PRIFYSGOL CYMRU ABERTAWE
UNIVERSITY OF WALES SWANSEA

UNIVERSITY OF WALES SWANSEA

REPORT SERIES

Experimental computation of real numbers by Newtonian machines
by

E.J. Beggs and J.V. Tucker

Report # CSR 9-2006

Computer Science
Gwyddor Cyfrifiadur

Experimental computation of real numbers by Newtonian machines[1]

E.J. Beggs[2] and J.V. Tucker[3]

University of Wales Swansea,
Singleton Park,
Swansea, SA3 2HN,
United Kingdom

Abstract

Following a methodology we have proposed for analysing the nature of experimental computation, we prove that there is a 3-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 $n-th$ approximation of x takes place in $O(n)$ time with $O(n)$ energy.* We describe a set of 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 for 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. The subject has been guided by its foundational origins and its applications to mathematics and computer science.

In the past two decades or so the ideas that *information and computation are physical processes* have emerged from different corners of logic, physics and computer science and begun to exert influence on Computabilty Theory. In mathematical logic, Kreisel drew

[1]To refer to this paper cite as one of the following: Research Report 06.11, Department of Mathematics, University of Wales Swansea, June 2006 or Technical Report CSR9-2006, Department of Computer Science, University of Wales Swansea, June 2006.

[2]Department of Mathematics. Email: e.j.beggs@swansea.ac.uk

[3]Department of Computer Science. Email: j.v.tucker@swansea.ac.uk

attention to the question "Is physical behaviour computable?" in papers, such as Kreisel [14], which stimulated research in Computable Analysis, including Pour El and Richards [21, 22, 23, 24]. In physics, work of Feynman on the physical basis of computation have led to subjects such Quantum Information Processing [19, 11]. Despite a large disparate literature, it is hard to answer the 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. This is because, in contrast to algorithmic computation, the idea of experimental computation is not understood. A rigorous 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 are based firmly on the precise formulation of physical theories [2, 3].

Here we a fix a small sub-theory T of Newtonian Kinematics and present a simple example of a Newtonian machine based on T that performs hyper-computation. Specifically, we consider an experimental procedure that computes a real number. The computations of physical quantities described by real numbers was first discussed in Geroch and Hartle [9], with physical constants in mind. We prove the following theorem.

Theorem 1.1. *There exists a 3-dimensional Newtonian kinematic machine which can compute the position of any point on a 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*

$$x \in [p_n, q_n] \text{ and } |p_n - q_n| < \tfrac{1}{2^n}$$

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 $n-th$ approximation takes place in linear time $O(n)$.

Energy: The calculation of the $n-th$ approximation uses linear $O(n)$ energy.

Corollary 1.2. *Every point in the unit interval $[0, 1]$ is computable by experiment by machines based on simple subtheories of Newtonian kinematics.*

Thus, our scatter machine shows that the existence of hyper-computers is consistent with a physical subtheory T of Newtonian kinematics. The crucial point for hyper-computation is the extent to which the assumption *any point on a 1-dimensional scale can be the position of an object*, and its variants, depend on T. Under reasonable general physical assumptions, we show there is a positive 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 types 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. These inconsistent refinements of T explain the need for us to keep the scatter machine non-deterministic. This type of refinement and dissection of kinematic theories is required by the working methodology.

With some modest assumptions on the scattering machine we can prove the following (Proposition 5.4).

Theorem 1.3. *A scatter machine that is deterministic and satisfies a simple translation invariance property can computably decide by experiment, given any points x, y in the unit interval $[0, 1]$, whether $x \leq y$, or whether $y \leq x$, 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 the question, many 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 this. In Section 2 we describe the methodology of our approach. In Section 3 the construction of a simple type of scatter machine is given. In Section 4 we consider the role of coordinate systems. In Section 5 we give further examples that refine the scatter machine and explore non-determinism. Finally, in Section 6 we offer some concluding remarks.

