machines. However, the behaviours of these scatter machines as $s \to 0$, and the particles approach point particles, are different. The inconsistency of these refinements of T explains the need for us to keep the scatter machine non-deterministic. This process of refining and dissecting kinematic theories is required by the working methodology.

With some modest assumptions on the scattering machine, we can prove the following (proposition 5.1).

Theorem 1.2. *A scatter machine that is deterministic and satisfies a simple translation invariance property can computably decide by experiment the following. Given any points x, y in the unit interval $[0, 1]$, is $x \leq y$, or $y \leq x$, and possibly both.*

At first sight, the theorems show that everything is computable by simple Newtonian mechanical systems. Given the Church–Turing thesis, this raises the question 'is the elementary theory of Newtonian kinematics undesirably strong?' To answer this question, further examples and subtheories need to be studied and the precise *physical* concepts and laws identified that cause non-computable functions and behaviours.

The structure of the paper is as follows. In §2, we describe the methodology of our approach. In §3, the construction of a simple type of scatter machine is given. In §4, we consider the role of coordinate systems. In §5, we give further examples that refine the scatter machine and explore non-determinism. In §6, we reflect on the nature of hyper-computation exhibited by the scatter machine. Finally, in §7, we offer some concluding remarks.

This paper is a sequel to Beggs & Tucker (2004, 2006, 2007). Only elementary Newtonian mechanics is needed to follow our technical arguments here. As background, it is helpful if the reader is familiar with some of the literature on hyper-computation (Cooper & Odifreddi 2003; Beggs & Tucker 2004; Davis 2004; Loff & Costa in press), with the theory of the functions computable by algorithms on discrete data (see Odifreddi 1989; Stoltenberg-Hansen & Tucker 1995, 1999a; Griffor 1999) and its extension to continuous data (Pour-El & Richards 1989; Stoltenberg-Hansen & Tucker 1999b; Tucker & Zucker 1999, 2000, 2004; Weihrauch 2000).

2. Principles for the study of experimental computation

We consider a set of principles for investigating experimental computation and hyper-computation in a rigorous way, first used in Beggs & Tucker (2006, 2007).

(a) Experimental computation

Definition 2.1. *Experimental computation* consists of an *experimental procedure* which is a specification for a finite or infinite process, called an *experiment*, made of primitive actions that are applied to a physical system,

called the *equipment*. The actions allow the construction, alteration, initialization and observation of the equipment. The actions define input and output data by setting parameters and measuring effects. The procedure schedules the actions, possibly in parallel, in time measured by clocks. Briefly,

$$\text{experimental computation} = \text{experimental procedure} + \text{equipment}.$$

For example, a partial function $y=f(x)$ can be computed by such experiments in three stages:

(i) input data x are used to determine initial conditions of the physical system,
(ii) the system operates for a finite or infinite time, and
(iii) output data y are obtained by measuring the observable behaviour of a system at specified times.

This idea of *experimental computation* captures a wide variety of examples, old and new. It depends upon a choice of physical system, which may be a part of nature or a machine. In turn, experimental computation is dependent on different physical theories that explain what can be computed. The concept can be found in modelling natural systems, e.g. in classical wave mechanics (Kreisel 1974b; Pour-El & Richards 1981, 1983, 1989; Weihrauch & Zhong 2001, 2002; Stoltenberg-Hansen & Tucker 2003) and in technologies for designing machines, e.g. in the nineteenth century mechanical systems of analogue computers (Thompson & Tait 1880; Bush 1931; Shannon 1941; Pour-El 1974; Lipshitz & Rubel 1987; Moore 1996; Graça & Costa 2003). Computation is a theme in studies of emergent behaviour (e.g. cellular lattices and neural networks, Holden *et al.* 1992; Siegelmann 1999; Hey 2002; Wolfram 2002) and of quantum computers (Nielsen & Chuang 2000; Hey 2002).

(*b*) *Methodological principles*

Physical theories play a complicated role in experimental computation. In seeking answers to the questions in the Introduction, we should expect to use a physical theory to

(i) define precisely the class K of physical systems under investigation.

To do this, the concepts and laws of the theory will be used to

(ii) design and construct the systems in K and define their primitive experimental operations,
(iii) explain and prove properties of their operation and behaviour, and hence
(iv) validate experimental computations by the systems of K.

If a hyper-computer has been found in K, then we should use the theory to

(v) evaluate the physical credibility of the system, or possibly
(vi) reveal weaknesses in the theory.

Finally, in each step above, we should use the theory to

(vii) make clear, precise and detailed statements about the whole process of experimental computation, especially about the experimental procedure and equipment.

Comprehensiveness, clarity, precision and details are in short supply in the literature on experimental hyper-computation. To change this, we have proposed, in Beggs & Tucker (2006, 2007), the following methodology based on four principles and stages for an investigation of some type of experimental computation.

Principle 2.1. 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.*

All physical theories are large and changing and it is not easy to define *exactly* such a theory. Therefore, one should begin to work on examples based on some small subtheory *T*, emphasizing conceptual precision and mathematical rigour. What is contained in *T* determines the set of possible experimental procedures and equipment. (Elsewhere, in Beggs & Tucker (2007), we discuss how to derive, from *T*, formally defined languages in which to specify both experimental procedures and equipment.) *It does not matter whether we think of the simple theory T as true, or roughly applicable, or know it to be false.* For example, in all work on Newtonian systems, here and elsewhere, it is *known* that assumptions about space and time are false of our Universe. If Newtonian examples can be intriguing, confusing and disputed then it is not surprising that relativistic and quantum examples are not definitive. The point can be made more strongly. It is possible to base an example on any clearly defined physical theory, including 'obsolete' theories, such as Descartes' theory of collisions. Of course, the problem with an old abandoned theory is that, typically, it may not provide us with the contemporary level of rigour that we need.

