Unifying science through computation: Reflections on computability and physics

Edwin J Beggs [1], José Félix Costa [1,2,3], and John V Tucker [1]

[1] School of Physical Sciences,
Swansea University, 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
[3] Centro de Matemática e Aplicações Fundamentais do Complexo Interdisciplinar,
Universidade de Lisboa,
Lisbon, Portugal

Abstract

Many activities of a contemporary working scientist involve the idea of the unity of science. There are countless examples where the ideas and methods of one subject find application in another. There are subjects that comfortably straddle the border between disciplines. There are problems that can only be tackled by multidisciplinary approaches. Science is a loose federation of diverse intellectual, experimental and material communities and cultures. However, these cultures are strong.

In this paper we reflect upon an area of research that is attracting the attention of computer scientists, mathematicians, physicists and philosophers: the relationship between theories of computation and physical systems. There are intriguing questions about the computability of physics, and the physical foundations of computability, that can set the agenda for a new subject, and that will not go away. Research is in an early phase of its development, but has considerable potential and ambition.

First, we will argue that concepts of computability theory have a natural place in physical descriptions. We will look at incomputability and (i) the idea that computers "exist" in Nature, (ii) models of physical systems and notions of prediction, and (iii) hypercomputation. We will reflect upon computability and physics as an example of work crossing the frontiers of two disciplines, introducing new questions and ways of argument in physics, and enabling a reappraisal of computers and computation. We will also notice the social phenomenon of

suspicion and resistance, as the theories are unbalanced by their encounter with one another.

1 Introduction

Scientists are surrounded by references to the unity of science. They can be found in tales about the historical development of science, and in the theories and practices of contemporary science. Long ago, biology was invaded by chemistry; chemistry was invaded by physics; and, long before, physics was conquered by mathematics. References can also be found in public policies about science — some sort of unity must be assumed to make sense of the ever fashionable desire for interdisciplinary and multidisciplinary research, for example. New subjects are born of old, such as computer science of mathematics, electronics, logic and linguistics. Computer science is trying to invade everything.

Historically, there are plenty of examples where the ideas and methods of one subject find application to problems in another. In judging the application, the "distance" between the subjects involved and the scope for new developments from the application are important criteria: the further apart, the more remarkable; the larger the legacy, the more significant. Perhaps, a whole new subject is formed that straddles the border between disciplines.

Mathematicians and physicists have a deep faith in the unity of their disciplines and make use of this unity in their research. They are blessed with long memories and long term goals. There are extraordinary examples in mathematics and physics. In mathematics, the 20th century saw algebra and topology combine with dramatic effects. Poincaré's fundamental group of a space is indeed a beautiful innovation but it is an amazingly humble origin for diverse new mathematics, including concrete topics, such as *combinatorial group theory* and *knot theory* to grandiose theories of everything, such as *category theory.* In physics, there is the study of conservation of energy from Grove to Helmholtz, or the on-going search to unify quantum theory and relativity for a physical theory of everything. Between mathematics and physics, the emergence of non-Euclidean geometry and its subsequent role in relativity is an example of unity. Mathematics and physics export great deal to other disciplines, but it all takes a very long time.

Computation is a unifying force in science: computers and software are *everywhere.* Why? Quantification, a fundamental process of science, rests upon the collection, generation, storage, processing and interpretation of data. *Therefore*, technologies for data have long been essential. Computer Science is the new discipline whose core concepts are data and algorithm. Actually, because of quantification, *the concepts of data and algorithm can be found everywhere.* What do our fundamental *theories* of data and algorithms have to offer science?

Computability theory, founded by Church, Turing and Kleene in 1936, is a deep theory for the functions computable by algorithms on particular forms of finite discrete data, such as strings over $\{0, 1\}$ and natural numbers $\{0, 1, 2, \ldots\}$. Digital data is *precisely* the data that can have finite representations, coded by strings or naturals. Computability theory has been extended to arbitrary data via generalisations to abstract algebras ([59, 61]), and, in particular, to continuous data, such as real numbers, via approximations ([61, 62]). Computability theory is at the heart of our understanding of data and algorithm. What has computability theory to offer scientific understanding?

In this paper we reflect upon the relationship between *theories* of computability and physical systems. There are intriguing questions about the computability of physics, and the physical foundations of computability, that can set the agenda for a new subject — they are questions that will not go away. We have a great deal knowledge to call upon. It is an area of research that is attracting the attention of computer scientists, mathematicians, physicists and philosophers. Research is in an early phase of its development, but has considerable potential and ambition.

First, we will argue that concepts from computability theory have a natural place in physical descriptions: we show how abstract machine models with oracles can frame models of the solar system. A connection between computability and physics introduces a connection between *in*computability and physics. We discuss three "causes" of incomputability:

1. *partial or insufficient information for computation*;
2. *unpredictability of properties of a model*;
3. *hyper-computational phenomena in the Universe.*

We reflect upon computability and physics as an example of work at the frontiers of two disciplines. It includes the introduction of new questions and ways of reasoning in physics, and enables a re-appraisal of what makes a computer. We also notice the interesting social phenomenon of suspicion and resistance, as the theories are knocked off balance by bumping into one another.

There is a great deal of background to this theoretical task. Computability Theory is being redeveloped, even reborn. The mathematical subject that was created by philosophical problems in the foundations of mathematics in the 1930s has seen fantastic technical developments and spectacular applications since then. One need only ponder Rogers text-book [53] of 1968(?). But, for a period of at least twenty years, roughly the 1970-90s, its intellectual vibrancy and centrality has been eclipsed by its intense technical development. Technicians forgot or avoided old messy debates about what computability theory is actually about, and whether it has useful consequences for mathematics and science generally. The image of computability theory was dominated by its internal technical agenda and achievements, perhaps most extremely expressed by *generalized recursion theory* (see [11, 16, 17]). For many computability theorists thinking about old debates, unfinished business, new applications and the education of young scientists, the prevailing technocratic view was to become an irritating problem. In the broad community of researchers in mathematical logic, theoretical computer science and philosophy, computability theory was considered a corpse for pure mathematicians to dissect.

The rebirth of computability theory involves a large scale investigation of fundamental questions of about computation. Are physical systems computable?