This paper is a sequel to Beggs and Tucker [1, 2, 3]. Only elementary Newtonian mechanics is needed to follow our technical arguments here. As background, it is helpful if the reader is familiar with the literature on hyper-computation (see: [1, 5]), with the theory of the functions computable by algorithms on discrete data (see: Odifreddi [20], Griffor [13], Stoltenberg-Hansen and Tucker [29, 30]) and its extension to continuous data (see: Pour-El and Richards [25], Stoltenberg-Hansen and Tucker [31], Tucker and Zucker [35, 36], Weihrauch [37]).

2 Principles for the Study of Experimental Computation

We consider a set of principles for investigating experimental computation and hypercomputation in a rigorous way, first used in Beggs and Tucker [2, 3].

2.1 Experimental Computation

Definition *Experimental computation* consists of an *experimental procedure* which is a specification for a finite or infinite process made of primitive experimental actions that are applied to a physical system called the *equipment*. The actions allow construction, alteration, initialisation and observation of the equipment. The procedure schedules the actions, step by step, in time measured by clocks. Briefly, *Experimental computation = Experimental procedure + Equipment.*

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.
We can expect to compute functions on continuous data, such as the set $\mathbb{R}$ of real numbers.

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 [23, 24, 25, 15, 38, 39, 32], and in technologies for designing machines, e.g., in the 19th century mechanical systems of analogue computers ([33, 4, 26, 21, 16, 10]. Computation is a theme in contemporary studies of emergent behaviour (e.g., cellular lattices and neural networks [12, 27, 40, 11]) and of quantum computers ([19, 11]).

2.2 Methodological Principles

Physical theories play a complicated role in experimental computation. In seeking answers to the questions in the Introduction, we should use a physical theory to
(1) define precisely the class K of physical systems under investigation;
To do this the concepts and laws of the theory will be used to
(2) design and construct the systems in K and define their primitive experimental operations;
(3) explain and prove properties of their operation and behaviour; and, hence,
(4) validate experimental computations by the systems of K.
If a hyper-computer has been found in K then we should use the theory to
(5) evaluate the physical credibility of the system or, possibly,
(6) reveal weaknesses in the theory.
Finally, in each case, we should use the theory to
(7) 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 and Tucker [2, 3], the following methodology based on four principles and stages for an investigation of some type of experimental computation.

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.*

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, emphasising conceptual precision and mathematical rigour. What is contained in T determines the set of possible experimental procedures and equipment. (Elsewhere, in Beggs and Tucker [3], we discuss how to derive, from T, formal languages to specify both experimental procedures and equipment.) *It does not matter whether we think of these simple theories as true, or roughly applicable, or know them 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 confusing and disputed then it is not surprising that Relativistic and Quantum examples are not definitive.

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.*

Physical theories do not contain notions of computation. The computation of functions and sets must be abstracted from, or represented by, 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 *simple physical subtheories.*

From our reading of the literature, systems perform hyper-computation by 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 a careful review. Hyper-computation by courtesy of such an embedding is a property of devices that we have called *deus ex machina* [1]: 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 [14, 15] to Davis [8]. Even after forty 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 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 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. But it is not obvious how to control non-computability by physical conditions.

Principle 4. Reviewing and Refining the Physical Theory: *Determine the physical relevance of the systems of interest by reviewing the valid scope or truth 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 the 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 validated by experimental facts. For any theory T there are restrictions that define that part of the theory that 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.

2.3 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

(a) projection of particles in space with arbitrary velocity, and

(b) observation of the position of particles at certain times.

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

Newtonian kinematics has been used in many discussions of computation and hypercomputation such as Moore [17], da Costa and Doria [6, 7], Yao [41] and Smith [28]. We have surveyed the quest for non-computable physical behaviour in our [1].