Principle 2.2. 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.*

Physical theories do not contain notions of computation. The computation of functions and sets must be abstracted from the operation of physical systems. For the purposes of comparison with algorithmic procedures, and especially for the study of hyper-computation, one may seek ways of embedding *algorithmically complex computations* into *simple physical systems* that are models of *simple physical subtheories*.
From our reading of the literature, systems perform hyper-computation using some embedding of a non-computable set or function into the system, e.g. in the initial states of systems or parameters. Often, such embeddings are hidden by the *complexity* of the examples and need to be unearthed through careful study. Hyper-computation by courtesy of such an embedding is a property of devices that we have called *deus ex machina* (Beggs & Tucker 2004): such devices are not 'genuine' hyper-computers unless there is a physical reason for the non-computable property to be present. The literature has long been plagued by uncertainty and controversy from Kreisel (1974a,b) to Davis (2004). Even after 40 years, investigations are still at an early stage so *we should not care about the*

validity of the physical theory, but we should care about being able to make theoretically complete models and precise mathematical analyses that lay bare all the concepts and technicalities.

Principle 2.3. 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 behaviours and seek necessary and sufficient conditions for the systems to implement precisely the algorithmically computable functions.*

It is a problem to isolate what properties of a physical theory permit computers and hyper-computers. In our earlier Newtonian work, it is obvious where non-computability enters and we can isolate some necessary and sufficient algorithmic conditions on the subtheories that can control the embedding of sets of natural numbers. However, it is *not* obvious how to control non-computability using purely physical conditions.

Principle 2.4. 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.*

The process of reviewing examples can lead to the refinement of the subtheory in different ways, perhaps adding more laws in the search of more realism, e.g. what happens if we add friction and/or elasticity to a Newtonian system. It certainly leads to a portfolio of theories and examples, which taken together allows us to draw rigorous conclusions.

Refinement can lead to the rejection of the subtheory and its systems as a basis for making a hyper-computer. As one moves out of T, one can contradict other theories—theories created to cover a larger experimental domain. In Newtonian kinematics, using arbitrary velocities contradicts relativity theory, and making arbitrary small components contradicts atomic theory.

A physical theory cannot be proved, but it can be tested and validated by experimental facts. For any theory T, there are restrictions that define that part of the theory which has been validated experimentally. These restrictions constitute a refinement theory $E(T)$, called the *experimental domain* of T. $E(T)$ will contain numerical bounds on physical measurements, for instance. In our work, we have worked with the unrestricted kinematic theory T, ignoring the experimental domain $E(T)$, and have concentrated on mathematical rigour, consistency and completeness.

(*c*) *Newtonian theories*

Newtonian kinematics is about systems of particles in n-dimensional space ($n = 1, 2, 3$) satisfying Newton's laws of motion and laws such as gravitation and conservation of energy. Typically, subtheories describe systems that allow the

(i) projection of particles in space with arbitrary velocity and
(ii) observation of the position of particles at certain times.

A subtheory of Newtonian kinematics may further assume that the space may be (i) structured in different ways, (ii) influenced by gravitational fields, (iii) populated by special bodies and obstacles, and (iv) shaped by geometries. It may also suppose that materials have different physical properties such as friction and elasticity.

Our methodology requires a careful formulation of a physical theory T, so the question arises what exactly constitutes a subtheory of Newtonian kinematics. This can best be answered by axiomatizations, ultimately formalized in a logical language.

Newtonian kinematics has appeared in several discussions of the physical basis of computation, including non-computability and hyper-computation (Moore 1990; da Costa & Doria 1991a,b; Xia 1992; Smith 2006), and complexity (Yao 2003). We have surveyed the quest for non-computable physical behaviour in our paper (Beggs & Tucker 2004).

3. The scattering machine

(a) A simple kinematic theory T

The *scatter machine* is based on a subtheory T of Newtonian mechanics comprising the following.

  (i) Point particles obey Newton's laws of motion in the two-dimensional plane.
 (ii) Straight line barriers have perfectly elastic reflection of particles, i.e. kinetic energy is conserved exactly in collisions.
(iii) Barriers are completely rigid and do not deform on impact.
 (iv) Cannons, which can be moved in position, can project a particle with a given velocity in a given direction.
  (v) Particle detectors are capable of telling if a particle has crossed a given region of the plane.
 (vi) A clock measures time.

The subtheory T is very simple. Furthermore, we will *not* need absolute precision in measurements, either in space or time. In fact, we can allow generous margins of error in measurements. We will *not* need to calculate with algebraic formulae. We will review the subtheory after we have constructed the machine and discussed its operation, and also when we consider possible refinements.

(b) Specification of the scattering machine

The machine consists of a cannon for projecting a point particle, a reflecting barrier in the shape of a wedge and two collecting boxes, as pictured in figure 1. The wedge can be anywhere, placed at random by, say, kicking the wedge. The cannon will be moved and operated repeatedly to find the position of the wedge of arbitrary accuracy.

In figure 1, the actual parts of the machine are shown in bold lines, with description and comments in narrow lines. The double-headed arrows give dimensions in metres, and the single-headed arrows show a sample trajectory of the particle after being fired by the cannon.
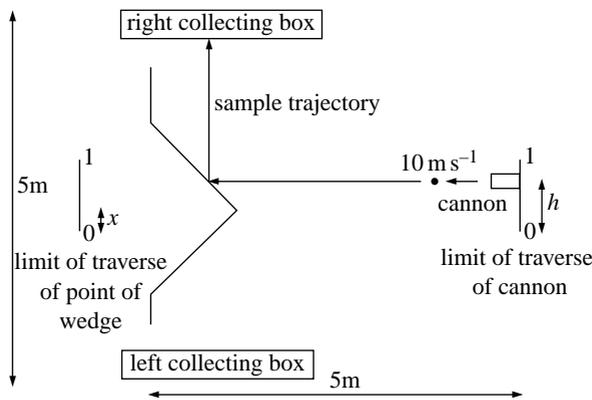
Figure 1. The scatter machine.

The sides of the wedge are at 45° to the line of the cannon, and we take the collision to be perfectly elastic, so the particle is deflected at 90° to the line of the cannon, and hits either the left or right collecting box, depending on whether the cannon is to the left or right of the point of the wedge. Since the initial velocity is 10 m s^{-1}, the particle will enter one of the two boxes within 1 s of being fired. Any initial velocity $v > 0$ will work with a corresponding waiting time. The wedge is sufficiently wide so that the particle can only hit the 45° sloping sides, given the limit of traverse of the cannon. The wedge is sufficiently rigid so that the particle cannot move the wedge from its position.

The reader may spot a problem here: what happens if the particle hits the point of the wedge? This is the only exception to the description above. In this case, we shall not make any assumption about what happens, as any assumption would doubtless prove controversial. We shall suppose that the particle could enter either box or not enter any box. This means that the scatter machine is non-deterministic.

