On the Complexity of Measurement in Classical Physics

Edwin Beggs[1], José Félix Costa[2,3,⋆], Bruno Loff[2,3], and John Tucker[1]

[1] Department of Mathematics and Department of Computer Science
University of Wales Swansea, Singleton Park, Swansea, SA3 2HN
Wales, United Kingdom
E.J.Beggs@Swansea.ac.uk, J.V.Tucker@swansea.ac.uk

[2] Department of Mathematics, Instituto Superior Técnico
Universidade Técnica de Lisboa
Lisboa, Portugal
fgc@math.ist.utl.pt, bruno.loff@gmail.com

[3] Centro de Matemática e Aplicações Fundamentais do Complexo Interdisciplinar
Universidade de Lisboa
Lisbon, Portugal

Abstract. If we measure the position of a point particle, then we will come about with an interval $[a_n, b_n]$ into which the point falls. We make use of a *Gedankenexperiment* to find better and better values of a_n and b_n, by reducing their relative distance, in a succession of intervals $[a_1, b_1] \supset [a_2, b_2] \supset \ldots \supset [a_n, b_n]$ that contain the point. We then use such a point as an oracle to perform relative computation in polynomial time, by considering the succession of approximations to the point as suitable answers to the queries in an oracle Turing machine. We prove that, no matter the precision achieved in such a *Gedankenexperiment*, within the limits studied, the Turing Machine, equipped with such an oracle, will be able to compute above the classical Turing limit for the polynomial time resource, either generating the class $P/poly$ either generating the class $BPP//log*$, if we allow for an arbitrary precision in measurement or just a limited precision, respectively. We think that this result is astonishingly interesting for Classical Physics and its connection to the Theory of Computation, namely for the implications on the nature of space and the perception of space in Classical Physics. (Some proofs are provided, to give the flavor of the subject. Missing proofs can be found in a detailed long report at the address http://fgc.math.ist.utl.pt/papers/sm.pdf.)

1 Introducing a *Gedankenexperiment*

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

⋆ Corresponding author. The authors are listed in alphabetic order.

M. Agrawal et al. (Eds.): TAMC 2008, LNCS 4978, pp. 20–30, 2008.

To pursue this question, we imagine using an experiment as an oracle to a Turing machine, which on being presented with, say, x_i as its i-th query, returns y_i to the Turing machine. In this paper we will consider this idea of coupling experiments and Turing machines in detail. The first thing to note is that choosing a physical experiment to use as an oracle is a major undertaking. The experiment comes with plenty of theoretical baggage: concepts of equipment, experimental procedure, instruments, measurement, observable behaviour, etc.

In earlier work [5], an experiment was devised to measure the position of the vertex of a wedge to arbitrary accuracy, by scattering particles that obey some laws of elementary Newtonian kinematics. Let *SME* denote this *Scatter Machine Experiment*. The Newtonian theory was specified precisely and the *SME* was put under a theoretical microscope: theorems were proved that showed that *the experiment was able to compute positions that were not computable by algorithms.* Indeed, the *SME* could, in principle, measure *any* real number. Thus, [5] contains a careful attempt to answer the question above, in the positive; it does so using a methodology developed and applied in earlier studies [3,4]. To address the question, here we propose to use the *SME* as an oracle to a Turing machine and to classify the computational power of the new type of machine, which we call an *analogue-digital scatter machine*. Given the results in [5], we expect that the use of *SME* will enhance the computational power and efficiency of the Turing machine.

To accomplish this, we must establish some principles that do not depend upon the *SME*. In a Turing machine, the oracle is normally specified very abstractly by a set. Here we have to design a new machine where the oracle is replaced by a specification of some physical equipment and a procedure for operating it. The design of the new machine depends heavily upon the interface and interaction between the experiment and the Turing machine.

