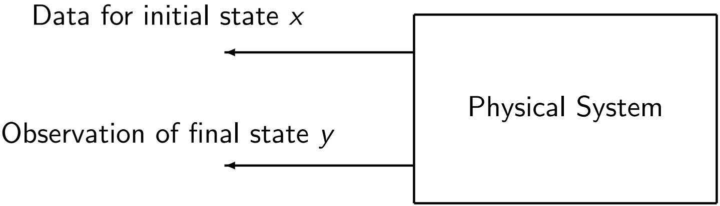
# Computation, Measurement and the Interface Between Physical Systems and Algorithms

Edwin Beggs and John V. Tucker, Swansea University

José Félix Costa, Universidade de Lisboa

# Computation by physical systems




1. Input data $x$ used to make initial conditions for system.
2. System evolves for a finite or infinite time.
3. Output data $y$ obtained by measuring some observable for the system.
4. Physical system computes a function $y = f(x)$.

## Some basic questions

What are the functions computable by experiments with physical systems?

How do they compare with the functions computable by algorithms?

Given a technology for building physical systems, does it define less or more functions than algorithms?

Does there exist a physical system that exhibits non-algorithmically computable behaviour?

How dependent are the answers on the physical theories being used?

Even if we cannot build a hypercomputer, might we just find one on some planet?

## Wave equations. Pour-El and Richards 1981

**Theorem**

*The linear wave equation in 3 space dimensions*

$$\frac{1}{c^2}\,\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}$$

*can show non-computable behaviour, i.e. there are computable initial conditions ($t = 0$) for which the wave at some future time ($t = 1$) is not computable.*

Can we use this as a practical method of hypercomputation? Weihrauch & Zhong (2002) showed that if you remove a hidden discontinuity in this problem, then solutions are computable.

# The beam balance scale

Figure: Balance scale.

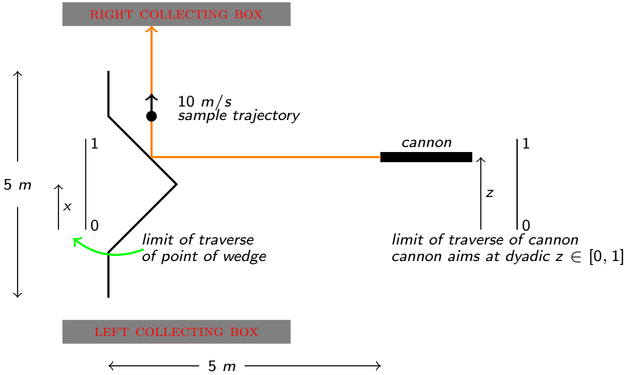
## The beam balance scale

Let the unknown mass be in the right pan:

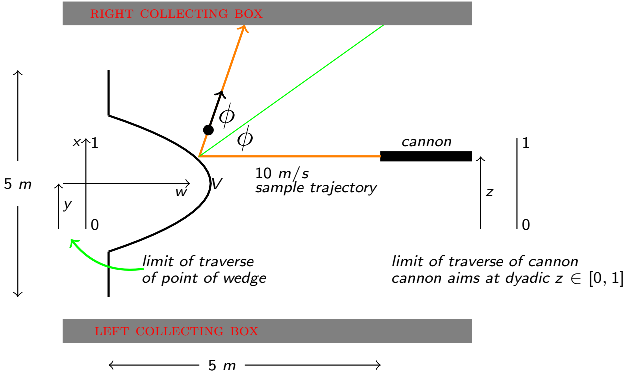
1. If the pan of standards (left) goes up, then we have to replace a weight by one of its multiples or to add one of its submultiples.
2. If the pan of standards goes down, then we have to replace one of the weights in the left pan by one of its submultiples.

The measurement of mass applies the binary search method.

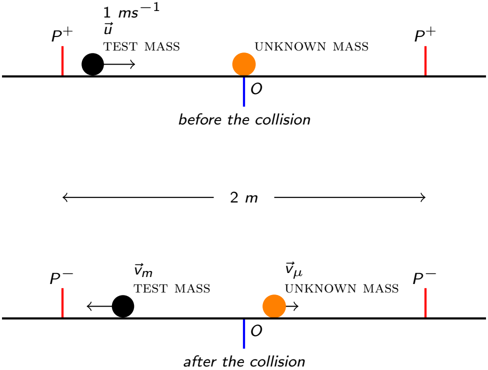
# Scatter machine experiment I: Beggs and Tucker (2007)



# Scatter machine experiment II: Beggs, Costa and Tucker (2012)



# Collider experiment: Beggs, Costa and Tucker (2010)



## Some reflections

### 1. Physical Quantities

Physical quantities are measured by real numbers e.g., using intervals $[0, 1]$. A real number stands for a process of approximation by rational numbers. The field of rationals is the data type of measurement.