(*c*) *Operation of the scattering machine*

Suppose that x is any position of the point of the wedge and is fixed. For a given cannon position h, there are three outcomes of an experiment.

(r) One second after firing, the particle is in the right box.
(l) One second after firing, the particle is in the left box.
(o) One second after firing, the particle is in neither box.

From the mechanics of the system, these outcomes can be connected to the parameters x and h in the following manner.

One second after firing, the particle is in the right box. Conclusion: $h \geq x$.
One second after firing, the particle is in the left box. Conclusion: $h \leq x$.
One second after firing, the particle is in neither box. Conclusion: $h = x$.

To actually conduct experiments with the scatter machine, rather than just making a single observation, we need to be able to alter the state of the machine.

Our task is to find x by altering h, so in our machine $0 \leq x \leq 1$ will be fixed, and we will perform observations at different values of $0 \leq h \leq 1$.

But what values of h are allowed? Let us take a special case of the *dyadic scatter machine*, where we can set h to be any fraction in the range $0 \leq h \leq 1$ with the denominator a power of 2.

Theorem 3.1. *Any position $x \in [0, 1]$ of the wedge can be calculated by the dyadic scatter machine.*

Proof. We will prove this by using an experimental procedure based on bisection. For any natural number N, we specify an accuracy 2^{-N}. We start by setting $n = 0$, $h_0 = 0$ and $h'_0 = 1$. Now use the following procedure.

(i) If $n = N$, stop.
(ii) Fire the cannon at positions h_n, h'_n and $(h_n + h'_n)/2$.
(iii) If the outcomes (one of r, l, o previous) at h_n and $(h_n + h'_n)/2$ are different, set $h_{n+1} = h_n$ and $h'_{n+1} = (h_n + h'_n)/2$ and go to (vii).
(iv) If the outcomes (one of r, l, o previous) at h'_n and $(h_n + h'_n)/2$ are different, set $h'_{n+1} = h'_n$ and $h_{n+1} = (h_n + h'_n)/2$ and go to (vii).
(v) If all three outcomes are right (case r previous), set $h_{n+1} = h_n$ and $h'_{n+1} = (h_n + h'_n)/2$ and go to (vii).
(vi) If all three outcomes are left (case l previous), set $h'_{n+1} = h'_n$ and $h_{n+1} = (h_n + h'_n)/2$ and go to (vii).
(vii) Increment n and go to (i).

The output is the numbers h_N and h'_N, and we are guaranteed that $h_N \leq x \leq h'_N$. As $h'_N - h_N = 1/2^N$, we have found x to the required accuracy. ∎

The analysis behind the program is as follows: at any stage n, the point of the wedge is in the interval $[h_n, h'_n]$ (where $h_n < h'_n$), and so must either be in the interval $[h_n, (h_n + h'_n)/2]$ or in $[(h_n + h'_n)/2, h'_n]$, or in both. If there is a difference in outcomes (i.e. r, l, o) between the end points of one of these two intervals, then the point of the wedge must be in the interval.

What if there is no difference between the end points of the two intervals? The reader should remember that complications arise when the point of the wedge is at an end point of an interval. If the cannon is fired at exactly this height, as we have not assumed that the machine is deterministic, we must assume that any of the outcomes r, l or o can happen. If all three outcomes are the same, they must be either rrr or lll. The only way the case rrr can happen is if the point of the wedge is at h_n, and the only way the case lll can happen is if the point of the wedge is at h'_n.

It is also possible to have different outcomes between the end points of both of the half-intervals, but in this case, the point of the wedge must be at the mid point, and it does not matter which half-interval we choose.

In the matter of the bounds, clearly the space and mass are bounded easily. The fixed initial velocity (10 m s^{-1}) is designed to make sure that each experiment that involves one shot is over by a certain amount of time τ, after which we can look at the result. Clearly, for the nth approximation the time taken is $n\tau$, i.e. $O(n)$. The energy per experiment is constant, say k, and hence for the nth approximation, the energy is nk, i.e. $O(n)$.

Table 1. Table for the bisection method.

outcomes	no. r	no. l	set $h_{n+1}=$	set $h'_{n+1}=$	point of wedge
orr	2	0	h_n	$(h_n + h'_n)/2$	h_n
lor	1	1	h_n	$(h_n + h'_n)/2$	$(h_n + h'_n)/2$
llo	0	2	$(h_n + h'_n)/2$	h'_n	h'_n
rrr	3	0	h_n	$(h_n + h'_n)/2$	h_n
lrr	2	1	h_n	$(h_n + h'_n)/2$	?
llr	2	1	$(h_n + h'_n)/2$	h'_n	?
lll	0	3	$(h_n + h'_n)/2$	h'_n	h'_n

(d) An alternative algorithm

We will consider an alternative analysis of the bisection experiment on the intervals $[h_n, (h_n + h'_n)/2]$ or in $[(h_n + h'_n)/2, h'_n]$ with outcomes r, l or o. There is at most one o outcome, as at most one of the three points can be at the point of the wedge. There are four allowable outcomes containing only r and l. These can be tabulated as shown in table 1, where we give the outcomes in the order $h_n, (h_n + h'_n)/2, h'_n$. The exact position of the point of the wedge is given in the cases where it can be deduced. Figure 2 shows how the outcomes (r, l, o) are related to the position of the cannon relative to the wedge.

## (e) Precision in measurements

Consider the precision of the experiment when measuring the output state. The situation is simple: either the ball is in one tray, or in the other tray, or in neither. Errors in observation do not arise. Now consider the non-trivial ways in which precision hinges on the positions of the cannon and the wedge.

First, we look at the standard procedure for using the machine, where we operate the canon in an experiment. Suppose the wedge position is given and fixed for an experiment. Then there are various different possibilities which we can postulate for the precision of the cannon, and we list some in the order of increasing strength.

(i) The cannon can be set only to within a given precision $\epsilon > 0$.
(ii) The cannon can be set only to within a non-zero, but arbitrarily small, precision.
(iii) The cannon can be set exactly to any given rational number.
(iv) The cannon can be set exactly to any height which can be constructed by a ruler and compass construction from a given unit length.
(v) The cannon can be set exactly to a given computable number.

In our bisection argument above, we used assumption (iii). With a little modification, the weaker assumption (ii) would have sufficed. To do this, we would have to use overlapping intervals, and calling it a 'times 2/3 method' rather than a bisection method would be more accurate.