Following some insights provided by the work of Hava T. Siegelmann and Eduardo Sontag [6], we use non uniform complexity classes of the form $\mathcal{B}/\mathcal{F}$, where $\mathcal{B}$ is the class of computations and $\mathcal{F}$ is the advice class. Context and proofs are, however, different. Examples of interest for $\mathcal{B}$ are P and BPP; examples for $\mathcal{F}$ are *poly* and *log*. The power of the machines correspond with the choice of different $\mathcal{B}/\mathcal{F}$.

2 The Scatter Machine

Experiments with scatter machines are conducted exactly as described in [5], but, for convenience and to use them as oracles, we need to review and clarify some points. The *scatter machine experiment (SME)* is defined within the Newtonian mechanics, comprising of the following laws and assumptions: (a) point particles obey Newton's laws of motion in the two dimensional plane, (b) straight line barriers have perfectly elastic reflection of particles, i.e., kinetic energy is conserved exactly in collisions, (c) barriers are completely rigid and do not deform on impact, (d) cannons, which can be moved in position, can project

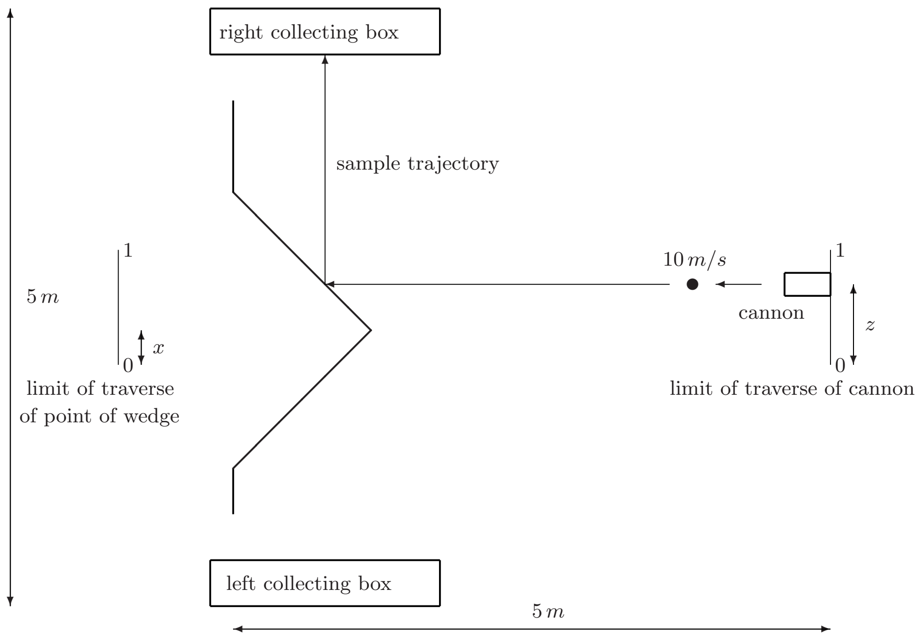

Fig. 1. A schematic drawing of the scatter machine

a particle with a given velocity in a given direction, (e) particle detectors are capable of telling if a particle has crossed a given region of the plane, and (f) a clock measures time.

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 in Figure 1.

The wedge, our motionless point particle, can be at *any* position. But we will assume it is fixed for the duration of all the experimental work. Under the control of a Turing machine, the cannon will be moved and fired repeatedly to find information about the position of the wedge. Specifically, the way the *SME* is used as an oracle in Turing machine computations, is this: a Turing machine will set a position for the canon as a query and will receive an observation about the result of firing the cannon as a response. For each input to the Turing machine, there will be finitely many runs of the experiment.

In Figure 1 the 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. Since the initial velocity is 10 m/s, the particle will enter one of the two boxes within 1 second 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° slopping 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. We make the further assumption, without loss of generality (see the report mentioned in the abstract) that the vertex of the wedge is *not* a dyadic rational.

Suppose that x is the arbitrarily chosen, but non dyadic and fixed, position of the point of the wedge. For a given cannon position z, there are two outcomes of an experiment: (a) one second after firing, the particle is in the right box — conclusion: $z > x$ —, or (b) one second after firing, the particle is in the left box — conclusion: $z < x$. The *SME* was designed to find x to arbitrary accuracy by altering z, so in our machine $0 \leq x \leq 1$ will be fixed, and we will perform observations at different values of $0 \leq z \leq 1$.