### 2. Experimental procedure

The procedures used to measure are systematic and a form of algorithm procedure. The procedures may involve error bounds and take time. An experiment is designed using a theory.

## 3. Measurable numbers

What numbers arise in physical experiments? What is the relationship between measurable numbers and computable numbers?

## 4. Anthropomorphic foundation

Computation is a mathematical process performed by people.
Measurement is a physical process performed by people.

## Six Principles

**Principle 1. Physical subtheories**

Define a physical sub-theory $T$ of a physical theory and examine computation by systems that are valid models of $T$. Define $T$-computablity.

**Principle 2. Computers and subtheories**

Find physical systems that are models of $T$ that can implement automata or computers. Seek ways of embedding sets and functions, models of computers and hypercomputers into $T$-systems.

**Principle 3. Border between computer and hypercomputer**

Seek necessary and sufficient conditions for $T$-systems to implement classes of functions, algorithmically computable or not.

**Principle 4. Reviewing and refining theories**

Determine the scope of the subtheory $T$ by strengthening or weakening the assumptions of $T$. Collect a portfolio of related theories.

**Principle 5. Combining experiments and algorithms**

Use the physical system, specified by subtheory $T$, as an oracle for a class of algorithms and determine how it extends computational power or efficiency.

**Principle 6. Algorithms controlling experiments**

Use algorithms to control the physical system, specified by subtheory $T$. Take the experimental procedure to be an algorithm and the equipment to be an oracle. Determine how this limits or enhances the accuracy or efficiency of measurement of the system.

# The Physical Oracle

## Turing on Oracles

1. "Let us suppose we are supplied with some unspecified means of solving number-theoretic problems; a kind of oracle as it were. We shall not go further into the nature of this oracle apart from saying that it cannot be a machine."
2. The halting problem for o-machines is undecidable by o-machines.
3. A M Turing, Systems of logic based on ordinals, *Proceedings London Mathematical Society*, Series 2, 45 (1939) 161-228.

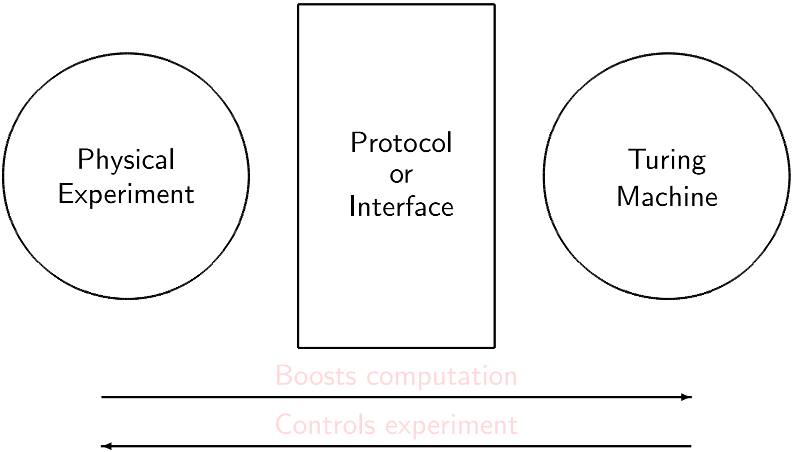
## Post on Orcales

1. Any set $S$ can be used as an oracle in an algorithm: from time to time, in the course of a computation, an algorithm produces a datum $x$, asks "Is $x \in S$?".
2. Algorithm receives an *exact* answer, "yes" or "no", in a *single step* of the computation.
3. Use: To boost the power of algorithms. Formulate reduction of problems and degrees. Classification of undecidable sets degrees.
4. E L Post, Recursively enumerable sets of positive integers and their decision problems, *Bulletin AMS* 50 (1944) 284-316.