What can computers, based upon the new technologies of quantum information, optics, etc., compute? These questions come from outside computability theory and confront it with questions that will not go away and uncomfortable notions like hypercomputation — can the technologies compute more, or more efficiently, than Turing machines? From inside computability theory, Cooper and Odifreddi have pressed for the exploration of how Turing's universe lies embedded in Nature ([11]).

Of course, the classical mathematical theory had long been at home with the non-computable; for example, through various kinds of hierarchies (arithmetic, hyperarithmetic, and analytic [28]) and, of course, the study of non-computable Turing degrees such as $0', 0'', \ldots$, to say nothing of the extremes of generalisations of computability theory to ordinals, searching for priority arguments to solve analogues of Post's Problem.

However, the new approach to the foundations of computability leads to fundamental questions about what are non-computable sets and functions; these lead to debate and controversy: "hyper-computation" becomes a forbidden concept because it corresponds to a concept of implementable that has the potential to contradict the Church-Turing Thesis, the primary legacy of the theory. We comment briefly on the origins of this criticism and misinterpretations of concepts such as super-Turing computational power.

Our reflections here are an initial attempt to examine the wider context of our current research programmes [1, 2, 8, 3–7].

2 Computability in Nature: Stonehenge as a calculator with oracles

Let us reflect on a seemingly complicated example of computation intimately related with Nature.

The astronomer Fred Hoyle showed in [30, 31] that Stonehenge can be used to predict the solar and the lunar eclipse cycles. Now, for our purposes, it *does not matter* whether the Ancient Britons did, or did not, use this huge monument to predict the eclipse cycles; but it *does matter* that we, in our times, can use Stonehenge to make good predictions of celestial events like the azimuth of the rising Sun and of the rising Moon, or that we can use this astronomical observatory as an eclipse predictor. Hoyle's method is based upon the structure called *Stonehenge I*; important in this computational task is the alignment of the Heelstone with the summer solstice, and the circle of *Aubrey holes*, made of 56 stones, buried until the 17th century, and discovered by the antiquary John Aubrey.

Hoyle's algorithm makes use of three counters for the task: the first counter, one little stone representing the sun, counts the days of the year along the circle of 56 Aubrey holes; the second counter, representing the moon, counts the days of the lunar month; finally, a third counter, takes care of the Metonic cycle, in which the same phases of the moon are repeated on the same date of the year to within an hour or so after a period of nineteen years, a fact discovered by

Meton around 430 BC though it is believed to have been known earlier. In other words the third small stone counts along the cycle of the lunar node, one of the intersection points of the ecliptic with the Moon's orbit.

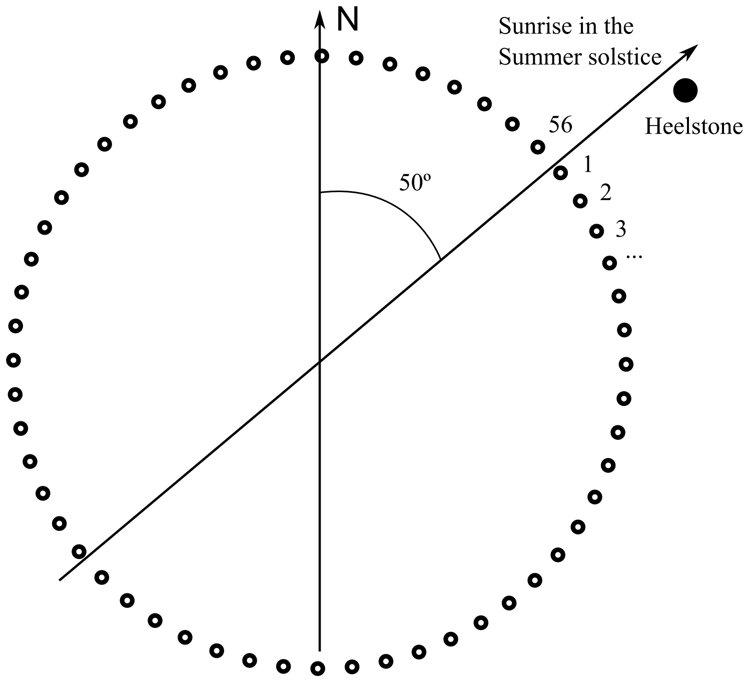

Fig. 1. A schematic drawing of Stonehenge I.

Since $56 \times \frac{13}{2} = 365$, the first counter has to move two places — two Aubrey holes — each 13 days (one place per week roughly speaking), counterclockwise. In a similar way, since $56 \div 2 = 28$, the second counter is allowed to move two places per day, counterclockwise. When the two counters meet at the same hole an eclipse becomes possible, but only if the Sun and the Moon are close to the lunar's node — intersection point of the ecliptic and the Moon's orbit. This point is represented by the third counter. Thus, the three counters have to meet at the same hole (more or less). This third little stone counts along the Metonic cycle: $56 \div 3 = 18.67$ (very close to the true value 18.61, a most strange coincidence), meaning that it has to move 3 places — 3 Aubrey holes — per year, clockwise.

Thus the movement of the three stones around the circle of Aubrey holes allows us to predict the solar and the lunar eclipse cycles. The seminal paper, in which Stonehenge is given an astronomical interpretation as a predictor of eclipses, was by the archeologist Hawkins, in [26], but the mathematical calculations were done by Hoyle, years later. See also [38] for a short introduction.

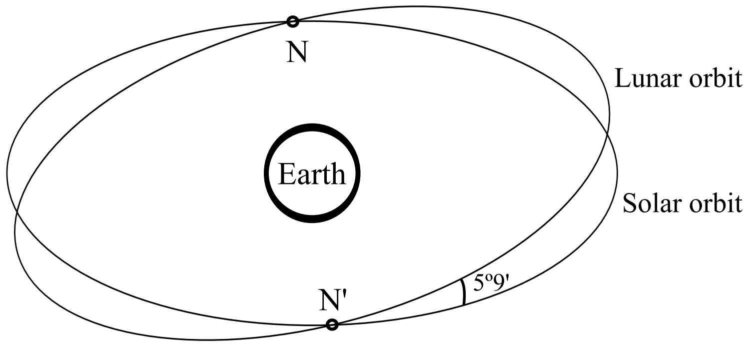

Fig. 2. Orbits.