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. Errors in observation do not arise. Now consider some of the non-trivial ways in which precision depends on the positions of the cannon. There are different postulates for the precision of the cannon, and we list some in order of decreasing strength:

Definition 2.1. *The* SME *is error-free if the cannon can be set exactly to any given dyadic rational number. The* SME *is error-prone with arbitrary precision if the cannon can be set only to within a non-zero, but arbitrarily small, dyadic precision. The* SME *is error-prone with fixed precision if there is a value $\varepsilon > 0$ such that the cannon can be set only to within a given precision ε.*

The Turing machine is connected to the *SME* in the same way as it would be connected to an oracle: we replace the query state with a *shooting state* (q_s), the "yes" state with a *left state* (q_l), and the "no" state with a *right state* (q_r). The resulting computational device is called the *analog-digital scatter machine*, and we refer to the *vertex position* of an analog-digital scatter machine when mean to discuss the vertex position of the corresponding *SME*.

In order to carry out a scatter machine experiment, the analog-digital scatter machine will write a word z in the query tape and enter the shooting state. This word will either be "1", or a binary word beginning with 0. We will use z indifferently to denote both a word $z_1 \ldots z_n \in \{1\} \cup \{0s : s \in \{0,1\}^*\}$ and the corresponding dyadic rational $\sum_{i=1}^{n} 2^{-i+1} z_i \in [0,1]$. In this case, we write $|z|$ to denote n, i.e., the size of $z_1 \ldots z_n$, and say that the analog-digital scatter machine is *aiming* at z. The Turing machine computation will then be interrupted, and the *SME* will attempt to set the cannon at z. The place where the cannon is actually set at depends on whether the *SME* is error-free or error-prone. If the *SME* is error-free, the cannon will be placed exactly at z. If the *SME* is error-prone with arbitrary precision, then the cannon will be placed at some point in the interval $[z - 2^{-|z|-1}, z + 2^{-|z|-1}]$ *with a uniform probability distribution over this interval.* This means that for different words representing the same dyadic rational, the longest word will give the highest precision. If the *SME* is error-prone with fixed dyadic precision ε, then the cannon will be placed somewhere in the interval $[z - \varepsilon, z + \varepsilon]$, again with a uniform probability distribution.

After setting the cannon, the *SME* will fire a projectile particle, wait one second and then check if the particle is in either box. If the particle is in the right collecting box, then the Turing machine computation will be resumed in the state q_r. If the particle is in left box, then the Turing machine computation will be resumed in the state q_l. With this behaviour, we obtain three distinct analog-digital scatter machines.

Definition 2.2. *An* error-free analog-digital scatter machine *is a Turing machine connected to an error-free* SME*. In a similar way, we define an* error-prone analog-digital scatter machine with arbitrary precision*, and an* error-prone analog-digital scatter machine with fixed precision*.*

The error-free analog-digital scatter machine has a very simple behaviour. If such a machine, with vertex position $x \in [0, 1]$, aims at a dyadic rational $z \in [0, 1]$, we are certain that the computation will be resumed in the state q_l if $z < x$, and that it will be resumed in the state q_r when $z > x$. We define the following decision criterion.

Definition 2.3. *Let $A \subseteq \Sigma^*$ be a set of words over Σ. We say that an error-free analog-digital scatter machine $\mathcal{M}$* decides *A if, for every input $w \in \Sigma^*$, w is accepted if $w \in A$ and rejected when $w \notin A$. We say that $\mathcal{M}$* decides *A* in polynomial time*, if $\mathcal{M}$ decides A, and there is a polynomial p such that, for every $w \in \Sigma^*$, the number of steps of the computation is bounded by $p(|w|)$.*