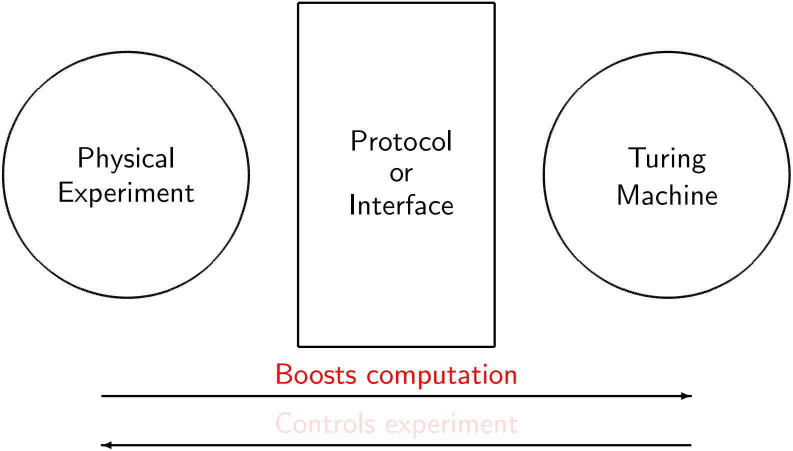
## Physical systems/devices/experiments as Oracles

1. Any physical system $S$ can be used as an oracle in an algorithm: from time to time, in the course of a computation, an algorithm produces a datum $x$, and uses this to initialise parameters of physical equipment that will result in a physical process.
2. The algorithm receives an *exact* or *approximate* value from the physical process, which will *take time*.
3. Beggs, Costa, Loff and Tucker, Computational complexity with experiments with oracles, *Proceedings Royal Society Series* A, 464 (2008) 2777-2801.

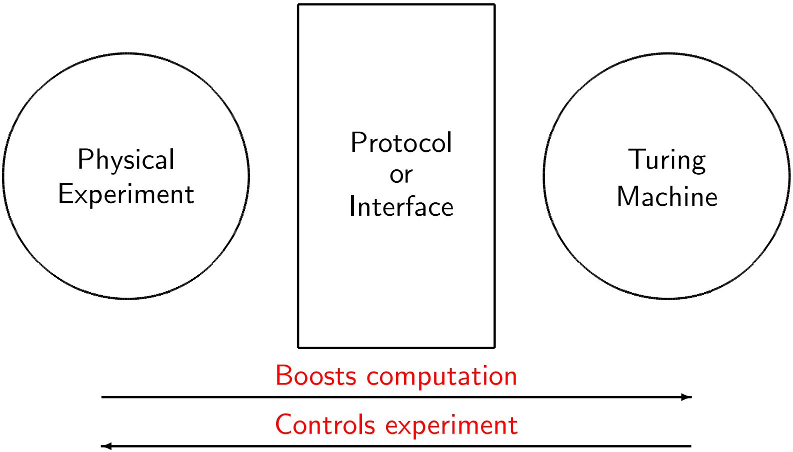
# The architecture



# The architecture



# The architecture



## Physical experiments

### A coin as physical oracle

1. Computational complexity theorists use one physical experiment without "being conscious" of it: a coin toss.
2. According to some accepting/rejecting criteria, they measure the power of Turing machines in polynomial time by defining several classes such as *PP*, *BPP*, *ZPP*, *NP*, etc.
3. Tossing a coin to make a guess is like using a physical experiment as resource!

## Physical experiments

We have considered experiments in Physics with the purpose of measuring a physical quantity:

1. Distance;
2. Geographical coordinates;
3. Inertial mass;
4. Electrical resistance;
5. Mass by the scattering of particles in a Coulombian field;
6. Brewster angle in Optics.

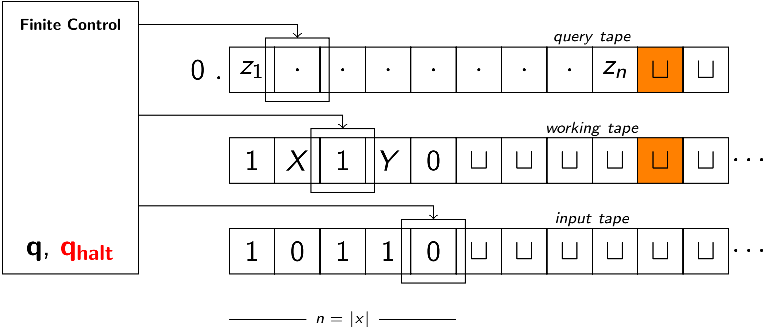
# The Concept of a Scientist

## Role of the Turing machine

An experimental procedure is a precise and detailed description of how to operate equipment to perform an experiment. The procedure can be followed by a technician independently.
The experimental procedure can be considered to be an algorithm, which can be coded as a Turing machine.