By calling Hoyle's method Hoyle's *algorithm*, we have introduced the idea of re-interpreting it using concepts from the theory of computation. What model of computation fits best Hoyle's algorithm? What have we got? The algorithm is based upon a simple machine with 3 counters and a memory so the n-counter machines[4] come to mind. In computability theory, Stonehenge I and Hoyle's algorithm shows that a 3-counter machine implements an eclipse cycle predictor using arithmetic modulo 56. Actually, we can also be more restrictive and think in terms of finite state automata; or we can be more general and involve Turing machines and register machines (which are "equivalent" to n counter machines for $n \geq 2$).

A quite straightforward algorithm is implemented by a special purpose analogue machine. It is analogue in the original sense that it is *analogous to the physical system it calculates*. When playing with the counter machine do we "see" in the movements of the three pebbles the Sun, a physical body, and the Moon, another physical body, and (the words are taken from Hoyle) a *holy spirit*, the lunar node, performing a dance in the sky, one that is projected onto the celestial sphere from earthly Wiltshire? Yes, if we have a Rite associated to the ballet of the stones.

However, this is not the whole story. Over time, the calculations with the counters loose accuracy. Once in a year, the Sun rises over the Heelstone. Some auxiliary stones (*the post holes*), to one side of the Heelstone, can be used to fine tune the counters: the site of the rising mid-summer Sun moves to the north and then back to the south, allowing us to fine tune the Sun's counter by observing from the center through the post holes its maximum azimuth. Some auxiliary stones also help to fine tune the second counter. The observations of the Sun and of the Moon operate like *oracles*. Thus, the calculating master, in the centre, can calculate the eclipse by the less accurate algorithm together with an oracle

[4] An n-counter machine is a register machine with n registers for natural numbers and simple assignments based upon successor and predecessor, and jumps based upon equality.

for the Sun and a second oracle for the Moon. Thus, *if the Ancient Britons were "aware" of algorithms then they were aware of algorithms with oracles!*.

The curious thing is that we have a reality in the sky and an analogous reality in the big circle. The algorithm is captured by the real world in the sense that the real world embeds or realises or embodies the algorithm (c.f. [64]).

Thesis 1 *When we abstract from physical entities, we can find special purpose computers existing in Nature.*

This idea of mathematics and computation residing in Nature can be seen in Galileo's work and we could be tempted to refer to the thesis as *Galileo's principle of natural computation*.

If computability is to be found in Nature then is incomputability to be found also? Cooper and Odifreddi address this idea in [11] for these authors (in)-computability sounds more like an intrinsic limitation of knowledge about the Universe than a physical manifestation of hypercomputation; we will discuss this aspect later.

The idea of *The Universe* that is often used has a disadvantage. When used alone, without specification of the model we have in mind, it conveys the impression that there is a "true nature" of the Universe and that we may know it. A *universe* is simply a *model* of some aspect of the Universe. The word *universe* has the further advantage that it may be used freely and loosely without any need to remind ourselves constantly that The Universe is still mysterious and unknown.

Our Stonehenge model of a fragment of The Universe is not so bad and it certainly makes a memorable connection between Computability Theory and Physics.

Stonehenge can be also used as a calculator and computer for doing arithmetic and (why not) for implementing some more sophisticated algorithms. Moreover, Stonehenge *implements natural phenomena*, the complex movements of the Sun and the Moon in the sphere. Stonehenge possesses the means of consulting oracles in Nature itself. Through observations of the world, oracles here handle some incomputabilities.

We end this section with a question: Do the incomputabilities we mentioned above come out of

(i) an unpredictable behaviour of a model of Nature, or

(ii) a really essential incomputability in Nature?

The latter supposes the existence of the hypercomputational character of some physical phenomenon. The most publicised example of this is what Penrose was looking for in [42–44]. But hypercomputational with respect to what models of computation?

Let us interpret predictability to be the ability to decide on whether or not a model has some property.

Moore was among the first to observe a failure of predictability. We are accustomed to think of chaos as sensitivity to initial conditions, which rules out reliable, reproducible predictions of system behaviour because of numerical precision. Moore showed that essential chaos exists in the sense that infinite precision in the initial conditions will not remove it. He showed that the collection of dynamical maps contains many instances of simulations of universal Turing machines. In other words, any property that a map can have, like being injective, or onto, or having an infinite domain, or having an infinite range, or being total, is undecidable unless it is trivial (cf. Rice's Theorem). Now, in terms of dynamical systems, these questions concern basins of attraction. These basins are, in general, non-computable, i.e., there is no hyper-algorithm that will tell us whether or not a point is in them. Recall Theorem 10 from [37]:

Proposition 2.1 (Moore's undecidability theorem). *The following questions about discrete-time dynamic systems are undecidable.*

(a) Given a point x and an open set A, will x fall into A?
(b) Given a point x and a periodic point p, will x converge to p? Will a dense set of points converge to p?

Incomputability is a riddle. What kind of incomputability should we search for in Nature? Given our discussion, here are three possible causes:

1. *partial or inadequate information for calculations*;
2. *essential unpredictability of properties of models*;
3. *hyper-computational phenomena in the Universe.*

In the first case, we cannot compute because we do not have all the necessary variables (e.g., the Adams-Leverrier discovery of Neptune; hidden variables in the Paris School of Quantum Mechanics). In the second case, we cannot decide algorithmically if properties of models of physical systems hold or not. In the third, Nature performs computations that our algorithms cannot.

Articles like [11] raise questions about the physical nature of computability and about the possibility of exporting new concepts from Computer Science to other sciences.

3 Computability in Nature: n-clocks machine

Let us return to the re-interpretation of Stonehenge's counters using computability theory. The 3-counter machine has a very well known property.

Proposition 3.1 (Universality of n-Counter machine). *There is a Turing universal 2-counter machine.*

The n-counter machines are primitive and troublesome to program. Students play with such machines to get acquainted with a model of computation, doing exercises for calculating the sum, product, etc. How remarkable, then, that the Stonehenge computer, a 2-counter machine, computes the cycle so neatly. Thus, in the Stonehenge case, the counter machine is a kind of natural computer.