The error-prone analog-digital scatter machines, however, do not behave in a deterministic way. If such a machine aims the cannon close enough to the vertex position, and with a large enough error, there will be a positive probability for both the particle going left or right. If the vertex position is x, and the error-prone analog-digital scatter machine $\mathcal{M}$ aims the cannon at z, then the probability of the particle going to the left box, denoted by $\mathbb{P}(\mathcal{M} \leftarrow \mathit{left})$, is given by:

$$\mathbb{P}(\mathcal{M} \leftarrow \mathit{left}) = \begin{cases} 1 & \text{if } z < x - \varepsilon \\ \frac{1}{2} + \frac{x-z}{2\varepsilon} & \text{if } x - \varepsilon \leq z \leq \tilde{x} + \varepsilon \\ 0 & \text{if } z > \tilde{x} + \varepsilon \end{cases}$$

The value ε will be either $2^{-|z|-1}$ or a fixed value, depending on the type of error-prone analog-digital scatter machine under consideration. The probability of the particle going to the right box is $\mathbb{P}(\mathcal{M} \leftarrow \mathit{right}) = 1 - \mathbb{P}(\mathcal{M} \leftarrow \mathit{left})$. We can thus see that a deterministic decision criteria is not suitable for these machines. For a set $A \subseteq \Sigma^*$, an error-prone analog-digital scatter machine $\mathcal{M}$, and an input $w \in \Sigma^*$, let the *error probability* of $\mathcal{M}$ for input w be either the probability of $\mathcal{M}$ rejecting w, if $w \in A$, or the probability of $\mathcal{M}$ accepting w, if $w \notin A$.

Definition 2.4. *Let $A \subseteq \Sigma^*$ be a set of words over Σ. We say that an error-prone analog-digital scatter machine $\mathcal{M}$* decides *A if there is a number $\gamma < \frac{1}{2}$, such that the error probability of $\mathcal{M}$ for any input w is smaller than γ. We say* *that $\mathcal{M}$* decides A in polynomial time*, if $\mathcal{M}$ decides A, and there is a polynomial p such that, for every input $w \in \Sigma^*$, the number of steps in every correct computation is bounded by $p(|w|)$.*

3 The Relevant Computational Classes

We will see, in the sections that follow, that the non-uniform complexity classes give the most adequate characterisation of the computational power of the analog-digital scatter machine. Non-uniform complexity classifies problems by studying families of finite machines (e.g., circuits) $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$, where each $\mathcal{C}_n$ decides the restriction of some problem to inputs of size n. It is called *non-uniform*, because for every $n \neq m$ the finite machines $\mathcal{C}_n$ and $\mathcal{C}_m$ can be entirely unrelated, while in uniform complexity the algorithm is the same for inputs of every size. A way to connect the two approaches is by means of *advice classes*: one assumes that there is a unique algorithm for inputs of every size, which is aided by certain information, called *advice*, which may vary for inputs of different sizes. The advice is given, for each input w, by means of function $f : \mathbb{N} \to \Sigma^*$, where Σ^* is the input alphabet.

Definition 3.1. *Let $\mathcal{B}$ be a class of sets and $\mathcal{F}$ a class of functions. The advice class $\mathcal{B}/\mathcal{F}$ is the class of sets A for which some $B \in \mathcal{B}$ and some $f \in \mathcal{F}$ are such that, for every w, $w \in A$ if and only if $\langle w, f(|w|)\rangle \in B$.*

$\mathcal{F}$ is called the *advice class* and f is called the *advice function*. Examples for $\mathcal{B}$ are the well known classes P, or BPP (see [1, Chapter 6]). We will be considering two instances for the class $\mathcal{F}$: *poly* is the class of functions with polynomial size values, i.e., *poly* is the class of functions $f : \mathbb{N} \to \Sigma^*$ such that, for some polynomial p, $|f(n)| \in O(p(n))$; log is the class of functions $g : \mathbb{N} \to \Sigma^*$ such that $|g(n)| \in O(\log(n))$. We will also need to treat prefix non-uniform complexity classes. For these classes we may only use prefix functions, i.e., functions f such that $f(n)$ is always a prefix of $f(n+1)$. The idea behind prefix non-uniform complexity classes is that the advice given for inputs of size n may also be used to decide smaller inputs.

