Computing with Newtonian machines

E.J. Beggs & J.V. Tucker

University of Wales Swansea, UK

• What laws of physics are needed to imply that any system obeying them exhibits computable behaviour?

• In exactly what sense can we say that the 'Laws of Physics' are computable? How can we write programs modelling them?

• Can we *really* build a hypercomputer (a machine that computes more than a Turing machine)?

All these questions require a careful study of exactly what model of physics we use. Rather than model the entire universe, we use a *fragment* of physical laws, and examine machines operating under them.

E.J. Beggs & J.V. Tucker, ‘Computations via experiments with kinematic systems’, Technical Report 5-2004, Department of Computer Science, University of Wales Swansea, March 2004.

E.J. Beggs & J.V. Tucker, ‘Embedding infinitely parallel computation in Newtonian mechanics’, Journal of Applied Mathematics and Computation, 178/1 (2006)

E.J. Beggs & J.V. Tucker, ‘Can Newtonian systems, bounded in space, time, mass and energy compute all functions’, Theoretical Computer Science, in press.

E.J. Beggs & J.V. Tucker, ‘Experimental computation of real numbers by Newtonian machines’, Swansea Maths Research Report 06-11 or CSR9-2006, Dept. of Computer Science.

• There are uncountably many subsets $A \subset \mathbb{N}$ of the natural numbers, therefore there are subsets which are not computable, i.e. for which there is no program which can determine if any $n \in \mathbb{N}$ is in the subset in finite time.

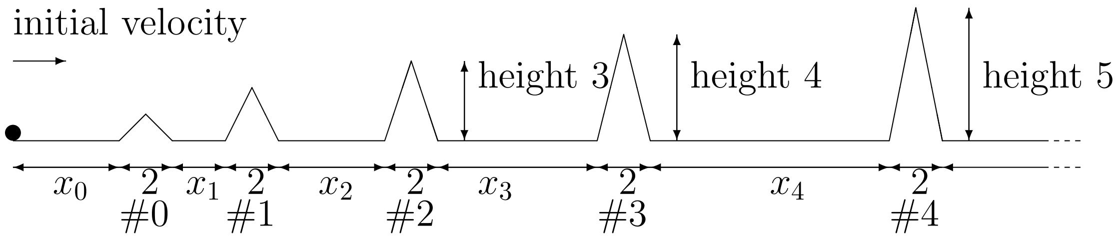
However there are machines for determining noncomputable subsets.

We consider a bagatelle game, operating under Newton's laws and a uniform gravitational field. A ball is fired into the bagatelle machine with a specified velocity, and the ball may or may not return in a given time period. Nothing else about the bagatelle is externally observable. The instructions for operating the bagatelle consist of a list of velocities V_1, V_2, V_3, etc. and a list of times T_1, T_2, T_3, etc. (independently of the subset).

Each machine can define a subset A of the natural numbers $\mathbb{N}$ as follows: Given $n \in \mathbb{N}$, you fire a ball into the machine at initial velocity V_n, and the ball returns in a time $\mathrm{Return}(V_n)$. Then

$$\begin{aligned} n \in A &\quad \text{if and only if} \quad \mathrm{Return}(V_n) \leq T_n \ , \\ n \notin A &\quad \text{if and only if} \quad \mathrm{Return}(V_n) \geq T_n + 1 \ . \end{aligned}$$

Inside the bagatelle:

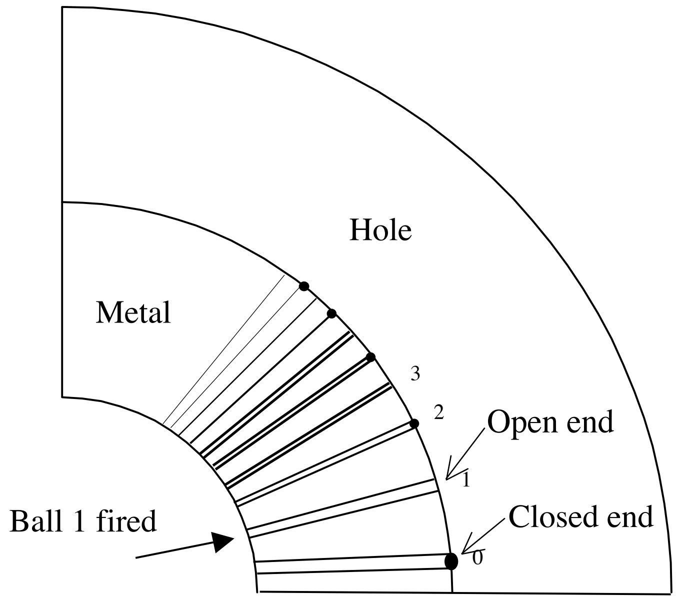


The machine continues indefinitely off the right hand side. For every $n \in A$ we have a barrier $\#n$ of height n positioned as shown. The ball continues to travel right, until it hits a barrier which it has not enough energy to cross, and then it is reflected. The distances x_n between the barriers are chosen so that the ball return timings are correct.

- Note that the bagatelle requires
  - arbitrarily high energies (V_n increases as n increases),
  - arbitrarily large times (T_n increases as n increases),
  - and is of infinite physical extent.

Can we design another machine without these problems?

The marble run. To find if $n \in A$, take the ball labelled n and fire it along track n with velocity 1. If the ball returns within time 3, then $n \in A$, if not then $n \notin A$.



• The price that we have paid for the boundedness in time and space is the introduction of arbitrarily small components (in this case, narrow tracks for the balls to run on).

By now, the listener has probably spotted the weak point in both the previous machines. They require a *deus ex machina*[1]. We might dig up an example of the machines from a mysterious archaeological site, and then use it. But how could we actually make one? Would this not require a knowledge of the subset that we wish to compute?

This problem of construction is vital - we must consider not only whether the machine can run under some fragment of physical theory, but whether it could be built under the same assumptions.

However there is an interesting question which we can formulate from these two machines:

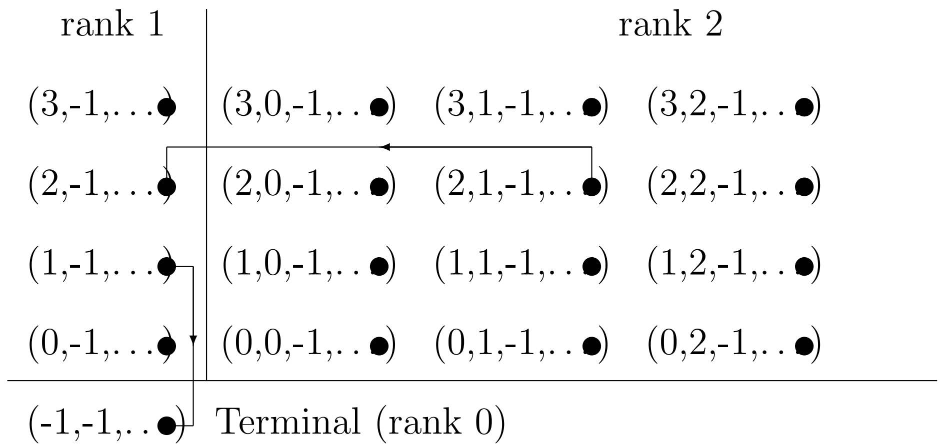
♣ Is it possible to have (not necessarily to make) a machine which can determine an non-computable subset of $\mathbb{N}$ which is:

- bounded in both large and small physical dimensions.
- doesn't require arbitrarily small intervals or infinite amounts of time.
- has components bounded both above and below in mass and energy.