3 The Scattering Machine

3.1 A simple kinematic theory T

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

1. Point particles obeying Newton's laws of motion in the two dimensional plane.
2. Straight line barriers, with perfectly elastic reflection of particles.
3. Cannons, which can be moved in position, and can project a particle with a given velocity in a given direction.
4. Particle detectors, capable of telling if a particle has crossed a given region of the plane.

5. A clock to measure time.

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 when we consider several possible refinements.

3.2 Specification of 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.

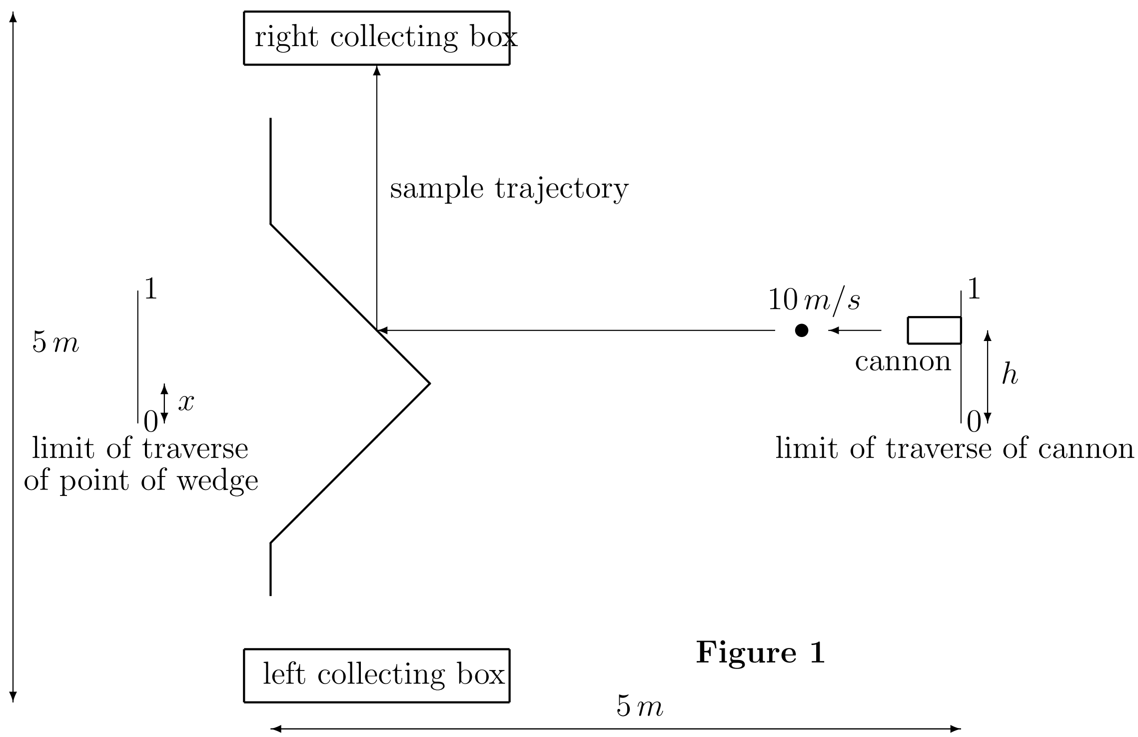

Figure 1

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 meters, and the single headed arrows show a sample trajectory of the particle after being fired by the cannon.

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. The particle will enter one of the two boxes within 1 second of being fired. 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 controvercial. We shall suppose that the particle could enter either box, or not enter any box. This means that the scatter machine is non-deterministic.

3.3 Operation of scattering machine

Suppose that x, the position of the point of the wedge, is fixed. For a given cannon position h, there are 3 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 make 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 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 follow the procedure:

(1) If $n = N$, stop.

(2) Fire the cannon at positions h_n, h'_n and $(h_n + h'_n)/2$.

(3) If the outcomes (one of r,l,o above) 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 (7).

(4) If the outcomes (one of r,l,o above) 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 (7).