Definition 3.2. *Let $\mathcal{B}$ be a class of sets and $\mathcal{F}$ a class of functions. The prefix advice class $\mathcal{B}/\mathcal{F}*$ is the class of sets A for which some $B \in \mathcal{B}$ and some prefix function $f \in \mathcal{F}$ are such that, for every length n and input w, with $|w| \leq n$, $w \in A$ if and only if $\langle w, f(n)\rangle \in B$.*

For the non-deterministic classes we need a refined definition:

Definition 3.3. *$BPP//poly$ is the class of sets A for which a probabilistic polynomial Turing machine $\mathcal{M}$, a function $f \in poly$, and a constant $\gamma < \frac{1}{2}$ exist such that $\mathcal{M}$ rejects $\langle w, f(|w|)\rangle$ with probability at most γ if $w \in A$ and accepts $\langle w, f(|w|)\rangle$ with probability at most γ if $w \notin A$. A similar definition applies to the class $BPP//\log*$.*

It can be shown that $BPP//poly = BPP/poly$, but it is unknown whether $BPP/\log* \subseteq BPP//\log*$. It is important to notice that the usual non-uniform complexity classes contain undecidable sets, e.g., $P/poly$ contains the *halting set*.

4 Infinite Precision

FIRST GEDANKENEXPERIMENT: the cannon can be placed at some dyadic rational with infinite precision. I.e., we began by investigating the error-free machine, which gives the simplest situation.

For every $n \in \mathbb{N}$ and $x \in \mathbb{R}$, let $x\lceil_n$ be the rational number obtained by truncating x after the first n digits in its binary expansion. Then we showed the following result:

Proposition 4.1. *Let $\mathcal{M}$ be an error-free analog-digital scatter machine with the vertex placed at the position x. Let $\tilde{\mathcal{M}}$ be the same machine, but with the vertex placed at the position $\tilde{x} = x\lceil_t$. Then, for any input w, $\mathcal{M}$ and $\tilde{\mathcal{M}}$, after t steps of computation, make the same decisions.*

We may now sketch the proof of the following main theorem of this section.

Theorem 4.1. *The class of sets decided by error-free analog-digital scatter machines in polynomial time is exactly $P/poly$.*

The sketch of the proof is done by the way of polynomial advice. Let A be a set in $P/poly$, and, by definition, let $B \in P$, $f \in poly$ be such that $w \in A \iff \langle w, f(|w|)\rangle \in B$. Let $\tilde{f} : \mathbb{N} \to \Sigma^*$ be a function, also in *poly*, such that, if the symbols of $f(n)$ are $\xi_1\xi_2\ldots\xi_{p(n)}$, then $\tilde{f}(n) = 0\xi_1 0\xi_2 \ldots 0\xi_{p(n)}$ We can create an error-free analog-digital scatter machine which also decides A, setting the vertex at the position $x = 0.\tilde{f}(1)11\tilde{f}(2)11\tilde{f}(3)11\ldots$ Given any input w of size n, the error-free analog-digital scatter machine $\mathcal{S}_x$ can use the bisection method to obtain $f(n)$ in polynomial time. Then the machine uses the polynomial-time algorithm which decides B, and accepts if and only if $\langle w, f(n)\rangle$ is in B. Thus we have shown that an error-free analog-digital scatter machine can decide any set in $P/poly$ in polynomial time.

As for the converse, let C be any set decided in polynomial time by an error-free analog-digital scatter machine with the vertex at the position x. Proposition 4.1 ensures that to decide on any input w, the machine only makes use of $p(|w|)$ digits of x, where p is a polynomial. Thus we can see that the set must be in $P/poly$, using the advice function $g \in poly$, given by $g(n) = x\lceil_{p(n)}$.