Assumption (iv) is one which would have appealed to the ancient Greeks—the ruler and compass construction is used to give the length of a measuring rod which is then used to position the cannon.
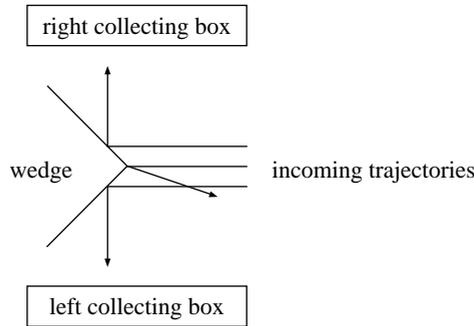
Figure 2. Possible outcomes r, l or o.

Assumption (v) is very strong. It can be combined with a machine with a known rational wedge position (e.g. 0) and the discussion in §5a to test the equality of computable numbers—a form of hyper-computation.

Now, we should examine the positioning of the wedge. The setup of the experiment will be discussed later in §6, but one point is relevant in discussing precision is the randomness of setting the wedge position. The ancient Greek geometers in possibility (iv) could mark off the wedge position on a measuring rod, and use that rod in addition to their standard rod to perform extended ruler and compass calculations. Then those results could be used to set the wedge or cannon positions. From our later discussion on randomness, we know that with positive probability the new rod would have a non-computable length. Thus, we can add a new category of precision, stronger than (iv) and not in general comparable to (v).

(vi) The cannon can be set exactly to any height which can be constructed by a ruler and compass construction from a given unit length and another length which may be a non-computable multiple of the unit.

In pure mathematical terms, this could be described as a field extension of the rationals, and it illustrates the use to which a non-computable number derived from a physical measurement, even if it is randomly generated, can be put.

4. The effect of coordinate systems

We shall now consider the effect of different coordinate systems on the machine. To simplify the description, we will take the machine to be in a vertical plane, though we shall not consider gravity. Consider a skewed coordinate system with a horizontal coordinate, and the vertical coordinate will be measured above the surface of the ground. The cannon is projected horizontally. We will allow the cannon to be positioned at dyadic rationals.

(a) A uniformly sloping hillside

If we have a sloping hillside with an non-computable angle α, we find that we can place the wedge at a rational position x and then compute a non-computable number h using the machine. As the cannon fires horizontally, the initial velocity is
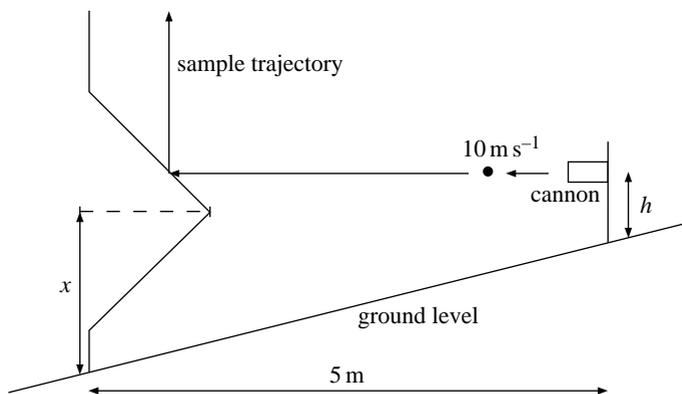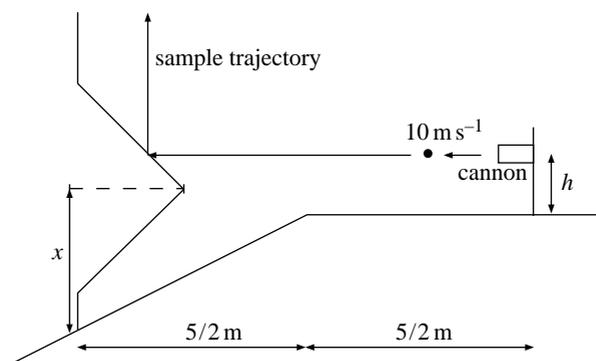
Figure 3. A non-computable slope.

Figure 4. A flat-topped hill.

at a non-computable angle to the coordinate lines, so we simply have a non-computable initial velocity (figure 3).

(b) A flat-topped hill

A more interesting example has a flat hilltop, from which the land slopes down at a non-computable angle α. In this case, the initial velocity is computable, but we can again compute a non-computable number h from a rational setting x (figure 4).

Here, the laws of motion are themselves not computable. The particle follows a straight line trajectory, and any straight line (apart from a vertical one) is not computable in this coordinate system.

5. The effect of refinements

(a) Edges, predictability and non-determinism

In the previous cases, we did not try to model what happened if the particle hit the point of the wedge. The particle would bounce somewhere, but we do not know where, until each experiment is complete. Thus, the machine was non-deterministic.

Now, we shall add to the theory T the assumption of *translation invariance*, i.e. if the system displays a given behaviour at one position where $x = h$, then it displays the same behaviour at every other position $x' = h'$. In particular, the machine behaves predictably and deterministically.

We can split the possible cases when the particle hits the point of the wedge.

(i) *Uniqueness.* The result of a particle hitting the point gives a single predictable result, in that the particle is reflected at a unique speed and angle.
(ii) *Non-uniqueness.* The result is unpredictable, in that there are at least two possible results given identical starting conditions.

Proposition 5.1. *If we have translation invariance and predictability, and the cannon position can be set to the position h, then we can either test $x \leq h$ or $x \geq h$ (possibly both). This depends on the behaviour of the particle on exactly hitting the point of the wedge as follows.*

(i) *The particle ends up in the left collecting box—we can test $x \geq h$.*
(ii) *The particle ends up in the right collecting box—we can test $x \leq h$.*
(iii) *The particle ends up in neither collecting box—we can test $x = h$.*

Proof. If the particle is reflected right or centrally on hitting the wedge, then it cannot enter the left collecting box. Then entering the left box happens if, and only if, $x > h$. Thus, the condition $x \leq h$ is computable by the machine.

If the particle is reflected left or centrally on hitting the wedge, then it cannot enter the right collecting box. Then entering the right box happens if, and only if, $x < h$. Thus, the condition $x \geq h$ is computable by the machine. ∎

Note that we cannot escape the consequences of this by only allowing h to be a dyadic rational. The proposition $x \leq 1/2$ is not computable in the standard algorithmic models of computation for a general computable number x (Pour-El & Richards 1989; Weihrauch 2000).