Just as Turing modelled a human calculator, we model – more generally! – a human experimenter ...
Many experiments are governed by software.

## Take a Turing machine



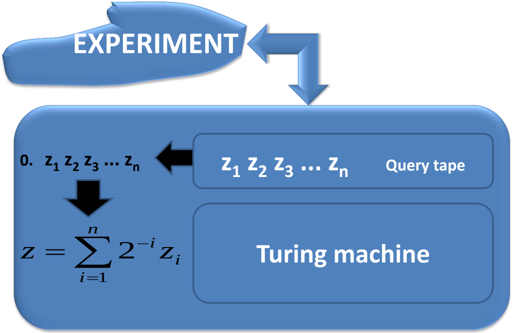



Figure: Take queries to be dyadic rationals

## Query tape

After writing a binary string $z$ in the query tape, the machine may enter in the query state, the experimental equipment is presented with $z$ as a dyadic rational. After some time the answer is returned and the machine enters in state

$$q_{yes} \text{ or } q_{no}.$$

## Timing

**Definition**

An protocol for the analogue-digital interface is $f$-bounded if the total time taken by the analogue-digital machine to process a query string, of size $n$, and return an answer, is bounded by $f(n)$ steps of the machine's clock.

# Schedule

**The sequence $T(1)$, ..., $T(n)$, ... constitute a schedule**

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Position of the digit* | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ⋯ |
| *Binary digit* | 0 . | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | ⋯ |
| *Duration of oracle call* | | 3 | 8 | 15 | 24 | 35 | 48 | 63 | 80 | 99 | 120 | ⋯ |
| *Time* | | T(1) | T(2) | T(3) | T(4) | T(5) | T(6) | T(7) | T(8) | T(9) | T(10) | ⋯ |

## Enriching the digital-analog physical experiment

Moreover, $T$ should constitute a "true" clock, i.e., it should be time constructible.

**Definition**

A function $T : \mathbb{N} \to \mathbb{N}$ is said to be time constructible if there exists a Turing machine that, for all inputs of size $n$, halts exactly in $T(n)$ steps.

# Non-uniform Complexity Classes

## The classes $\mathcal{B}/\mathcal{F}$

**Definition**

Let $\mathcal{B}$ be a class of sets and $\mathcal{F}$ a class of total functions. The non-uniform 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 $w$,

$$w \in A \text{ if, and only if, } \langle w, f(|w|)\rangle \in B.$$

## The classes $\mathcal{B}/\mathcal{F}^{\star}$

**Definition**

Let $\mathcal{B}$ be a class of sets and $\mathcal{F}$ a class of functions. The prefix non-uniform 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 $|w| \leq n$,

$$w \in A \text{ if, and only if, } \langle w, f(n) \rangle \in B.$$

## BPP//poly

**Definition**

*BPP*//*poly* is the class of sets $A$ for which a probabilistic 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 $\gamma$ if $w \in A$ and accepts $\langle w, f(|w|)\rangle$ with probability at most $\gamma$ if $w \notin A$.

# BPP//log*

## Definition

$BPP// \log^\star$ is the class of sets $A$ for which a probabilistic Turing machine $\mathcal{M}$, a prefix function $f \in \log$, and a constant $\gamma < \frac{1}{2}$ exist such that, for every length $n$ and input $w$ with $|w| \leq n$, $\mathcal{M}$ rejects $\langle w, f(n) \rangle$ with probability at most $\gamma$ if $w \in A$ and accepts $\langle w, f(n) \rangle$ with probability at most $\gamma$ if $w \notin A$.

## Structural complexity

**Definition**

The sparse halting set is

$$\mathrm{HALT} = \{0^n : n \text{ codes for a TM that halts on input } 0\}$$

**Proposition**

*The halting problem, coded by the sparse halting set, is in $P/poly$ and $P/\log^\star$.*

## Structural complexity

**Proposition**

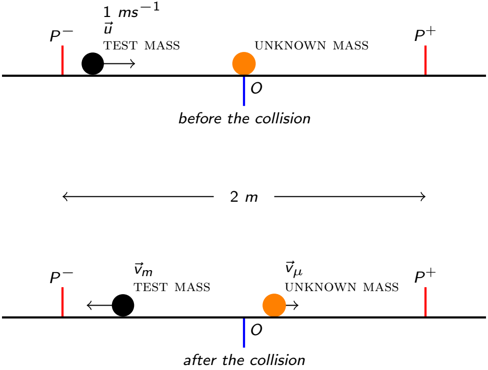
*If SAT* $\in P/\log$*, then* $P = NP$.