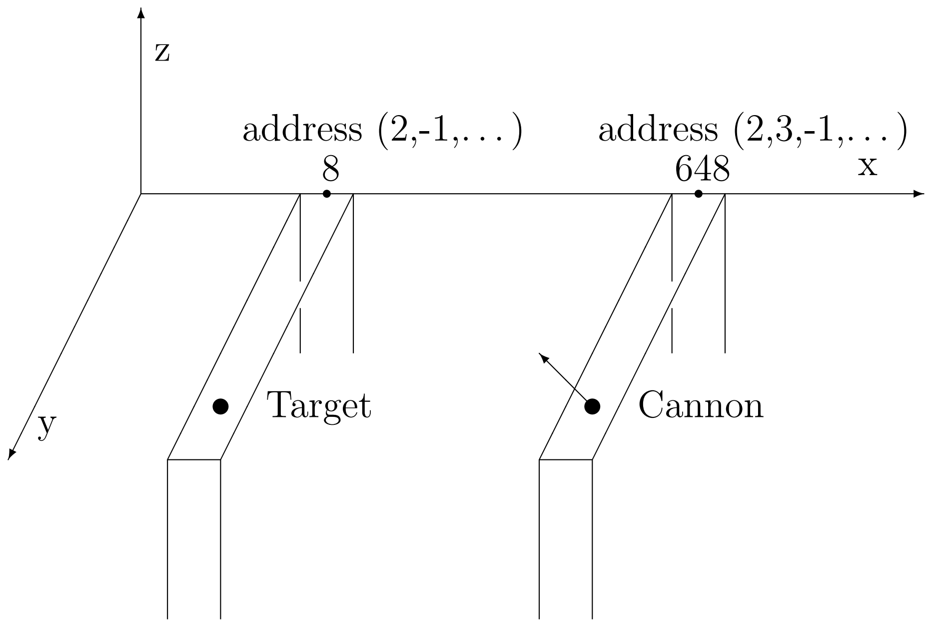
[1]in classical Greek drama, a god who would appear at the end of the performance and set the world to rights

The Infinite machine: A fully programable machine, to address the construction problem.

Consider the set L of sequences of integers, consisting of of natural numbers followed by infinitely many -1s. For each element $\underline{n}$ of L we have a 'node' $C_{\underline{n}}$, consisting of a register machine (identical except for their address registers) and a communications bus. The nodes are organised into ranks, where the rank is the number of natural numbers preceeding the first -1. Typical communications are shown by arrows.



How can we physically implement this machine? With nodes (of possibly infinite extent) placed along an axis according to prime power coding of their address, with cannons for communications, and lots of gunpowder for long flight paths. A rigid rod along the x axis allows for similtaneous programming of all the nodes.



A sample program

Suppose that $f(x_1, \ldots, x_n)$ can be calculated in m time steps. Each node checks whether it received an input and send an output in r steps. The terminal can checks whether it received an input, and prints an output in t steps. Now, consider a case with run time $m + 3\,r + t$ steps:

$$\exists x_1 \in \mathbb{N}\ \forall x_2 \in \mathbb{N}\ \exists x_3 \in \mathbb{N}\ f(x_1, x_2, x_3)$$

1) The node at $(x_1, x_2, x_3, -1, \ldots)$ calculates $f(x_1, x_2, x_3)$ while the other machines wait .

2) The node at $(x_1, x_2, x_3, -1, \ldots)$ sends a message to $(x_1, x_2, -1, \ldots)$ and stops if its calculation had value *true*, otherwise stops.

3) The node at $(x_1, x_2, -1, \ldots)$ determines if it received a message. If not, it sends a message to $(x_1, -1, \ldots)$ and stops, otherwise stops.

4) The node at $(x_1, -1, \ldots)$ determines if it received a message. If not, it sends a message to $(-1, \ldots)$ and stops, otherwise stops.

5) The node at $(-1, \ldots)$ determines if it received a message. If it did, it prints 'true', otherwise it prints 'false'.

Of course, the infinite machine has problems:

• It is of infinite[2] spatial extent and has infinitely many components. However it is 'locally finite', and so could be built by an infinite number of gangs of workmen in finite time.

• The nodes have to do operations on arbitrary length integers in one time step. This *should* be able to be implemented by a machine with registers an infinite sequence of holes containing one ball, with operations implemented by rigid rods. (Remember that the nodes can be of infinite spatial extent.) However this will probably end up with arbitrarily small diameter rigid rods.