Let us consider a more sophisticated machine, one that is not well known but also fits with Stonehenge. The model was introduced in Killian and Siegelmann [32]. Although introduced for rather ad hoc purposes in a proof, the model is attractive when reflecting on natural computing, such as in Stonehenge.

The model of computation is called the *n-alarm clock machine*, or just n-clock machine and is made of abstract clocks with programmable alarms. Time is an abstraction that belongs in the physical world. By an *alarm* in the physical world we can consider things in macroscopic or microscopic worlds that signal special events, e.g., an astronomical ephemeris, like a conjunction of planets, or the reaching of a perihelion, or an eclipse.

A n-alarm clock machine $\mathcal{A}$ consists of n clocks. Each clock $1 \leq i \leq n$ is represented by a pair (p_i, t_i), where $p_i \in \mathbb{N}$ is the period of clock i and $t_i \in \mathbb{N}$ is the next time at which the clock i sounds its alarm. Thus, a state or configuration of the machine is a vector of the form

$$((p_1, t_1), \ldots, (p_n, t_n)).$$

The behaviour of the machine is determined by a transition function, which is a total function that selects sets of instructions to perform on the clocks: we suppose

$$\mathcal{A} : \{0,1\}^{5n} \to 2^{\{delay(i), lengthen(i): 1 \leq i \leq n\} \cup \{halt\}}.$$

that satisfies $\mathcal{A}(0 \ldots 0) = \emptyset$. Intuitively, this latter condition means that the machine must be asleep until it is woken.

The fact that $\mathcal{A}$'s domain is $\{0,1\}^{5n}$ means that $\mathcal{A}$'s input is the information of which alarm clocks have alarmed in the last 5 time steps[5] and when they did so. $\mathcal{A}$'s output is simply which clocks to *delay*, which clocks to *lengthen*, and whether the machine *halts* or not. Let $\delta(t)$ denote such a set of actions at time t.

Given a n-alarm clock machine $\mathcal{A}$, and an initial configuration

$$c(0) = ((p_1, t_1), (p_2, t_2), \ldots, (p_n, t_n)),$$

the computation of $\mathcal{A}$ on the given configuration is a sequence

$$c(0), \ldots, c(t-1), c(t) = ((p_1(t), t_1(t)), (p_2(t), t_2(t)), \ldots, (p_n(t), t_n(t))), \ldots$$

of configurations over time t that satisfies for all t,

$$p_i(t+1) = \begin{cases} p_i(t) + 1 & \text{if } lengthen(i) \in \delta(t) \\ p_i(t) & \text{otherwise} \end{cases}$$

$$t_i(t+1) = \begin{cases} t_i(t) + 1 & \text{if } delay(i) \in \delta(t) \text{ or } lengthen(i) \in \delta(t) \\ t_i(t) + p_i(t) & \text{if } t_i(t) = t \text{ and clock } i \text{ alarms} \\ t_i(t) & \text{otherwise} \end{cases}$$

[5] The number 5 is considered here just because we know the existence of a universal n-clock machine with constant 5. We do not know if there exists a universal n-clock machine exhibiting a smaller structural constant.

The role of the clocks of the alarm clock machine is to store information on the frequency with which they alarm. In Turing machines the tapes are potentially infinite, but at any given instant only a finite amount of information is actually stored on the tape. In the same way, the period of the clocks may increase without limit, but at any given instant all alarm clocks have a period bounded by some constant.

Proposition 3.2 (Simulation of the n-counter machine). *For a n-counter machine that computes in time T, there exists a k-alarm clock machine that simulates it in time $O(T^3)$ with $k \in O(n^2)$.*

In consequence,

Proposition 3.3 (Universality of n-clock machine). *There is a universal n-clock machine, for some n.*

The observation is that this kind of machine can implement astronomical models of the dynamics of the Solar System in a natural way, partially generalising the way the Stonehenge token game implements the eclipse cycle in a natural way. We can add oracles to the machines, say at precise conjunctions of heavenly bodies. In particular, the model is universal and suitable for thinking about physical bodies, e.g., in the Newtonian gravitational field.

Thesis 2 *When we abstract from physical entities, general computers exist in Nature.*

This thesis could be referred to as a *physical principle of general computation.* The standard model of computation, the Turing machine, can be described in an equivalent way, via the n-clock machine model, which resembles the process of making astronomical observations in the manner of the Ancients. Of course, all machine models are abstractions of material components and systems, as are their abstract resources, such as time and space.

The experience of celestial conjunctions described above is not unlike looking at the concept of incomputability as *action at a distance* in the time of Newton. Oracles are needed to fine tune the system once in a while, not only to remove errors (of truncation of real numbers), but also *to remove unpredictability.*

Smith's construction in [58] can be viewed in the following light: It was known that there were mechanical systems whose asymptotic (long time) behaviour was not known — any computation up to any finite time would hit the problem that just because a given event had not happened yet, did not mean that it would not happen in the future. However, Smith's gravitational machine uses the behaviour of point particles approaching arbitrarily closely to allow uncountably many topologically distinct paths of a point particle in finite time. If you can observe which path the particle actually takes, then you can work out, in finite time, much more information about the initial state of the particles than a numerical error bound. A complication is that there is an infinite set of measure 0 initial states close to the original one where a collision of point particles occurs — a situation where Newton's laws cannot predict the outcome. Smith observed that although we are able to show that Newton's gravitation admits a non-computable orbit, and hence a kind of incomputability of the third kind (3), *special relativity removes it from consideration.*

Expressed in a rather different and radical way, the discovery of a non-computable orbit in Newtonian mechanics refutes Newtonian gravitation theory, *because* it contradicts the physical Church-Turing thesis; in the same way, philosophically speaking, the curvature of light rays from distant stars in the proximity of the Sun refute Newtonian gravitation theory. This adds to the philosophic riddles.