We then conclude that *measuring the position of a motionless point particle in Classical Physics, using a infinite precision cannon,* [1] *in polynomial time, we are deciding a set in P/poly. Note that, the class P/poly includes the* Halting Set. But, most probably, if we remove the *infinite precision criterion for the cannon*, we will loose accuracy of observations, and we will loose the computational power of our physical oracle...

[1] Remember that by infinite precision we mean the cannon to be settle at a rational point with infinite precision.

5 From Infinite to Unlimited Precision

SECOND GEDANKENEXPERIMENT: the cannon can be placed at some dyadic rational z up to some dyadic arbitrary precision ε, let us say $\varepsilon = 2^{-|z|-1}$. I.e., we will be investigating the error-prone machine, which gives the next simplest situation. We will conclude that the computational power of these machines is not altered by considering such a small error, since they decide exactly $BPP//poly = BPP/poly = P/poly$ in polynomial time. As in Section 4, we showed that only a number of digits of the vertex position linear in the length of computation influences the decision of such an error-prone analog-digital scatter machine. Notice that the behaviour of the error-prone analog-digital scatter machines is probabilistic, because if the machine shoots close enough to the motionless point particle position — the vertex —, and with a large enough error ε, the projectile can go both left or right with a non-zero probability. Thus, we can not ensure that when we truncate the vertex position to $O(t(n))$ digits, the state of the machine will be exactly the same after $t(n)$ steps. Instead we showed that if a machine decides in time $t(n)$, then by truncating the vertex position to $O(t(n))$ digits, the machine will decide the same set for inputs up to size n. In the following statement, note that if a set is decided by an error-prone analog-digital scatter machine with error probability bounded by γ, we may assume without loss of generality that $\gamma < \frac{1}{4}$.

Proposition 5.1. *Let $\mathcal{M}$ be an error-prone analog-digital scatter machine with arbitrary precision, with the vertex at x, deciding some set in time $t(n)$ with error probability bounded by $\gamma < \frac{1}{4}$. Let $\tilde{\mathcal{M}}$ be an error-prone machine with arbitrary precision, with the same finite control as $\mathcal{M}$ and with the vertex placed at $\tilde{x} = x\restriction_{5t(n)}$. Then $\mathcal{M}$ and $\tilde{\mathcal{M}}$ make the same decision on every input of size smaller or equal to n.*

This allows us to show the following.

Proposition 5.2. *Every set decided by an error-prone analog-digital scatter machine with arbitrary precision in polynomial time is in $BPP//poly$.*

Let A be a set decided by a precise error-free analog-digital scatter machine $\mathcal{M}$ in polynomial time p, and with a error probability bounded by $\frac{1}{4}$. Let x be the position of the vertex of $\mathcal{M}$. We use the advice function $f \in poly$, given by $f(n) = x\restriction_{5p(n)}$, to construct a probabilistic Turing machine $\tilde{\mathcal{M}}$ which decides A in polynomial time.

Given any dyadic rational $\tilde{x} \in [0, 1]$, the machine $\tilde{\mathcal{M}}$ can carry out a Bernoulli trial X with an associated probability $\mathbb{P}(X = 1) = \tilde{x}$. If $\tilde{x}$ has the binary expansion $\xi_1 \ldots \xi_k$, the machine $\tilde{\mathcal{M}}$ tosses its balanced coin k times, and constructs a word $\tau_1 \ldots \tau_k$, where τ_i is 1 if the coin turns up heads and 0 otherwise. The Bernoulli trial will have the outcome 1 if $\xi_1 \ldots \xi_k < \tau_1 \ldots \tau_k$, and 0 otherwise, and this will give the desired probability.

The probabilistic machine $\tilde{\mathcal{M}}$ will decide if $w \in A$ by simulating $\mathcal{M}$ on the input w with the vertex placed at the position $\tilde{x} = x \restriction 5p(n)$. In order to mimic the shooting of the cannon from the position z, which should have an error $\varepsilon = 2^{-|z|-1}$, the machine will carry out a Bernoulli trial X with an associated dyadic probability.