(*b*) *Spherical projectiles*

Next, the reader may try to escape this by questioning the use of point particles—after all, all the observed projectiles have size (we will avoid a discussion of quantum mechanics at this point and stick to such projectiles as cannon balls, etc.). The reader may observe that if we use spherical projectiles, and scatter off a wedge with a sharp point, we get *continuity* of scattered angle. In figure 5, the dashed lines and incoming velocity are horizontal and the cannonball has radius r.

The scattered angle is given by the following formula, which gives a continuous function:

$$\text{scattered angle} = \begin{cases} 180^{\circ} - 2\alpha & h > r \cos \alpha, \\ 2 \arcsin(h/r) & |h| \leq r \cos \alpha, \\ -180^{\circ} + 2\alpha & h < -r \cos \alpha. \end{cases}$$

The reader may now think that having finite-sized spheres as particles, and if necessary rounding off a few corners, will get rid of the undefined trajectories,
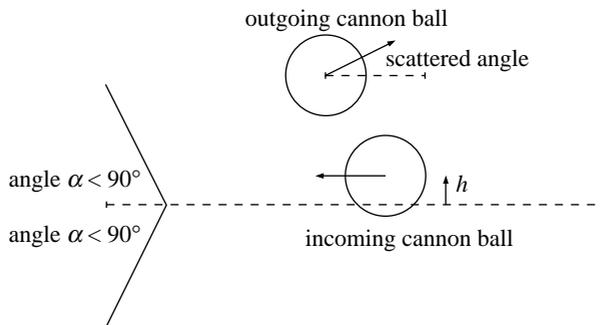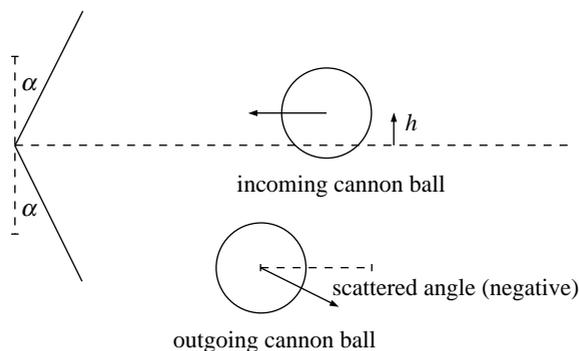
Figure 5. A finite-sized projectile.

Figure 6. Discontinuous scattering.

and ensure continuity. However, consider the machine in figure 6 with a radius r cannon ball. Note that to avoid secondary collisions, we assume that $\alpha < 30°$.

In this case, we have a discontinuous scattered angle, as is shown by the formula

$$\text{scattered angle} = \begin{cases} -2\alpha & h > 0, \\ 0 & h = 0, \\ 2\alpha & h < -0. \end{cases}$$

So, what is the scattered angle for $h=0$? By symmetry, it would contact both lines of the wedge at the same time. If both the sides had the same mechanical properties, the resulting forces (transmitted along the normals to the lines) would be equal, so the ball would bounce straight back, i.e. the scattered angle would be 0.

This has absolutely nothing to do with the sharp corner where the lines meet and is just drawn for simplicity. As the sphere has a finite size, it cannot approach the corner within a certain distance, so we can round the corner in any manner we please without altering the result.

Hence, just what in the kinematics has replaced the infinitesimal size of the point particle and allows us to have this discontinuity? It is the rigidity assumption, which says that the spherical ball does not deform on impact—thus still giving us arbitrarily small spatial resolution.
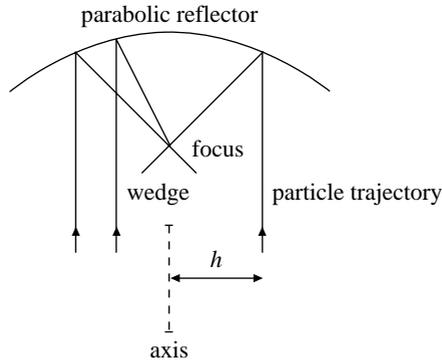
Figure 7. Focusing trajectories.

Note that if the surfaces have friction then, for projectiles of non-zero size, collisions with the rigid walls of the wedge produce some angular momentum. The effect of this is to alter the final velocity and angle of reflection. This can be accommodated into the design of a scatter machine.

(c) Focusing

So far, we have seen problems caused by a single value of a parameter. At a given position h for the cannon, the cannon ball hits the point of the wedge, or something similar. For all other values, the system behaves in a much better fashion. However, this is also misleading and there are systems which display bad behaviour for a range of values of parameters, including non-zero length intervals.

A parabolic reflector will reflect point particles which are approaching parallel to its axis onto a common focus (figure 7). If we position the point of a wedge at the focus, we can have potentially undefined behaviour for a large range of values of h. Similarly, for an ellipse, we can position a cannon at one focus, and the point of the wedge at another focus. Even if we only allow circles, we can position a cannon on a circular rail, so that if it fires towards the centre of the circle and at the centre of the circle we can put the point of a wedge.

(d) Problem of point particle projectiles

What is a point particle, and do we know how it will behave? One idea is that it is the limit of a finite sized projectile, as the size tends to zero (keeping the mass fixed, if that is required for the dynamics).

In calculus, the derivative of a function is defined by taking the gradient of line joining two points on the graph, and then letting the size of the line tend to zero. For a differentiable function, the limiting gradient is independent of exactly how the length of the line is shrunk to zero. Let us consider figure 8, where we let the size of the projectile tend to zero. The triangle has a horizontal top, and the other sides are of equal length, making an angle of β with the horizontal. The height of the triangle is $2d$ and h is the height of the midpoint above the horizontal through the point of the wedge. We assume that β is greater than α, the angle of the wedge.
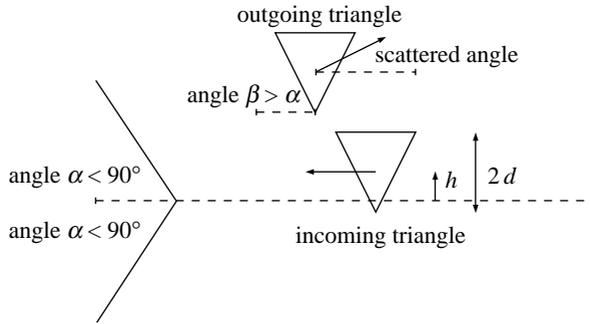
Figure 8. A triangular projectile.