If the reader takes a closer look at the discussion on the Church's Thesis in Odifreddi's text books [40, 41] (e.g., pp. 101-123 in [40]), he or she will find different formulations of the classical Church's Thesis, such as Kreisel's Thesis M (for *mechanical*) or Kreisel's Thesis P (for *probabilistic*), and so on. Here, at this precise point of his text we find *In the extreme case, any physical process is an analogue calculation of its own behaviour*. And Odifreddi adds a quite interesting footnote:

> *In this case, Church's Thesis amounts to saying that the universe is, or at least can be simulated by, a computer. This is reminiscent of similar attempts to compare Nature to the most sophisticated available machine, like the mechanical clock in the 17th Century, and the heat engine in the 19th Century, and it might soon appear as simplistic.*

In fact, Thesis P states that *any possible behaviour of a discrete physical system (according to present day physical theory) is computable*. Our various systems disprove this [5–7, 1]. Also Smith disproves this statement: there exists a Newtonian non-computable orbit. It is not relevant that Relativity Theory removes this pathology: no one would ever believe a few years ago that Thesis P would not be valid. Or is it still valid? Well, physicists say, we don't have point masses or two masses can not come as close as we want. We argue that these are qualitatively physical aspects that are *not in* the formulation of Newtonian gravitation. We will restate saying that no one would ever believe a few years ago that Thesis P would not be valid even for the abstract gravitation theory taught in Physics courses.[6] Is there a student's course notes that considers from scratch the problem of two bodies with non-zero volume. But if then Professor X shows that the n-spherical-body problem gives rise to a non-computable orbit, physicists will say that planets are not really spheres. Here are things in need of better explanations.

Thus, when Cooper and Odifreddi write in [11]:

> *Fortunately, there is another approach — let's call it the "mathematical" approach — which renews the link to Newton. This is a direction rooted*

[6] The most comprehensive study we know is a Treatise about stability of a spacecraft, considering 2-body dynamics, on one side the spacecraft with non-zero dimensions and on the other side the Earth just substituted by its centre of gravity.

> *in the old debate about whether computability theory has any useful consequences for mathematics other than those whose statements depend on recursion theoretic terminology.*

We would add the direction is also rooted in the *new* debate about whether computability theory has any useful consequences for physics.

We can say that Nature has an algorithmic content: it is greater than the algorithmic content of the Solar System, greater than the algorithmic content of the system Moon–Sun–Earth, greater than the algorithm content of Stonehenge I. Imagine that Stonehenge IV would have been built then, certainly, it would implement the n-clock machine. Does the Universe, or just the universes, have an algorithmic content greater than the algorithm content of Stonehenge IV, abstracting from bounded resources?

We have made a case for the models of computation being intimately related with physical models and physical behaviour. Let us turn to incomputability.

4 Incomputability and predictability: the discovery of Neptune

Many physical theories provide methods of measurement and calculation. If the calculations are not consistent with measurements then the theory has a problem. The desired measurements are not predicted by the calculations, i.e., they are incomputable. Continuing with the mechanical examples, we will reflect on mechanics, especially Newtonian's theory of gravitation, as a method of calculation. The theme is *preserving and restoring computation* when confronted with incomputabilities.

4.1 Action at a distance

Newton's gravitational law introduced the metaphysical concept of action at a distance. For Newton, action at a distance was done by means of God: space is the *Sensorium Dei* by means of which He stabilizes the system. In his *Opticks*, Newton wrote:

> *... can be the effect of nothing else than the Wisdom and Skill of a powerful ever-living Agent, who being in all Places, is more able by his Will to move the Bodies within his boundless uniform* Sensorium*, and thereby to form and reform the Parts of the Universe, than we are by our Will to move the Parts of our own Bodies. And yet we are not to consider the World as a Body of God, or the several Parts thereof, as the Parts of God. He is an uniform Being, void of Organs, Members or Parts, and they are his Creatures subordinate to him, and subservient to his Will; and he is no more the Soul of them, than the Soul of Man is the Soul of the Species of Things carried through the Organs of Sense into the place of its Sensation, where it perceives them by means of its immediate Presence, without the Intervention of any third thing.*

The removal of *action at a distance* from Physics is not unlike the removal of the Rite in Stonehenge I. The removal of the Rite in Stonehenge I is also like Laplace's removal of the *Sensorium Dei* from Newton's space.[7] But with a *Sensorium Dei* or without it, Smith proved the existence of non-computable orbits: an incomputability of the third kind (3), although the proof is based on work of Gerver in [19], and others for particular cases.

Cooper and Odifreddi raise the question ([11]):

> *Why should those without a direct career interest care whether actual incomputability (suitably formalized) occurs in Nature? Even if it did occur, for all practical purposes, how would it be distinguishable from theoretically computable but very "complex" phenomena? Whether chaotic phenomena — such as turbulence — involve complexity or incomputability is interesting, but does it really "matter"?*

The question is also related to Cooper's ideas in his later ([9]). We think that the answer to this question is not easy.

4.2 Waves

Differential equations do exist, having computable coefficients and given computable initial conditions, which cannot be numerically solved by a digital computer: their solution are beyond the Turing limit. Pour-El and Richards provided examples in [49–51]. For example, they considered the three-dimensional wave-equation in [50]. It is well known that the solution $u(x, y, z, t)$ is uniquely determined by the initial conditions u and du/dt at time $t = 0$. They asked whether computable initial data can give rise to non-computable solutions and gave the answer Yes. They gave an example in which the solution $u(x, y, z, t)$ takes a non-computable value at a computable point in space-time.

However, these examples have initial conditions, or boundary conditions, which are not smooth enough to describe real physical situations.

Are all *physical laws* digitally reproducible by a digital computer? If so, then we may talk about non-computable functions as those functions that can not be known through numerical computation using digital computers, despite the fact that they satisfy very simple differential equations. Calculating positions of planets (ignoring some possible incomputabilities suggested in [58]) was, in fact, a problem of precision. The intrinsically non-computable functions of Pour-El and Richards are of a different kind.

Do we have a model to classify such sources of uncomputability found in [49–51]? No, we don't. Do you imagine an equation — Poisson's equation — as simple as

$$\begin{aligned} \psi(x,0) &= f(x), \\ \frac{\partial^2\psi}{\partial x^2} - \frac{\partial^2\psi}{\partial t^2} &= 0, \end{aligned}$$

[7] Newtonian space, like Descartes' substantial space, was not empty but the *Nervous System of God*.

having a non-computable unique solution (non-computable in the sense of conventional computable analysis): there exists not a program such that giving the values of computable numbers x and t with increasing precision will provide $\psi(x,t)$ with increasing precision, despite existing such a program for the function f.