(5) If all three outcomes are right (case r above), set $h_{n+1} = h_n$ and $h'_{n+1} = (h_n + h'_n)/2$ and go to (7).

(6) If all three outcomes are left (case l above), set $h'_{n+1} = h'_n$ and $h_{n+1} = (h_n + h'_n)/2$ and go to (7).

(7) Increment n and go to (1).

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 goes 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 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,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 doesn't matter which half-interval we choose.

3.4 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 follows, 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.

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

Figure 2 shows how the outcomes (l,r,o) are related to the position of the cannon relative to the wedge.

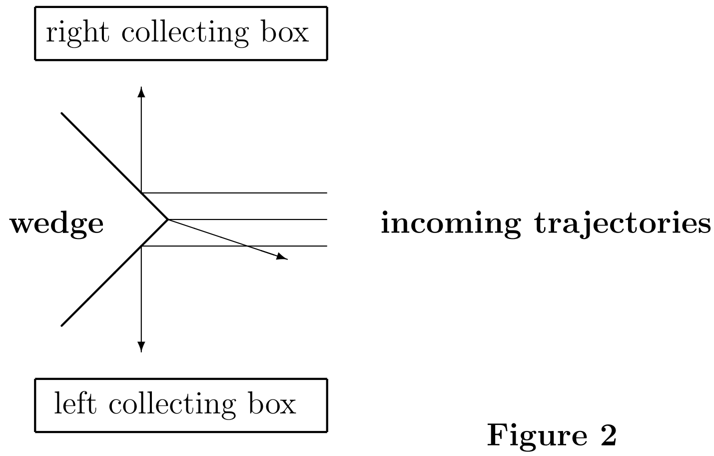

Figure 2

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.

4.1 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.

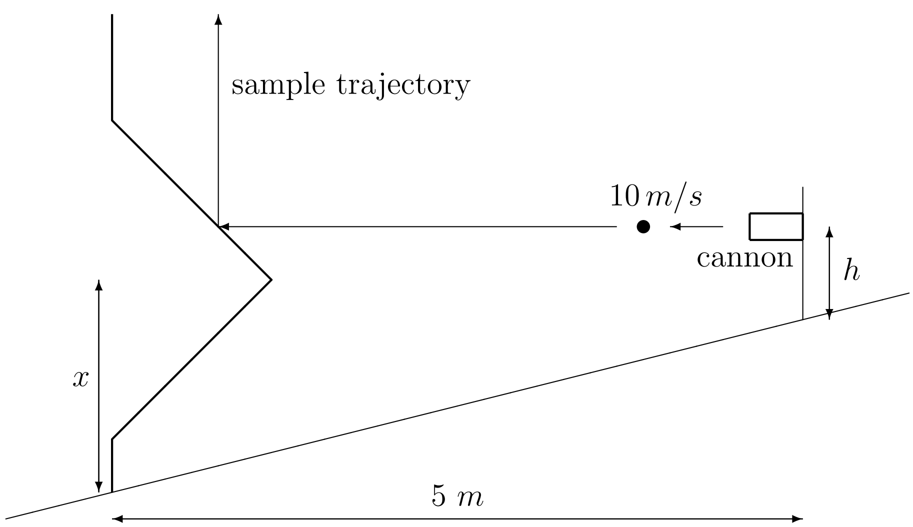

Figure 2

As the cannon fires horizontally, the intial velocity is at a non-computable angle to the coordinate lines, so we simply have a non-computable initial velocity.

4.2 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.