Then $\tilde{\mathcal{M}}$ will simulate a left hit when $X = 1$ and a right hit when $X = 0$. As we have seen in the previous Proposition 5.1, $\mathcal{M}$ will, when simulated in this way, decide the same set in polynomial time and with bounded error probability.

In a way similar to the first part of the proof of Theorem 4.1 we can prove the statement:

Proposition 5.3. *An error-prone analog-digital scatter machine with arbitrary precision can obtain n digits of the vertex position in $O(n^2)$ steps.*

The conclusion for this section comes with the following theorem.

Theorem 5.1. *The class of sets decided by error-prone analog-digital scatter machines with arbitrary precision in polynomial time is exactly $BPP//poly = P/poly$.*

It seems that measurement in more reasonable conditions do not affect the computational power of a motionless point particle position taken as oracle. I.e., making measurements in Classical Physics with incremental precision decide languages above the Turing limit, namely the *halting set*. But, surely, the computational power of a fuzzy motionless point particle position, i.e., a point to which we have access only with a finite *a priori* precision in measurement, will drop above the Turing limit...

6 *A Priori* Finite Precision

THIRD GEDANKENEXPERIMENT: the cannon can be placed at some dyadic rational z up to some dyadic fixed precision ε. We will show that such machines may, in polynomial time, make probabilistic guesses of up to a logarithmic number of digits of the position of the vertex. We will then conclude that these machines decide exactly $BPP// \log *$.

Proposition 6.1. *For any real value $\delta < \frac{1}{2}$, prefix function $f \in \log$, there is an error-prone analog-digital scatter machine with fixed precision which obtains $f(n)$ in polynomial time with an error of at most δ.*

The proof will take two steps. First we show that if f is a prefix function in log, then there is a real value $0 \leq r \leq 1$ such that it is possible to obtain the value $f(n)$ from a logarithmic number of digits of r. Then we will show that by carefully choosing the vertex position, we can guess a logarithmic number of digits of r with an error rate δ. If $f(n)$ is ultimately constant, then the result is trivial, and so we assume it is not so. After the work of [2], we can assume, without loss of generality, that there exist $a, b \in \mathbb{N}$ such that $|f(n)| = \lfloor a \log n + b \rfloor$.

Since f is a prefix function, we can consider the infinite sequence φ which is the limit of $f(n)$ as $n \to \infty$. Let φ_n be the n-th symbol (1 or 0) in this sequence, and set $r = \sum_{n=1}^{\infty} \varphi_n 2^{-n}$. Then, since f is not ultimately constant, the digits in the binary expansion of r are exactly the symbols of φ. Then the value $f(n)$ can be obtained from the first $\lfloor a \log n + b \rfloor$ digits of r.

The real number r is a value strictly between 0 and 1. Now suppose that ε is the error when positioning the cannon. We then set the vertex of our analog-digital scatter machine at the position $x = \frac{1}{2} - \varepsilon + 2r\varepsilon$. Our method for guessing digits of r begins by commanding the cannon to shoot from the point $\frac{1}{2}$ a number z of times. If the scatter machine experiment is carried out by entering the shooting state after having written the word 01 in the query tape, the cannon will be placed at some point in the interval $[\frac{1}{2} - \varepsilon, \frac{1}{2} + \varepsilon]$, with a uniform distribution. Then we conclude that the particle will go left with a probability of r and go right with a probability of $1 - r$.