**Proposition**

*There exits sets in EXPSPACE not in* $P/poly$.

# Measuring Mass

# Collider experiment



# Collider experiment

## Observations

1. Test particle $m$ is detected backward, in time $t$:
   $m < \mu$ in time $t$;
2. Test particle $m$ is detected forward, in time $t$:
   $\mu < m$ in time $t$;
3. Test particle $m$ not seen within time $t$:
   $m = \mu$ up to time $t$.

# To read the first $n$ bits of an unknown mass $\mu$

1. Input the required precision $n$;
2. $m_1 = m = 0$, $m_2 := 1$ { $\mu \in (0, 1)$ };
3. While $|m| \leq n$ do begin { *loop 1* }
    1. Loop { *loop 2* }
        1. $m := (m_1 + m_2)/2$;
        2. Place the unknown particle of mass $\mu \in (0, 1)$ at origin;
        3. Fire the test particle of mass $m$ to collide;
        4. If test particle crosses $P^-$ in time $T(|m|)$, then begin $m_1 := m$;append 1
        5. If test particle crosses $P^+$ in time $T(|m|)$, then begin $m_2 := m$;append 0
        6. If no test particle crosses the flags in time $T(|m|)$, then begin return "time out"
    2. End Loop { *loop 2* }
4. End While; { *loop 1* }
5. Output the dyadic rational denoted by $m$.

# Experimentally undecidable

## Proposition

*That the test mass $m$ coincides with the given unknown mass $\mu$ cannot be established experimentally in finite time by the experimental apparatus under any experimental procedure.*

## Proposition

*To know if the unknown mass $\mu$ is a dyadic rational cannot be established experimentally in finite time by the experimental apparatus under any experimental procedure.*

The initial velocity for the test particle (mass $m$) is approximately 1m/s. The unknown mass $\mu$ is observed to cross the flags $P_{\pm}$ within the time limit, or not (out of time). The time taken to reach the flags is (for some constants $A$ and $B$)

$$\frac{A}{|m-\mu|} \leq t_{exp} \leq \frac{B}{|m-\mu|} \ . \tag{1}$$

## Proposition

*To measure the unknown mass with accuracy $|m-\mu| \leq 2^{-n}$, substituting into inequality (1), we need exponential time:*

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

*The $t_{exp} \to \infty$ as the test particle's mass $m \to \mu$. If $m = \mu$ we will wait forever for the result.*

## Three possible scenarios for the test mass:

1. Infinite precision: The test particle has a dyadic rational mass $m$.
2. Unbounded finite precision: The test particle has a dyadic rational mass m with unbounded but finite precision, let us say $m \pm 2^{-|m|-1}$.
3. Fixed finite precision: The test particle has mass $m \pm \varepsilon$, where $\varepsilon$ is fixed a priori.

## We describe two more scenarios, for the unknown mass:

1. The unknown mass can be set exactly at the real value $\mu$ — infinite precision.
2. The unknown mass can be set at the real value $\mu$, but only with unbounded but finite precision.

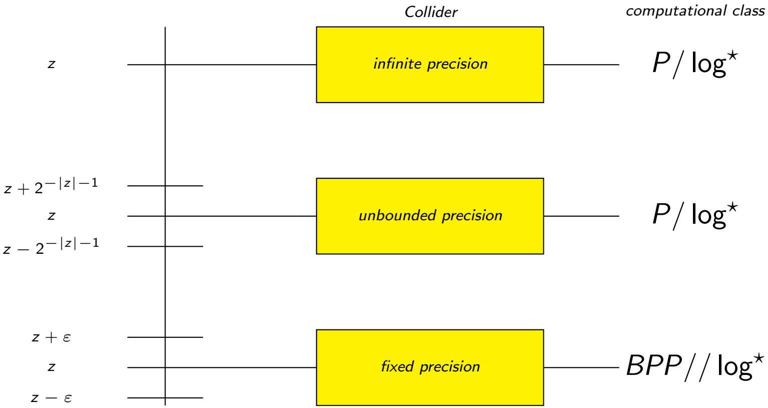
## Complexity of two classes of experiments

**Power**

1. What is the computational power of Turing machines boosted by the CME as oracle?
2. What is the computational power of such machines in polynomial time?