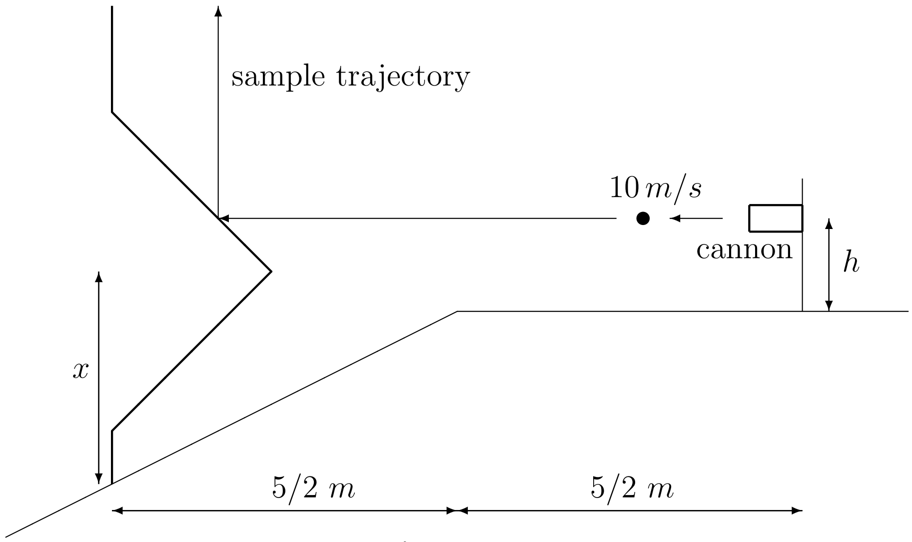

Figure 3

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

5.1 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. 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 behavoiur at every other position $x' = h'$.

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).*

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. $\square$

Note that we cannot escape the consequences of this by only allowing h to be a dyadic rational. The proposition $x \leq \frac{1}{2}$ is not computable for a general computable number x ([25], [37]).

5.2 Spherical projectiles

Next the reader may try to escape this by questioning the use of point particles - after all, all 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, that we get *continuity* of scattered angle. In Fig. 3 the dashed lines and incoming velocity are horizontal, and the cannonball has radius r.

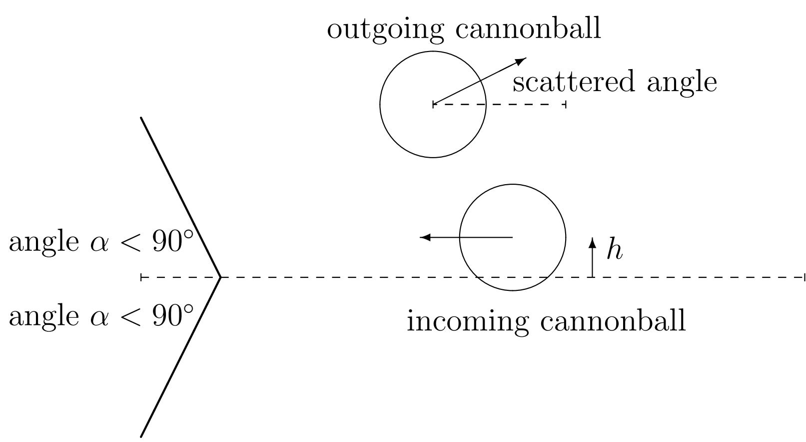

Figure 3

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, and ensure continuity. However consider the following example for scattering our radius r cannonball. Note that to avoid secondary collisions, we assume that $\alpha < 30^\circ$.

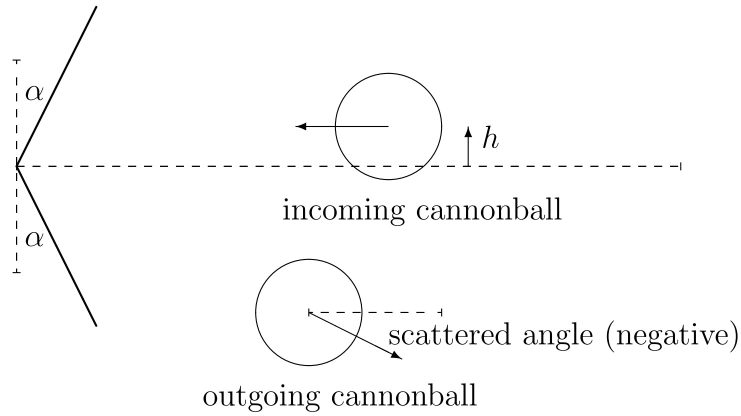

Figure 4

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 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 zero.