To simplify matters, we impose the constraint that the top edge of the triangle remains horizontal. Otherwise, we would have spinning triangles and moments of inertia to take into account. We assume (as before) that the forces on impact are transmitted perpendicularly to the lines. Then, the scattered angle is

$$
\text{scattered angle} = \begin{cases}
180^{\circ} - 2\alpha & h > d, \\
?? & h = d, \\
180^{\circ} - 2\beta & -d < h < d, \\
?? & h = -d, \\
-180^{\circ} + 2\alpha & h < -d.
\end{cases}
$$

The problematical cases shown are where a vertex of the triangle hits the point of the wedge. Taking the limit of this as d tends to zero gives a very different result from taking the limit of the case in figure 5 as the radius tends to 0. The underlying difference is that the system is no longer symmetric under reflecting in the horizontal dotted line through the point of the wedge.

In fact, the situation is even worse than this example might suggest, as can be seen by further considering the spherical projectile case in figure 5. If we were to measure the distance h not to the middle of the ball, but to a position a fixed way up the sphere (e.g. 3/4 of the way from bottom to top), then we would obtain different limiting behaviour. A rather imprecise, but illuminating, way of thinking about this is that the fraction (e.g. $6/4r$) becomes an infinitesimal number as $r \rightarrow 0$, which has a dramatic effect on the scattered angle. As specifying a real number does not specify this 'infinitesimal' component, we have an uncertainty which reveals itself as a probabilistic outcome to the experiment.

6. Non-computability and initial positions

In what sense, and to what extent, can the scatter machine, and hence the subtheory T, perform hyper-computations? Quite different views on the technical nature of hyper-computation are possible (Loff & Costa in press).

Our scatter machine calculates the position of the wedge to arbitrary accuracy. To be more precise, we have a procedure for the scatter machine that, for *every* position of the wedge, can generate an unending sequence of dyadic intervals which

locates the position of the wedge with increasing accuracy. The infinite sequence of dyadic intervals converges to a real number that is unique to the position of the wedge. This is proved on the basis of the Newtonian subtheory T, supported by some Euclidean geometry and classical logic. For the moment, let us say that each real number produced by the scatter machine is a T-computable number.

Let us begin with a simple question: are there T-computable real numbers that are not Turing computable real numbers? The answer depends upon the initial position of the wedge.

Consider the following general questions.

(i) *Can any point on the one-dimensional scale be the position of an object?*
(ii) *Does each point on the one-dimensional scale correspond with a real number in* [0, 1]*?*

The *standard* answers to these questions in Newtonian mechanics are: yes. This implies that all numbers in [0, 1] are T-computable numbers and so, by cardinality, most T-computable numbers that are not Turing computable. This standard interpretation is used in the formulation of the theorems in §1.

However, consider the *physical* process of positioning the wedge.

(iii) *Can the wedge be positioned at some points that are measured by non-computable numbers on a one-dimensional scale*?

If the answer to question (iii) is positive, then we know that the scatter machine is capable of some stronger form of hyper-computation.

To reason about and understand the operation of scatter machine, we had *no* need for a discussion on the position of the wedge: its operation is defined uniformly for *all* positions. We did not need anything in the theory T to explain the positioning of the wedge. But does the physics rule out, or ensure, that the wedge can be positioned at a non-computable number?

First, consider some kind of deterministic physical process. Suppose we add to the system a procedure to position the wedge. This imposes constraints on the wedge positions, namely: each wedge position is described by a finite sequence of actions generated by following the procedure. A rigorous way to analyse this idea is to suppose that a procedure can be written down as a well-formed formula in a formal language with a countable syntax. Since the set of all well-formed formulae is a countable set, the set of all procedures and, hence, positions of the wedge is countable.

Now, if the resulting positions are all computable then the procedures must be equivalent to algorithms in some sense. By exploring the idea of a systematic method for positioning the wedge, we may be able to rule out a non-computable position, or push the source of non-computability somewhere else (e.g. to input parameters). However, it is artificial to introduce the theory of algorithms into the physical picture, as it would extend the physical theory T prematurely and inappropriately.

Second, consider the idea that physically the position can be anywhere, and so can be chosen at random. The easiest way to think about this non-deterministic case is to appeal to probability. Suppose we employ a random process to move the wedge. The outcome of such a process would be a probability distribution for the position of the wedge. On the basis of many physical examples, we expect the distribution to be continuous.

Consider the absence of non-computable points: would it be reasonable for such a probability distribution of a random variable X to have a probability of one of giving a computable number? We shall argue that for a practical scatter machine, this is not the case and it is almost inevitable that we could place the wedge at a non-computable position.

Lemma 6.1. *A Borel probability measure on the real line which assigns a probability of zero to the non-computable real numbers must be discontinuous for the usual metric topology on the computable numbers.*

Proof. First note that if closed sets are measurable (the usual Borel assumption on probabilities on the real line), then we can assign a probability to X taking a single real value. The set of computable numbers is countable, and since the total probability is one, there must be a computable number c with $P[X=c]>0$ (the probability of X taking the value c). If the probability distribution were continuous, there would be a $\delta>0$ so that, for all computable numbers c' with $|c'-c|<\delta$, we have $P[X=c']>P[X=c]/2$. But there are infinitely many such computable numbers c' with $|c'-c|<\delta$, so using the additivity of the probabilities, we get $P[X\in(c-\delta,\ c+\delta)]>1$, a contradiction. ∎

Such a discontinuous probability distribution would be quite perverse. The problem is that, given a program in a black box for computing a real number x to any specified accuracy, we cannot compute its probability $P[X=x]$, as it is a discontinuous function in the metric topology (Ceitin's theorem; Weihrauch 2000). Of course, if we were given the Gödel number of the program, we might have a formula to work out the probability from that, but that construction takes us well outside the intuitive idea of real number. Hence, in this sense, if we restrict the probability distribution $P[X]$ to computable numbers, it can only be done at the cost of making the probability distribution non-computable.