E J Beggs, J F Costa and J V Tucker, Limits to measurement in experiments governed by algorithms, *Mathematical Structures in Computer Science*, 20 (2010) 1019-1050.

# Precision and Computational Power



## Readable Information Content for Scatter Experiments

Table: Reduction of Readable Information Content

| Vertex | Protocol | Precision | Lower Bound | Upper Bound |
|---|---|---|---|---|
| $C^0$ not $C^1$ | const. time or poly. time | infinite | $P/poly$ | $P/poly$ |
| | | unbounded | $P/poly$ | $P/poly$ |
| | | fixed | $BPP//\log^\star$ | $BPP//\log^\star$ |
| $C^n$, $n \geq 2$ | exp. time | infinite | $P/\log^\star$ | $P/\log^\star$ |
| | | unbounded | $P/\log^\star$ | $P/\log^\star$ |
| | | fixed | $BPP//\log^\star$ | $BPP//\log^\star$ |

# Axiomatic Specification of the Interface

## A template for the axioms of a physical oracle

1. *Real values:* A real number physical parameter is being measured — position, inertial mass, etc.;
2. *Queries:* Each query is a rational test value, which is compared to the actual 'physical' value in an experiment, giving a decision procedure, such as $<$ or $>$.
3. *Finite output:* Only qualitative output is allowed in a single experiment: *right* or *left*, *yes* or *no*, *north* or *south*, etc.; or a trivalent output, adding the response *time out*.

## A template for the axioms of a physical oracle II

1. *Protocol timer:* The protocol assigns a time for the experiment, if this is exceeded *time out* is returned.
2. *Sufficiency of the protocol:* Enough experiments are completed within the protocol timer to give results that can be used to give more physical information.
3. *Repeatability:* Whether the same query will result in the same answer.

## The axioms for $E((0,1),\mathcal{G})$

Let $\mathcal{G}$ be a class of functions, where each $g \in \mathcal{G}$ is a function $g : \mathbb{N} \to \mathbb{N}$.

1. *Real values:* The experiment is designed to find a physical parameter $x \in (0,1)$.
2. *Queries:* Each query string is interpreted as a binary string of 0s and 1s, $y_1 y_2 \ldots y_k$, giving a dyadic rational $y = 0{\cdot}y_1 y_2 \ldots y_k$.
3. *Finite output:* The result is either $y < x$, $y > x$ (correctly assigned) or *time out*.

## The axioms for $E((0,1),\mathcal{G})$

Let $\mathcal{G}$ be a class of functions, where each $g \in \mathcal{G}$ is a function $g : \mathbb{N} \to \mathbb{N}$.

1. *Protocol timer:* There is a $g \in \mathcal{G}$ so that the time taken for the query $y_1 y_2 \ldots y_k$ is $g(k)$.
2. *Sufficiency of the protocol:* If $|x - y| > 2^{-k}$, then the result is either $y < x$ or $y > x$.
3. *Repeatability:* The same query will result in the same answer.

# The computational power of the oracle $E((0,1), EXP)$.

Take $EXP$ to be the class of functions that are bounded above and below by exponentials of the form $A\,2^{kn}$.

Theorem: Given a word decision problem in $P/\log *$, there is an $0 < x < 1$ so that any oracle in $E((0,1), EXP)$ for the real value $x$ can be used by a Turing machine to solve the problem in polynomial time. Conversely, any decision problem that can be solved by a Turing machine with oracle in $E((0,1), EXP)$ in polynomial time is in $P/\log *$.

# How to make an $E((0,1), EXP)$ oracle

Theorem: Suppose we are given an experimental procedure to measure $x \in (0,1)$ in the following manner. Suppose there are rationals $A, B, E > 0$ and integers $n, q, p \geq 1$ so that

(a) Given $y = 0{\cdot}y_1 y_2 \ldots y_k$ we can set up the experiment with a test value $y' \in \mathbb{R}$ with $|y - y'| < 2^{-s}$ in time $2^{p \max\{s,k\}} E$.

(b) We can run the experiment to determine if $x < y'$ or $x > y'$ in time

$$\frac{A}{|x - y'|^q} \leq T_{\text{experiment}}(x, y') \leq \frac{B}{|x - y'|^n} .$$

Then there is a physical oracle using the experiment which obeys the axioms for $E(x, \mathrm{EXP})$.