This has absolutely nothing to do with the sharp corner where the lines meet, this 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.

5.3 Focusing

So far, we have seen problems caused by a single value of a parameter. At a given position h for the cannon, the cannonball 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, 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.

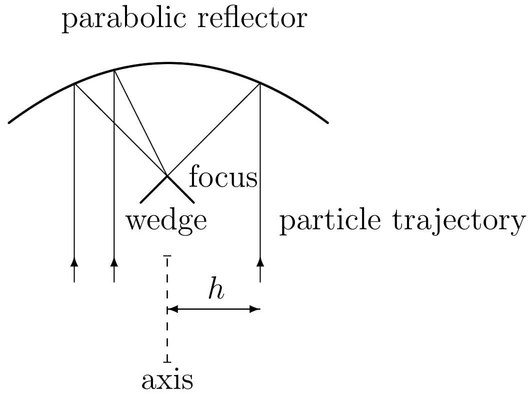

Figure 5

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 fires towards the center of the circle, and at the center of the circle we can put the point of a wedge.

5.4 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 tends 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 letting the size of a given finite sized triangular 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 $2\,d$, 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.

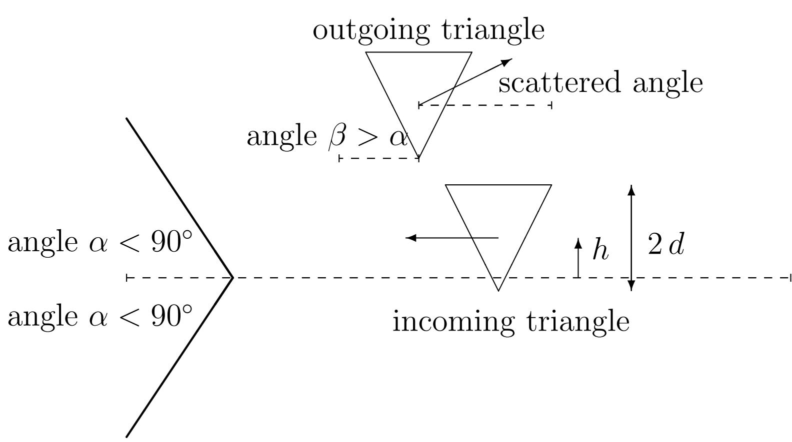

Figure 6

To simplify matters, we impose the constraint that the top edge of the triangle remain 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 3 as the radius tends to zero. 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 3. 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. $\frac{3}{4}$ 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. $\frac{6}{4}\,r$) becomes an infinitesimal number as $r \to 0$, which has a dramatic effect on the scattered angle. As specifying a real number does not specify this 'infinitesimal' component, we have an uncertanty which reveals itself as a probabalistic outcome to the experiment.

5.5 Non-computable initial positions

Our scatter machine calculates the position of the wedge. For it to qualify as a hyper-computer we need to address the following questions:

1. *Can any point on a 1-dimensional scale be the position of an object?*

2. *Can some non-computable points on a 1-dimensional scale be the position of an object?*

Specifically, to show that the existence of a hyper-computer is consistent with a physical subtheory T of Newtonian kinematics and that the answer to question 2 is Yes.

How can we ensure that the wedge is positioned at a non-computable number? It we try to calculate the position of the wedge, then we merely push the source of non-computability somewhere else and introduce the theory of algorithms into the physical picture prematurely and inappropriately.

From a physical point of view, the easiest way is to appeal to probability. Suppose we employ a random process to move the wedge, for example Brownian motion (where the individual atoms in the air collide with the wedge and give it minute pushes). 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 probability one of giving a computable number? We shall argue that for a practical scatter machine this is not the case and that it is almost inevitable that we could place the wedge at a non-computable position.