After shooting the cannon z times in this way, we count the number of times that the particle went left, which we denote by L. The value $\tilde{r} = \frac{L}{z}$ will be our estimation of r. In order for our estimation to be correct in the sufficient number of digits, it is required that $|r - \tilde{r}| \leq 2^{-|f(n)|-1}$. By shooting the cannon z times, we have made z Bernoulli trials. Thus L is a random variable with expected value $\mu = zr$ and variance $\nu = zr(1-r)$. By Chebyshev's inequality, we conclude that, for every Δ,

$$\mathbb{P}(|L - \mu| > \Delta) = \mathbb{P}(|z\tilde{r} - zr| > \Delta) = \mathbb{P}\left(|\tilde{r} - r| > \frac{\Delta}{z}\right) \leq \frac{\nu}{\Delta^2}.$$

Choosing $\Delta = z2^{-|f(n)|-1}$, we get

$$\mathbb{P}(|\tilde{r} - r| > 2^{-|f(n)|-1}) \leq \frac{r(1-r)2^{2|f(n)|+2}}{z}$$

And so, the probability of making a mistake can be bounded to δ by making $z > \delta^{-1} r(1-r) 2^{2a \log n + 2b + 2} \in O(n^{2a})$ experiments.

The proposition above will guarantee us that for every fixed error ε we can find a vertex position that will allow for an *SME* to extract information from this vertex position. It does not state that we can make use of any vertex position independently of the fixed error ε. It can be shown, however, that if ε is a dyadic rational, then we may guess $O(\log n)$ digits of the vertex position in polynomial time.

Proposition 6.2. *The class of sets decided in polynomial time by error-prone analog-digital scatter machines with* a priori *fixed precision is exactly* $BPP//\log *$.

Since the class $BPP//\log *$ include non-recursive sets, measurements in this more realistic condition still decide super-Turing languages.

We can even think about changing the (uniform) probability distribution in the last two sections, making experiments closer and closer to reality...

7 Conclusion

We have seen that every variant of the analogue-digital scatter machine has a hypercomputational power. For instance, if $K = \{0^n :$ the Turing machine coded by n halts on input $0\}$, then K can be decided, in polynomial time, either by an error-free analog-digital scatter machine, or by an error-prone analog-digital scatter machine with arbitrary precision. It is obvious that the hypercomputational power of the analog-digital scatter machine arises from the precise nature of the vertex position. If we demand that the vertex position is a computable real number, then the analog-digital scatter machine can compute no more than the Turing machine, although it can compute, in polynomial time, faster than the Turing machine, namely $REC \bigcap P/poly$, the recursive part of $P/poly$.

In order to use the scatter machine experiment as an oracle, we need to assume that the wedge is sharp to the point and that the vertex is placed on a precise value x. Without these assumptions, the scatter machine becomes useless, since its computational properties arise exclusively from the value of x. The existence of an arbitrarily sharp wedge seems to contradict atomic theory, and for this reason the scatter machine is not a valid counterexample to the physical Church–Turing thesis. If this is the case, then what is the relevance of the analog-digital scatter machine as a model of computation? The scatter machine is relevant when it is seen as a *Gedankenexperiment*. In our discussion, we could have replaced the barriers, particles, cannons and particle detectors with any other physical system with this behaviour. So the scatter machine becomes a tool to answer the more general question: *if we have a physical system to measure an answer to the predicate $y \leq x$, to what extent can we use this system in feasible computations?*

As an open problem, besides a few other aspects of the measurement apparatus that we didn't cover up in this paper, we will study a point mass in motion, according to some physical law, like a Newtonian gravitation field, and we will apply instrumentation to measure its position and velocity.

References

1. Balcázar, J.L., Días, J., Gabarró, J.: Structural Complexity I. Springer, Heidelberg (1988)
2. Balcázar, J.L., Hermo, M.: The structure of logarithmic advice complexity classes. Theoretical Computer Science 207(1), 217–244 (1998)
3. Beggs, E., Tucker, J.: Embedding infinitely parallel computation in newtonian kinematics. Applied Mathematics and Computation 178(1), 25–43 (2006)
4. Beggs, E., Tucker, J.: Can Newtonian systems, bounded in space, time, mass and energy compute all functions? Theoretical Computer Science 371(1), 4–19 (2007)
5. Beggs, E., Tucker, J.: Experimental computation of real numbers by Newtonian machines. Proceedings of the Royal Society 463(2082), 1541–1561 (2007)
6. Siegelmann, H.: Neural Networks and Analog Computation: Beyond the Turing Limit. Birkhäuser (1999)