If we have access to virtually any continuous random variable in the world where the machine lives, we can find a non-computable number with a probability of one. For example, if there was a gas undergoing Brownian motion, we could confine some in a container with a piston at one end, and let the piston push the wedge for a fixed amount of time. If friction involved a continuous random variable, we could kick the wedge and see where it came to rest.

Of course, there are ways of finding non-computable numbers with a Turing machine, if we add a discrete random number generator (e.g. a fair six-sided die). We could compute a number by taking a binary expansion $0\cdot1011011000100\ldots$, where the nth binary place is 0 if throwing the die for the nth time gives 1, 2, 3, and one of the results is 4, 5, 6. This gives a uniform probability distribution—the number x has an equal chance of being in $[0,\ 1/2]$ or in $[1/2,\ 1]$, and then the next binary place again assigns an equal probability of being in half of one of those intervals, and so on. (Remember that if X is uniformly distributed on $[0,\ 1]$, then the probability of it being in a particular subset of $[0,\ 1]$ is just given by the length of the set.) For some given purpose, we can compute and store x to a given number of binary places, and later, if required, add more binary places (again entirely randomly). We cannot access all of x in a finite time, and so cannot run comparisons such as $\geq$ with it. However, in the scatter machine, the result of a probabilistic process is stored

in the position of the wedge, and we can run comparisons with other numbers (such as $\geq$) as described previously. Thus, we have storage of infinite precision numbers implemented in a finite time, by using positions.

7. Concluding remarks

Experimental computation is a basic idea that connects computation with any part of science. Experimental computation is not well understood even in the case of kinematics, possibly the simplest physical theory. The questions asked in §1 are open. Through work on examples, we are developing a methodology to *play* with physical theories and their connections with computational ideas, in order to explore the relationship between experimental and algorithmic procedures.

We have shown that the scatter machine, based on a simple Newtonian theory, can calculate any point on the unit scale to arbitrary accuracy. The scatter machine has finite size and mass, but operates with time and energy linear in the degree of accuracy. The machine is non-deterministic and non-computability is not coded and hidden in its structure or operation. Here, non-computability is to be found in its initial state, i.e. the position of the wedge. We have argued that the physical theory cannot rule out these non-computable positions and, moreover, that they are physically inevitable. Hence, the scatter machine is a Newtonian machine that can perform some form of hyper-computation, according to physical theory, and it does not have the *deus ex machina* property as found in many devices. The existence of the scatter machine advances theoretically the reflections of Geroch & Hartle (1986) on the computability of constants and measurements in physical experiments.

These results complement and improve on the exotic examples in our earlier papers. In Beggs & Tucker (2004), we showed there exist kinematic systems, which operate under the theory of (i) Newtonian mechanics and (ii) relativistic mechanics, that can decide the membership of *any* subset A of the set $\mathbb{N} = \{0, 1, 2, ...\}$ of natural numbers. The systems were two-dimensional bagatelles with single particles that needed *unbounded* space, time and energies for their computations. In Beggs & Tucker (2007), we improved on these bagatelle machines by constructing new Newtonian machines that are marble runs that each require *bounded* space, time, mass and energy to decide $n \in A$ for all n. These later examples need the assumption that space can be infinitely divided into smaller and smaller units, a valid assumption in Euclidean geometry and Newtonian mechanics not used on the bagatelles or here.

Finally, let us note some directions for further research to add to those given in earlier papers.

The discussion of the machine in §6 can be taken further technically and philosophically by axiomatizing the subtheory T, exploring the interface between numbers and their representation via physical quantities, and the logical foundations of the arguments.

The scatter machine can generate all the digits or symbols in an infinite representation of a real number from $[0, 1]$ to arbitrary accuracy. The scatter machine operation resembles an implementation of the process of using oracle in relativized computability theory and complexity theory. Felix Costa has pointed out that this aspect of the scatter machine suggests interesting connections with

structural complexity theory. It is worth investigating the complexity of computations by the scatter machine and comparing its power with that of other complexity models.

The basic principles of the scatter machine may suggest new machines based on other physical theories, such as light.

We thank our two referees for their useful comments on this paper.

References

Beggs, E. J. & Tucker, J. V. 2004 Computations via experiments with kinematic systems. Research Report no. 4.04, Department of Mathematics, University of Wales Swansea or Technical Report no. 5-2004, Department of Computer Science, University of Wales Swansea.

Beggs, E. J. & Tucker, J. V. 2006 Embedding infinitely parallel computation in Newtonian kinematics. *Appl. Math. Comput.* **178**, 25–43. (doi:10.1016/j.amc.2005.09.068)

Beggs, E. J. & Tucker, J. V. 2007 Can Newtonian systems, bounded in space, time, mass and energy compute all functions? *Theor. Comput. Sci.* **371**, 4–19. (doi:10.1016/j.tcs.2006.10.010)

Brockett, R. W. 1991 Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems. *Linear Algebra Appl.* **146**, 79–91. (doi:10.1016/0024-3795(91)90021-N)

Brockett, R. W. 1994 Dynamical systems and their associated automata. In *Systems and networks: mathematical theory and applications* (eds U. Helmke, R. Mennicken & J. Saurer), pp. 49–69. Berlin, Germany: Akademie Verlag.

Bush, V. 1931 The differential analyser. A new machine for solving differential equations. *J. Franklin Inst.* **212**, 447–488. (doi:10.1016/S0016-0032(31)90616-9)

Cooper, S. B. & Odifreddi, P. 2003 Incomputability in nature. In *Computability and models: perspectives east and west* (eds S. B. Cooper & S. S. Goncharov), pp. 137–160. Dordrecht, The Netherlands: Kluwer Academic/Plenum Publishers.

da Costa, N. C. A. & Doria, F. A. 1991*a* Undecidability and incompleteness in classical mechanics. *Int. J. Theor. Phys.* **30**, 1041–1073. (doi:10.1007/BF00671484)

da Costa, N. C. & Doria, F. A. 1991*b* Classical physics and Penrose's thesis. *Found. Phys. Lett.* **4**, 363–373. (doi:10.1007/BF00665895)

Davis, M. 2004 The myth of hypercomputation. In *Alan Turing: life and legacy of a great thinker* (ed. C. Teuscher), pp. 195–211. Berlin, Germany: Springer.

Geroch, R. & Hartle, J. B. 1986 Computability and physical theories. *Found. Phys.* **16**, 533–550. (doi:10.1007/BF01886519)

Graça, D. S. & Costa, J. F. 2003 Analog computers and recursive functions over the reals. *J. Complexity* **19**, 644–664.