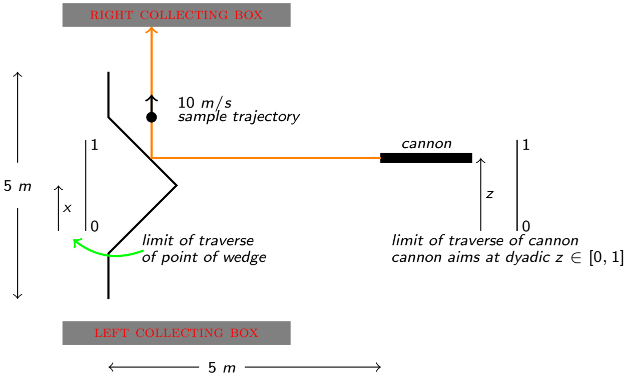
# A Gallery of Measurement Experiments

**Type I: Two-sided experiments**

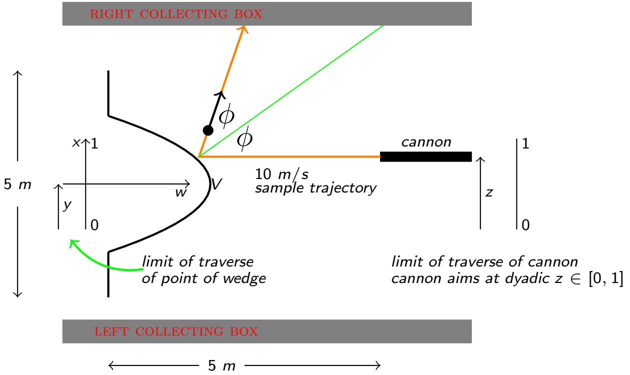
A qualitative output is allowed in each single experiment: yes/no, right/left, north/south, etc.; possibly adding no answer.

The experiment ultimately implements a decision procedure such as less than, possibly leaving equality undecided, or leaving less than semidecided.

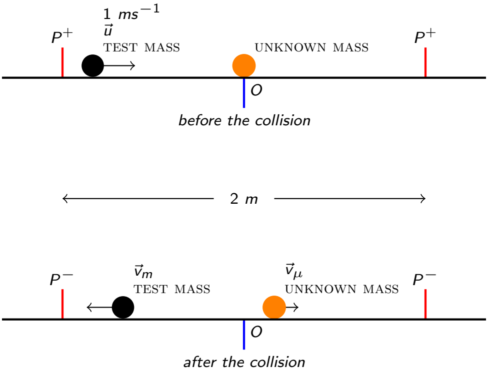
# Scatter machine, Beggs and Tucker (2007)