If you still don't like our machine, you are invited to read 'Building infinite machines'[3] by E.B. Davies, in which he discusses the difficulty of actually building a Turing machine...

[2]'In an infinite Universe, anything, even the Hitch Hikers Guide to the Galaxy, is possible', Douglas Adams

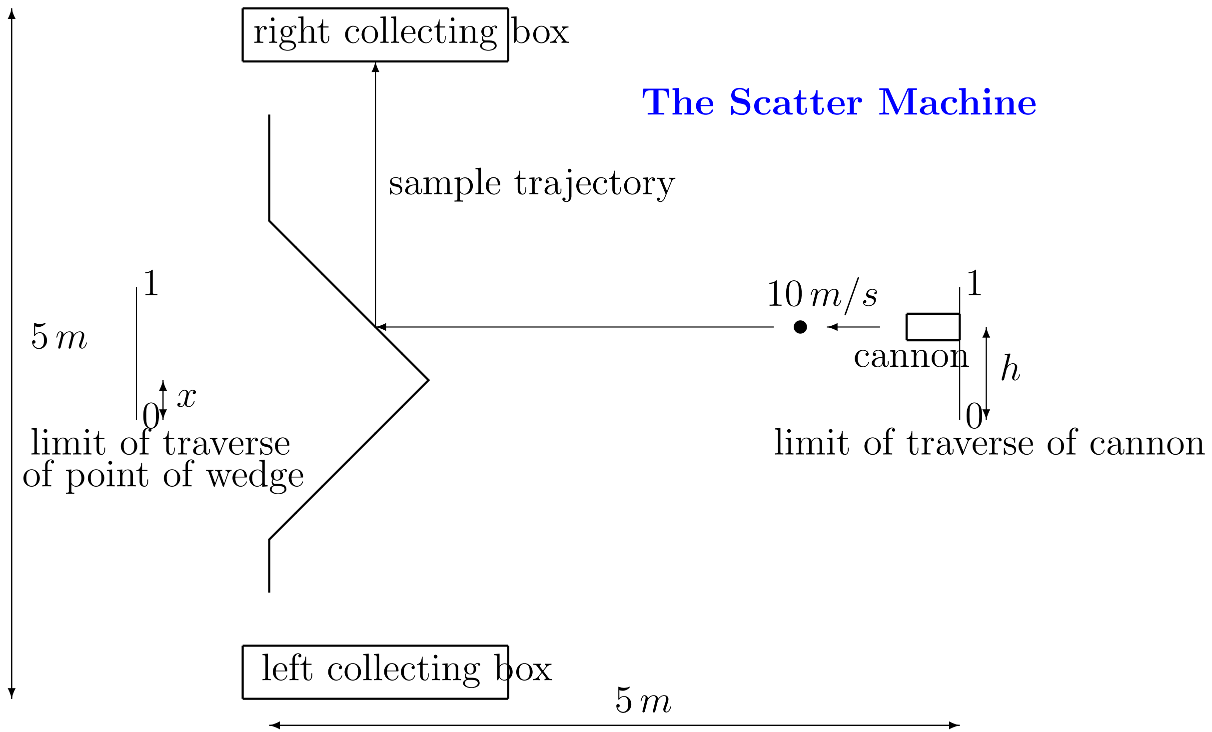
[3]British Journal for the Philosophy of Science 52 (2001) 671-582

Enough of inventing cunning devices - back to what we *can* compute! To formulate an exact computational model of a fragment of physical laws, we should start with a simple system. How about Newton's laws of motion for point particles in a two dimensional world consisting of finitely many straight lines? Well, even this is not quite so simple as it seems...

• A real number $x \in \mathbb{R}$ is computable if there is a program which, given any error $\epsilon > 0$ outputs a rational number $q \in \mathbb{Q}$ so that $|x - q| < \epsilon$. As there are uncountably many real numbers, there must be real numbers which are not computable.

• Given two computable reals x and y, we can compute the propositions $x > y$ and $x < y$, but not whether $x = y$ or $x \geq y$.