Penrose rejects these examples as useful to a forthcoming *Non-computable Physics*, since the boundary conditions or initial conditions involved are not smooth enough. In [42] he stresses this fact before considering the non-computable ultimate physical theory to come and the human mind:

> *Now, where do we stand with regard to computability in classical theory? It is reasonable to guess that, with general relativity, the situation is not significantly different from that of special relativity — over and above the differences in causality and determinism that we have just been presenting. Where the future behaviour of the physical system is determined from initial data, then this future behaviour would seem (by similar reasoning to that we presented in the case of Newtonian theory) also to be computably determined by that data (apart from unhelpful type of non-computability encountered by Pour-El and Richards for the wave equation, as considered above — and which does not occur for smoothly varying data). Indeed, it is hard to see that in any of the physical theories that we have been discussing so far there can be any significant "non-computable" elements. It is certainly to be expected that "chaotic" behaviour can occur in many of these theories, where very slight changes in initial data can give rise to enormous differences in resulting behaviour. But, as we mentioned before, it is hard to see how this type of non-computability — i.e. "unpredictability" — could be of any "use" in a device which tries to "harness" possible non-computable elements in physical laws.*

Computation is preserved by declaring that the boundary conditions are not well-posed physically.

Now, what is the consequence of this to Science? Even for *complex phenomena* like the dynamics of the atmosphere we have strong methods of numerical modelling. We take the Navier-Stokes equation and assume (a) spherical coordinates, (b) that the Earth is not an inertial reference frame, (c) boundary conditions around east North-America's shore and West-Europe's and North-Africa's coast. We presume that (i) such differential equations are integrable by numerical methods and (ii) a prediction of the weather for *tomorrow* can be obtained *before tomorrow.* Thus we still have computability considerations and computational complexity considerations. Philosophically speaking, we turn to models of Nature which are predictable. Science in this way is used to make a synthesis of our knowledge about the Universe and to forecast future events. We think that the answers to the questions raised by Cooper and Odifreddi in our last quotation of their article (just before Section 4.2) are "Yes", "We don't know", and "Yes". A non-computable world, like the model desired by Penrose, would have a quite different meaning. Assuming that no more computational power is added to computers, we wouldn't have general predictions. The model would be looked upon as divine: suddenly a pattern formation occurs out of the model and some sophisticated computer programs would be able to trace and forecast its trajectory, like a hurricane that although cannot be exactly predicted can be expected and followed, either by satellites or computer programs. A Non-computable Science would be more like a painting in the National Gallery — to look at with respect, admiration, and fascination, being interpreted the critics (would it meet Susan Sontag's *Against Interpretation*). Maybe the questions of Cooper and Odifreddi become:

(a) Does our contemporary science contain patterns of a non-computable model? or

(b) Do we already have a Non-computable Science, hidden in our theoretical achievements? or

(c) Non-computable Science is no more than contemporary fiction, a motor and product of the creative process, like the *stone* was for Alchemy.

No matter the true answer, they make the concluding statement that:

> *Our model says nothing about the mystery of material existence. But it does offer a framework in which a breakdown in reductionism is a commonplace, certainly not inconsistent with the picture given of levels we do have some hope of understanding. It can tell us, in a characteristically schematic way, how "things" come to exist.* [8]

4.3 Universes

We also know that *modern science* is losing some coherence and identity. There is just one Newtonian gravitation theory,[9] but with the advent of the General Theory of Relativity, physicists realized that Einstein's beautiful field equation

$$R_{ij} - \frac{1}{2}\, g_{ij}\, R \;=\; \kappa\, T_{ij}$$

could be replaced by different field equations delivering the same realities, delivering the same predicted observations of our Universe. Most probably a non-computable model will deliver also a class of similar observations of the Universe. For instance, is Hoyle's or Hoyle-Narlikar's field of creation *ex nihilo* non-computable? This is not philosophy, since Hoyle's field of creation out of nothing is hard mathematics, although it is refutable nowadays, and not accepted by the

[8] The reader can have a look at the Eddington's Cosmic Equation, that was a source of explanation of how the Universe came into existence.

[9] Though, in fact, for some time, physicists were tempted to define the law

$$\frac{1}{r^{2.0\cdots025}}.$$

Ridiculous, isn't it? But it worked for a few years, when physicists lost their faith for reasons that will become clear soon.

scientific community, as the Big Bang Theory is the standard model. Yet, it explains the same observations as the Einstein field equations at some level.

In the 1950s it was a perceived that The Universe was expanding. There was no evidence of the universe showing signs of age. If The Universe was in a steady state then it need to gain hydrogen atoms to preserve the density of hydrogen. Hoyle arrived at the alternative equation

$$R_{ij} - \frac{1}{2}\, g_{ij}\, R\; + C_{ij} = \;\kappa\, T_{ij}.$$

Associated with the creation tensor C_{ij} was a vector field parallel to a geodesic at each point of the homogeneous and isotropically expanding universe. The field was written

$$C_m = \frac{3c}{a}\;(1, 0, 0, 0),$$

where a is a constant. Hoyle then showed that the solution of the field equations would be given by a metric with space of zero curvature. One can interpret the step as an attempt at preserving the form of Einstein's equations and calculations.

Bondy, Gold, and Hoyle used the word creation rather than formation, just to emphasise the existence of matter where none had been before.

With the Hoyle-Bondi-Gold's model we can evaluate the amount of matter being created at any step of time. But can we predict the point in space where a proton (Hoyle guessed that the spontaneous creation of matter might possibly be in the form of neutrons) will next appear? This is an example of how a non-computable aspect of a theory (we cannot even guess a distribution of matter created[10]) can deliver also computable trajectories of our Universe.