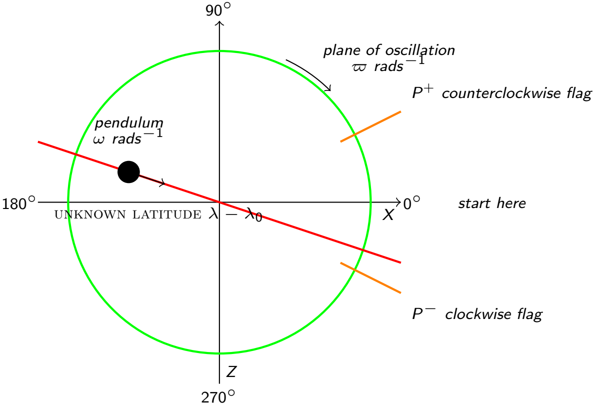
# Generalized scatter machine



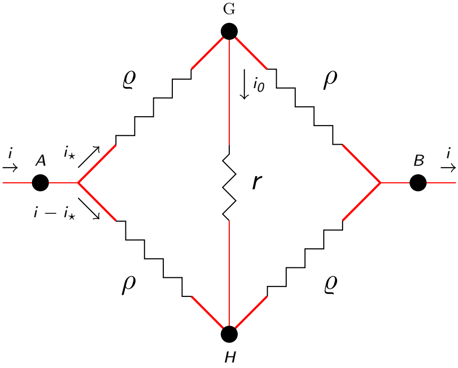
# Collider experiment



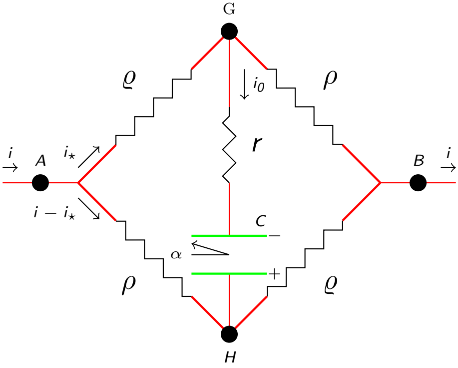
# Foucault's pendulum



# A Wheatstone bridge made simple



# A Wheatstone bridge made simple, with a capacitor



# The Concept of Measurable

## Geroch and Hartle I

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

Robert Geroch and James B. Hartle, Computability and physical theories, *Foundations of Physics*, 16(6), 1986.

## Geroch and Hartle II

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

# Geroch and Hartle III

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

## Geroch and Hartle IV

*The notion "measurable" involves a mix of natural phenomena and the theory by which we describe those phenomena. Imagine that one had access to experiments in the physical world, but lacked any physical theory whatsoever. Then no number $w$ could be shown to be measurable, for, to demonstrate experimentally that a given instruction set shows $w$ measurable would require repeating the experiment an infinite number of times, for a succession of $\varepsilon$s approaching zero.*

## Geroch and Hartle V

*One could not even demonstrate that a given instruction set shows measurability of any number at all, for it could turn out that, as $\varepsilon$ is made smaller, the resulting sequence of experimentally determined rationals simply fails to converge. It is only a theory that can guarantee otherwise. The situation is analogous to that of trying to demonstrate that a given [...] program shows some number to be computable. There is no general algorithm for deciding this. In particular, it would not do merely to run the program for a few selected values of $\varepsilon$.*

## Geroch and Hartle VI

*Every computable number is measurable. This is easy to see: Let the instructions direct that the raw materials be assembled into a computer, and that a certain [...] program — one specified in the instructions — be run on that computer. That is, every digital computer is at heart an analog computer.*

## Geroch and Hartle VII

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

# Computable concept of measurement

## Concept of measurable

**Definition**

A mass is said to be measurable if there exists a Turing machine, equipped with a computable schedule $T$, such that it prints the first $n$ bits of $m$ on the output tape in less than $T(n)$ time steps without timing out in any query. Similarly it is said to be feasibly measurable if $T$ can be chosen to be time constructible.

## Universality of binary search

**Theorem of universality of binary search for TYPE 1 experiments**

If a mass (or a position, or a resistance, etc.) is measurable, then it is measurable by binary search.

## Measuring mass

**Proposition**

*There are programs $N_k$ (with integer $k \geq 1$), with specified waiting times (say $T_k$), so that the following is true: For any non-dyadic value $\mu \in (0, 1)$ and any $n > 0$, there is a $k$ so that the program will find the first $n$ binary places of $\mu$.*

## Measuring mass

**Proposition**

*There are uncountable many $\mu \in [0,1]$ so that, for any program P with specified waiting times, there is a n so that P can not determine the first n binary places of $\mu$.*

# Measurable masses

**Proposition**

*For the CME with unknown mass $\mu$ (not a dyadic rational), written according to the pattern:*

$$\mu = 0{\cdot}\underbrace{1\ldots1}_{u_1}\underbrace{0\ldots0}_{u_2}\underbrace{1\ldots1}_{u_3}\underbrace{0\ldots0}_{u_4}\underbrace{1\ldots1}_{u_5}\underbrace{0\ldots0}_{u_6}\ldots$$

*where $u_1 \geq 0$, $u_i \geq 1$ $(i \geq 2)$.*

1. *If $\mu$ is measurable by any program, then the sequence $u_k$ is bounded by a computable function.*
2. *If the sequence $u_k$ is bounded by a computable function, then $\mu$ is measurable by the linear search method.*

## The Busy Beaver

**Proposition**

*There are functions that grow faster than any computable function. The Busy Beaver function is one of these functions.*

# The Church-Turing Thesis

**Turing's Theorem — A standard CT Thesis**

Every function computable by an abstract human being following an algorithmic procedure is Turing machine computable.

## Conclusions

1. In this lecture we explored an unexpected limitation of physical theories, whenever scientists are no more powerful than Turing machines:
2. The Turing experimenter, in the world of continuous physical variables, (even) theoretically equipped with infinite precision instruments, does not have access to the values of quantities above a finite number of bits, rendering the infinite inaccessible (even) in *gedankenexperimente*.