Lemma 5.2. *A Borel probability measure on the real line which assigns probability zero to the non-computable reals 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 by 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, [37]). 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. So, 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 probability 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 6 sided die). We could compute a number by taking a binary expansion $0{\cdot}1011011000100\ldots$, where the nth binary place is zero if throwing the die for the nth time gives $1, 2, 3$, and one of the result is $4, 5, 6$. This gives a uniform probability distribution - the number x has an equal chance of being in $[0, \frac{1}{2}]$ or in $[\frac{1}{2}, 1]$, 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). However 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.

6 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 the introduction are open. Through work on examples, we are developing a methodology to *play* with with physical theories and their relations with computational ideas, in order to explore the interface between experiments and algorithms.

We have shown that the scatter machine, based on a simple Newtonian theory, can calculate any point $x \in [0, 1]$ 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 and output, 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 inevitable. Hence, the scatter machine is a hyper-computer that does not have the *deus ex machina* property of devices. The existence of the scatter machine advances theoretically the informal reflections of Geroch and Hartle [9] 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 and Tucker [1], 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 $\mathbb{N} = \{0, 1, 2, \ldots\}$ of natural numbers. The systems were 2 dimensional bagatelles with single particles that needed *unbounded* space, time and energies for their computations. In Beggs and Tucker [2] 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 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.

References

[1] E J Beggs and J V Tucker, Computations via experiments with kinematic systems, Research Report 4.04, Department of Mathematics, University of Wales Swansea, March 2004 or Technical Report 5-2004, Department of Computer Science, University of Wales Swansea, March 2004.

[2] E J Beggs and J V Tucker, Embedding infinitely parallel computation in Newtonian kinematics, *Applied Mathematics and Computation*, in press.

[3] E J Beggs and J V Tucker, Can Newtonian systems, bounded in space, time, mass and energy compute all functions?, *Theoretical Computer Science*, in press.

[4] V Bush, The differential analyser. A new machine for solving differential equations, *Journal of Franklin Institute* 212 (1931), 447-488.

[5] S B Cooper and P Odifreddi, Incomputability in nature, in S. B. Cooper and S. S. Goncharov, (eds.), *Computability and Models: Perspectives East and West*, Kluwer Academic/Plenum Publishers, Dordrecht, 2003, pp. 137-160.

[6] N C A da Costa and F A Doria, Undecidability and incompleteness in classical mechanics, *International Journal of Theoretical Physics* 30 (1991), 1041-1073.

[7] N C A da Costa and F A Doria, Classical physics and Penrose's thesis, *Foundations of Physics Letters* 4 (1991), 363-373.

[8] M Davis, The myth of hypercomputation, in C Teuscher (ed), *Alan Turing: Life and Legacy of a Great Thinker*, Springer, 2004, 195-211.

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

[10] D S Graça and J F Costa, Analog computers and recursive functions over the reals, Journal of Complexity, 19 (5) (2003) 644-664.

[11] A J G Hey, (ed) *Feynman and computation*, Westview Press, 2002.

[12] A V Holden, J V Tucker, H Zhang and M Poole, Coupled map lattices as computational systems, *American Institute of Physics - Chaos* 2 (1992), 367-376.

[13] E Griffor, (ed) *Handbook of Computability Theory*, Elsevier, 1999.

[14] G Kreisel, A notion of mechanistic theory, *Synthese* 29 (1974), 9-24.

[15] G Kreisel, Review of Pour El and Richards, *Journal of Symbolic Logic* 47 (1974), 900-902.

[16] L Lipshitz and L A Rubel, A differentially algebraic replacement theorem, *Proceedings of the American Mathematical Society* 99(2) (1987), 367 – 372.