There is a more recent discussion on the evaporation of black holes which is also of relevance here. Hawking [24] showed that combining quantum field the-

---

[10] Harrison explains these features in [23]:

> *There are two kinds of creation: creation of the universe and creation in the universe. On one hand, we have creation (as in cosmogenesis) of the whole universe complete with space and time; on the other, we have creation of things in the space and time of an already existing universe. In the Big Bang universe, everything including space and time is created; in the steady-state universe [of Bondi, Gold, and Hoyle], matter is created in the space and time of a universe already created. Failure to distinguish between the two violates the containment principle... The steady-state theory employs creation in the magical sense that at certain place in space at a certain instant in time there is nothing, and at the same place a moment later is something. But the creation of the universe has not this meaning, unless we revert to the old belief that time and space are metaphysical and extend beyond the physical universe; in that case, creation of a universe is in principle the same as the creation of a hazel nut. But in fact uncontained creation (cosmogenesis) is tottally unlike contained creation. Cosmogenesis involves the creation of space and time, and this is what makes it so difficult to understand.*

ory with general relativity gave a prediction that black holes should evaporate - that is they radiate particles which results in a loss of mass to the black hole, until eventually the black hole disappears. Similarly to Hoyle's creation field, this particle creation could be viewed as a creation of particles at an essentially random piece of space in the vicinity of the event horizon. This seeming randomness became the subject of a thought experiment: If an observer were to drop an encyclopedia into a black hole, and then wait until it evaporated, would there be enough information in the particles radiated from the black hole to reconstruct the encyclopedia [25]? This subject of information loss has become hotly debated in recent years.

Cooper and Odifreddi recognizes these different presentations of the Universe stating that:

> *we look for a mathematical structure within which we may informatively interpret the current state of the scientific enterprise. This presentation may be done in different ways, one must assume, but if differing modes of presentation yield results which build a cohesive description of the Universe, then we have an appropriate modeling strategy.*

(See [11].) Furthermore:

> *... non-locality was first suggested by the well-known Einstein-Podolsky-Rosen thought experiment, and again, has been confirmed by observation. The way in which definability asserts itself in the Turing universe is not known to be computable, which would explain the difficulties in predicting exactly how such a collapse might materialize in practice, and the apparent randomness involved.*

4.4 Neptune

The n-clock machine can be implemented with bounded resources in Stonehenge using colored stones, a color for each clock, 5 colored tokens for each clock.[11] It would have made Stonehenge a huge Observatory (although many existent stones — like the post holes — can handle a large number of calculations that, despite the non-existence of a useful — to the Ancients — implementation of the n-clock machine, make Stonehenge I and II a rather huge Astronomical Observatory). But we are going to talk now about a feature that can not be implemented in Stonehenge: the discovery process!

After Herschell's discovery of *Uranus*, deviations from computed orbit, using Gaussian methods, produced more and more observations of the new slow planet, leading to calculations of more and more accurate orbits. But the new planet always failed to meet the computed orbit: Uranus escaped computation: it was incomputable. If T is Newton's theory of gravity then Uranus was not T-computable.

[11] It would be like a Calendar with many entries (cf. Reingold and Dershowitz' *Calendrical Calculations*.

There are two attitudes. First, one accepts the problem for a period but refuses to give up the calculation. The ancients failed in the prediction of planet cycles. Stonehenge fails as Observatory, but perhaps the memory of the glorious Stonehenge I compels the building of Stonehenge III, the colossal construction of central 3-liths.

Secondly, one accepts that the Newtonian law is wrong and begins the search for the new "true" law of gravitation.

But, as we know, it was too early to reject the Newtonian theory of the Universe. Observations failed; the law $\frac{1}{r^2}$ failed. But Leverrier and Adams, one in Paris, the other in London, proposed that a new planet existed — later called Neptune — to justify departure from predicted orbits and to justify the *true* (Newtonian) law of gravitation[12]. Uranus was again T-computable.

This step cannot be done by a computer program[13]. What is the difference (if the planet was not found) between predicting a planet and replacing Newton's law by another law, being it computable or not? Is the discovery of a new planet a kind of *removal of incomputability*?

How did the scientists respond to the predictions of Leverrier and Adams? They rejected them; they didn't believe them. Is it not a common reaction of an established scientist's mind: *if some hypothesis not in the system is suggested, then it should be immediately rejected*. For example, Airy rejected Adams several times: how, we would like to go to Greenwich and knock at the door to hear him saying *no*! As Morton Grosser tells the story in [22], *Airy was an extreme perfectionist, and he divided the people around him into two groups: those who had succeeded and were worthy of cultivation, and those who had not succeded and were beneath consideration [...] Adams solution of the problem of inverse perturbation was thus a direct contradiction of Airy's considered opinion. The Astronomer Royal's negative feelings were indicated by the unusually long time he waited before replying. Airy habitually answered his correspondence by return mail*. In Adams' case he delayed the answer as much as he could.

It would have been enough to look to the sky with a telescope using calculated positions of Neptune.

Feel the pleasure of the following letter of Airy to Adams; it could be adapted to a letter of caution about any thing by an illustrious scientist of our times:

> *We have often trought of the irregularity of* Uranus*, and since the receipt of your letter have looked more carefully to it. It is a puzzling subject, but we give it as my opinion, without hesitation, that it is not yet in such a state as to give the smallest hope of making out the nature of any external action on the planet [...] But [even] if it were certain that there were any extraneous action, we doubt much the possibility of determining the place of a planet which produced it. We are sure it could not be done*

[12] The difficulties lay with the computation of planet path around the sun without interaction by other planets; Uranus was not computable as a 2-body problem.

[13] Well, Herbert Simon said that it can! — at least Kepler's laws can be rediscovered by computer programs, given Tycho Brahe's data; but not the prediction of a new planet; see [52].

> *till the nature of the irregularity was well determined from successive revolutions.*

In a further letter, Airy writes to Adams:

> *We are very much obliged by the paper of results which you left here a few days since, showing the perturbations on the place of* Uranus *produced by a planet with certain assumed elements. The latter numbers are all extremely satisfactory: we are not enough acquainted with Flamsteed's observations about 1690 to say whether they bear such an error, but we think it extremely probable.*
> *But we should be very glad to know whether this assumed perturbation will explain the error of the radius vector of* Uranus*. This error is now very considerable.*

According to [22], on September 18, 1846, Leverrier wrote to Johann Gottfried Galle, assistant to Olaus Roemer. This letter reached Galle on September 23, and he immediately asked his superior, Johann Franz Encke, Director of the Berlin Observatory, for permission to search for the planet. The same night Galle and d'Arrest found the planet: *that star is not on the map* — exclaimed d'Arrest; right ascension $22^h\ 53^m\ 25.84^s$ against the predicted value of Leverrier $22^h\ 46^m$. Although impressive, this accuracy is smaller than Stonehenge's accuracy for the eclipse cycle.