The Scatter Machine



Given a wedge position $0 \le x \le 1$, set the cannon position $0 \le h \le 1$, fire, and see which box the cannonball ends up in.

a) After firing, the particle is in the right box. Conclusion: $h \geq x$.
b) After firing, the particle is in the left box. Conclusion: $h \leq x$.
c) After firing, the particle is in neither box. Conclusion: $h = x$.

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

Theorem Any $x \in [0, 1]$ can be calculated by the dyadic scatter machine.
Proof: Specify an accuracy 2^{-N}. Start with $n = 0$, $h_0 = 0$, $h'_0 = 1$.
(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 a,b,c 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 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 outcomes are right, $h_{n+1} = h_n$, $h'_{n+1} = (h_n + h'_n)/2$, goto (7).
(6) If all outcomes are left, $h'_{n+1} = h'_n$, $h_{n+1} = (h_n + h'_n)/2$, goto (7).
(7) Increment n and go to (1).

The output is 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. □

Predictability In the previous case, we did not try to model what happened if the particle hit the point of the wedge. Now we shall only make the assumption of translation invariance, that is that if the system displays a given behavior at one case where $x = h$, then it displays the same behavior at every $x' = h'$. There are two cases:

• The result of a particle hitting the point gives a single predictable result, in that the particle is reflected at a single speed and angle.

• The result is unpredictable, in that there are at least two possible results given identical starting conditions.

♠ 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).

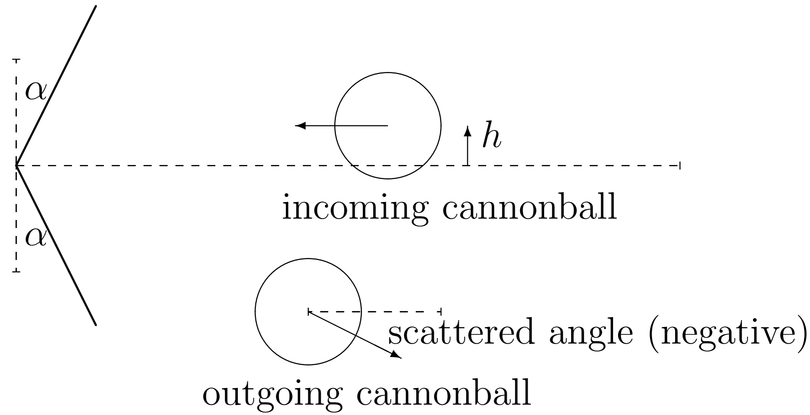
Note that we cannot escape the consequences of this by only allowing h to be a dyadic rational. The proposition $x \leq \frac{1}{2}$ is not computable for a general computable number x.

Can we ensure that the wedge is positioned at a non-computable number? The easiest way is to appeal to probability. We employ a random process, for example Brownian motion. The outcome would be a probability distribution for the position of the wedge. Would it be reasonable for such a probability distribution of a random variable X to have probability one of giving a computable number?

$\star$ Assume that we have a probability distribution on the computable numbers in $\mathbb{R}$. The set of computable numbers is countable, so there must be a computable number c with $P[X = c] > 0$. If the probability distribution were continuous, there would be a $\delta > 0$ so that, for all computable c' with $|c' - c| < \delta$, $P[X = c'] > P[X = c]/2$. But there are infinitely many such computable c' with $|c' - c| < \delta$, so by using the additivity we get $P[X \in (c - \delta, c + \delta)] > 1$, a contradiction.

We deduce that if there is a probability distribution only taking values on computable numbers, then it must be discontinuous with respect to the usual metric topology on the computable numbers. But then it cannot be a computable function of the computable numbers!

You may think that having finite sized spheres as cannonballs, and rounding off a few corners will remove these problems. However consider the following example for scattering a radius $r > 0$ cannonball.



In this case we have a discontinuous scattered angle:

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

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.

Summary

- If you are not precise about what you mean by computation, terrible things happen.
- You must differentiate between what can exist, and what can be made.
- Need for methodology based on
  - lots of examples
  - rigorous mathematical analysis
  - formal studies of physical theories