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

[18] C Moore, Recursion theory on the reals and continuous time computation, *Theoretical Computer Science* 162 (1996), 23 – 44.

[19] M A Nielsen and I L Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.

[20] P Odifreddi, *Classical Recursion Theory*, Studies in Logic and the Foundations of mathematics, Vol 129, North-Holland, Amsterdam, 1989.

[21] M B Pour-El, Abstract computability and its relations to the general purpose analog computer, *Transactions of the American Mathematical Society* 199 (1974), 1 – 28.

[22] M B Pour-El and J. I. Richards, A computable ordinary differential equation which possesses no computable solution, *Annals of Mathematical Logic* 17 (1979), 61 – 90.

[23] M B Pour-El and J. I. Richards, The wave equation with computable initial data such that its unique solution is not computable, *Advances in Mathematics* 39 (1981), 215 – 239.

[24] M B Pour-El and J. I. Richards, Computability and noncomputability in classical analysis, *Transactions of the American Mathematical Society* 275 (1983), 539 – 560.

[25] M B Pour-El and J. I. Richards, *Computability in Analysis and Physics*, Perspectives in Mathematical Logic, Springer-Verlag, Berlin, 1989.

[26] C Shannon, Mathematical theory of the differential analyser, *Journal of Mathematics and Physics* 20 (1941), 337-354.

[27] H T Siegelmann, *Neural networks and analog computation: Beyond the Turing limit*, Birkhäuser, Boston, 1999.

[28] W D Smith, Church's thesis meets the N-body problem, in *Applied Mathematics and Computation*, to appear.

[29] V Stoltenberg-Hansen and J V Tucker, Effective algebras, in S Abramsky, D Gabbay and T Maibaum (eds.) *Handbook of Logic in Computer Science. Volume IV: Semantic Modelling*, Oxford University Press, 1995, pp.357-526.

[30] V Stoltenberg-Hansen and J V Tucker, Computable rings and fields, in E. R. Griffor (ed.), *Handbook of Computability Theory*, Elsevier, 1999, 363 – 447.

[31] V Stoltenberg-Hansen and J V Tucker, Concrete models of computation for topological algebras, *Theoretical Computer Science* 219 (1999), 347 – 378.

[32] V Stoltenberg-Hansen and J V Tucker, Computable and continuous partial homomorphisms on metric partial algebras, *Bulletin for Symbolic Logic* 9 (2003), 299 – 334.

[33] W Thompson and P G Tait, *Treatise on Natural Philosophy*, Second Edition, Part I, Cambridge UniversityPress, 1880, Appendix B', 479-508.

[34] J V Tucker and J I Zucker, Computation by while programs on topological partial algebras, *Theoretical Computer Science* 219 (1999), 379 – 421.

[35] J V Tucker and J I Zucker, Computable functions and semicomputable sets on many sorted algebras, in S. Abramsky, D. Gabbay and T Maibaum (eds.), *Handbook of Logic for Computer Science* volume V, Oxford University Press, 2000, 317 – 523.

[36] J V Tucker and J I Zucker, Abstract versus concrete computation on metric partial algebras, *ACM Transactions on Computational Logic*, 5 (4) (2004) 611-668.

[37] K Weihrauch, *Computable Analysis, An introduction*, Springer-Verlag, Heidelberg, 2000.

[38] K Weihrauch and N Zhong, Is wave propogation computable or can wave computers beat the Turing machine?, *Proceedings of London Mathematical Society* 85 (2002), 312-332.

[39] K Weihrauch and N Zhong, Is the Schrödinger propagator Turing computable?, in J. E. Blanck, V. Brattka and P. Hertling (eds.), *Computability and complexity in Analysis*, Springer Lecture Notes in Computer Science, Volume 2064, 2001.

[40] S Wolfram, *A New Kind of Science*, Wolfram Media, Champaign, 2002.

[41] A Yao, Classical physics and the Church Turing thesis, *Journal ACM* 50 (2003), 100-105.