Griffor, E. (ed.) 1999 *Handbook of computability theory*, vol. 140. Studies in logic and the foundations of mathematics. Amsterdam The Netherlands: Elsevier.

Hey, A. J. G. (ed.) 2002 *Feynman and computation.* Reading, MA: Westview Press.

Holden, A. V., Tucker, J. V., Zhang, H. & Poole, M. 1992 Coupled map lattices as computational systems. *Am. Inst. Phys. Chaos* **2**, 367–376. (doi:10.1063/1.165879)

Kreisel, G. 1974*a* A notion of mechanistic theory. *Synthese* **29**, 9–24. (doi:10.1007/BF00484949)

Kreisel, G. 1974*b* Review of Pour-El and Richards. *J. Symb. Logic* **47**, 900–902. (doi:10.2307/2273108)

Lipshitz, L. & Rubel, L. A. 1987 A differentially algebraic replacement theorem. *Proc. Am. Math. Soc.* **99**, 367–372. (doi:10.2307/2046643)

Loff, B., & Costa, J. F. In press. Five views of hypercomputation, Department of Mathematics, Technical University of Lisbon, 2006. *Int. J. Unconventional Comput.*

Moore, C. 1990 Unpredictability and undecidability in dynamical systems. *Phys. Rev. Lett.* **64**, 2354–2357. (doi:10.1103/PhysRevLett.64.2354)

Moore, C. 1996 Recursion theory on the reals and continuous time computation. *Theor. Comput. Sci.* **162**, 23–44. (doi:10.1016/0304-3975(95)00248-0)

Nielsen, M. A. & Chuang, I. L. 2000 *Quantum computation and quantum information*. Cambridge, UK: Cambridge University Press.

Odifreddi, P. 1989 *Classical recursion theory*, vol. 129. Studies in logic and the foundations of mathematics. Amsterdam, The Netherlands: North-Holland.

Pour-El, M. B. 1974 Abstract computability and its relations to the general purpose analog computer. *Trans. Am. Math. Soc.* **199**, 1–28. (doi:10.2307/1996870)

Pour-El, M. B. & Richards, J. I. 1979 A computable ordinary differential equation which possesses no computable solution. *Ann. Math. Logic* **17**, 61–90. (doi:10.1016/0003-4843(79)90021-4)

Pour-El, M. B. & Richards, J. I. 1981 The wave equation with computable initial data such that its unique solution is not computable. *Adv. Math.* **39**, 215–239. (doi:10.1016/0001-8708(81)90001-3)

Pour-El, M. B. & Richards, J. I. 1983 Computability and noncomputability in classical analysis. *Trans. Am. Math. Soc.* **275**, 539–560. (doi:10.2307/1999038)

Pour-El, M. B. & Richards, J. I. 1989 *Computability in analysis and physics*. Berlin, Germany: Springer.

Shannon, C. 1941 Mathematical theory of the differential analyser. *J. Math. Phys.* **20**, 337–354.

Siegelmann, H. T. 1999 *Neural networks and analog computation: beyond the Turing limit*. Boston, MA: Birkhäuser.

Smith, W. D. 2006 Church's thesis meets the *N*-body problem. *Appl. Math. Comput.* **178**, 154–183. (doi:10.1016/j.amc.2005.09.077)

Stoltenberg-Hansen, V. & Tucker, J. V. 1995 Effective algebras. In *Handbook of logic in computer science. Volume IV*: *sematic modelling* (eds S. Abramsky, D. Gabbay & T. Maibaum), pp. 357–526. Oxford, UK: Oxford University Press.

Stoltenberg-Hansen, V. & Tucker, J. V. 1999*a* Computable rings and fields. In *Handbook of computability theory*, vol. 140 (ed. E. R. Griffor), Studies in logic and the foundations of mathematics, pp. 363–447. Amsterdam, The Netherlands: Elsevier.

Stoltenberg-Hansen, V. & Tucker, J. V. 1999*b* Concrete models of computation for topological algebras. *Theor. Comput. Sci.* **219**, 347–378. (doi:10.1016/S0304-3975(98)00296-5)

Stoltenberg-Hansen, V. & Tucker, J. V. 2003 Computable and continuous partial homomorphisms on metric partial algebras. *Bull. Symb. Logic* **9**, 299–334. (doi:10.2178/bsl/1058448675)

Thompson, W. & Tait, P. G. 1880 *Treatise on natural philosophy*, pp. 479–508, 2nd edn. Cambridge, UK: Cambridge University Press. Part I, Appendix B.

Tucker, J. V. & Zucker, J. I. 1999 Computation by while programs on topological partial algebras. *Theor. Comput. Sci.* **219**, 379–421. (doi:10.1016/S0304-3975(98)00297-7)

Tucker, J. V. & Zucker, J. I. 2000 Computable functions and semicomputable sets on many sorted algebras. In *Handbook of logic for computer science. Volume V*: *logic and semantic methods* (eds S. Abramsky, D. Gabbay & T. Maibaum), pp. 317–523. Oxford, UK: Oxford University Press.

Tucker, J. V. & Zucker, J. I. 2004 Abstract versus concrete computation on metric partial algebras. *ACM Trans. Comput. Logic* **5**, 611–668. (doi:10.1145/1024922.1024924)

Weihrauch, K. 2000 *Computable analysis, an introduction*. Heidelberg, Germany: Springer.

Weihrauch, K. & Zhong, N. 2001 Is the Schrödinger propagator Turing computable? In *Computability and complexity in analysis*, vol. 2064 (eds J. E. Blanck, V. Brattka & P. Hertling), Springer lecture notes in computer science, pp. 369–377. Berlin, Germany: Springer.

Weihrauch, K. & Zhong, N. 2002 Is wave propogation computable or can wave computers beat the Turing machine? *Proc. Lond. Math. Soc.* **85**, 312–332. (doi:10.1112/S0024611502013643)

Wolfram, S. 2002 *A new kind of science*. Champaign, IL: Wolfram Media.

Xia, Z. 1992 The existence of non-collision singularities in the *n*-body problem. *Ann. Math.* **135**, 411–468. (doi:10.2307/2946572)

Yao, A. 2003 Classical physics and the Church Turing thesis. *J. ACM* **50**, 100–105. (doi:10.1145/602382.602411)