The serendipity of discovery — whether of the kind in the case of Archimedes' *Eureka!*, or in Kepler's laws, or even Kepler's laws according to Herbert Simon's program — are different from the kind of discovery of Neptune.

Thesis 3 *We cannot use Natural laws to make a hypercomputer but we can observe objects whose behaviour is hypercomputational.*

The observation of real number values of physical measurements is a starting point for the argument. We don't say that hypercomputation but incomputability is the cause of the discovery of Neptune. Citing Cooper and Odifreddi, *Science since the time of Newton, at least, has been largely based on the identification and mathematical description of algorithmic contents in the Universe. We will look at phenomena — primarily subatomic phenomena — which appear to defy such description.* The "hidden planet" Neptune was a hidden variable that preserves computation.

This incomputability can be seen with the help of the n-clock machine.

We all know that pendulum clocks are quantum systems: each one has exactly two different energy levels, two oscillatory modes: one with the pendulum at rest and other with the pendulum oscillating in a stable orbit. We all know that clocks on the same wall propagate across the wall sound waves, together with their delays or advances, forcing the (coupled) clocks altogether, to a common delay or advance. In Stonehenge, this effect cannot be seen between colored tokens, but on the human machinery that puts the little tokens in motion. Some people forget clocks when they think about the quantum realm. Quantum mechanics in this way also applies to the macroscopic world (of course, not in the sense of making Planck's constant go to zero![14]), in the sense of operators, eigenvectors and eigenvalues.

It works like the ancients, who with the same teleological thinking, finding a disagreement in the predicted Metonic cycle are compelled them to add a further token to the game.

5 Algorithmic contents of laws

We disagree with a few statements in [11]:

> *In fact ... no discrete model — finite or otherwise — presents a likely host for incomputable phenomena.*

We have at least two exceptions: Wolpert in [65] studies a discrete neural model with super-Turing capabilities, but with a transfinite number of neurons, and Pollack in [46] proved that a model of higher-order neural nets is at least universal. Other results on neural networks involve the real numbers. We have an idea that infinite automata can have super-Turing powers, even not involving the real numbers. Secondly, scientifically presenting the Universe with real numbers is not enough to embed in it super-Turing powers. We are always amazed when we hear that a computational model equipped with real numbers allows for hypercomputation. We will start by "defining" super-Turing power of the scientifically presented Universe.

A physical process takes place in time. It is described or understood using specific observable variables which constitute a notion of state. Therefore, the process consists of states that are measurable, either numerically or qualitatively, evolving in time. What makes such a process a computation? The role of initial states and the structure of the observable histories they generate. The role of data in setting initial states and interpreting behaviour as output.

Thesis 4 *Up to Turing power, all computations in the Universe are describable by suitable programs, which involve the prescription by finite means of rational number parameters of the system or some computable real numbers; the computations can be generated by a program. Beyond Turing power, we have computations that are not describable by finite means; computations that cannot be generated by any program.*

Computation without a program! When we observe natural phenomena and endow them with computational significance, it is not the algorithm we are

[14] It is a good exercise to retrieve

$$\boldsymbol{F} \;=\; m\,\frac{d^2\boldsymbol{r}}{dt^2}$$

from Schröndinger's equation with $\boldsymbol{F}$ given by $-grad\ U$, where U is the classical potential field in the original equation.

observing but the process. Some objects near us may be performing hypercomputation: we observe them, but we will never be able to simulate their behaviour on a computer. What is then the profit to Science of such a theory of computation? The point is that the principle does not tell us about hyper-machines. In this sense hypercomputation can exist. We presume that most of the reactions of the scientific community against hypercomputation are mainly related with *the crazy idea of building a hyper-computer*. We think it is also one of the sources of criticism against the work of Siegelmann and Sontag in [55, 56]

But to help the reader to understand that the real numbers alone are not enough to produce any kind of hypercomputation we call upon Analogue Computation.

In the 1940s, two different views of the brain and the computer were equally important. One was the analog technology and theory that had emerged before the war. The other was the digital technology and theory that was to become the main paradigm of computation (see [39]). The outcome of the contest between these two competing views derived from technological and epistemological arguments. While digital technology was improving dramatically, the technology of analog machines had already reached a significant level of development. In particular, digital technology offered a more effective way to control the precision of calculations. But the epistemological discussion was, at the time, equally relevant. For the supporters of the analog computer, the digital model — which can only process information transformed and coded in binary — wouldn't be suitable to represent certain kinds of continuous variation that help determine brain functions. With analog machines, on the contrary, there would be few or no steps between natural objects and the work and structure of computation (cf. [39, 27]). The 1942–52 Macy Conferences in cybernetics helped to validate digital theory and logic as legitimate ways to think about the brain and the machine [39]. In particular, those conferences helped make the McCulloch-Pitts' digital model of the brain [36] a very influential paradigm. The descriptive strength of McCulloch-Pitts model led von Neumann, among others, to seek identities between the brain and specific kinds of electrical circuitry [27].

The roots of the theory of Analog Computation lie with Claude Shannon's so-called *General Purpose Analog Computer (GPAC)*.[15] This was defined as a mathematical model of an analog device, the Differential Analyzer, the fundamental principles of which were described by Lord Kelvin in 1876 (see [10]). The Differential Analyzer was developed at MIT under the supervision of Vannevar Bush and was indeed built in 1931, and rebuilt with important improvements in 1941. The Differential Analyzer input was the rotation of one or more drive shafts and its output was the rotation of one or more output shafts. The main units were gear boxes and mechanical friction wheel integrators, the latter invented by the Italian scientist Tito Gonella in 1825 ([10]). From the early 1940's, the differential analyzers at Manchester, Philadelphia, Boston, Oslo and Gothenburg, among others, were used to solve problems in engineering, atomic theory,

[15] In spite of being called "general", which distinguish it from special purpose analog computing devices, the GPAC is not a uniform model, in the sense of von Neumann.

astrophysics, and ballistics, until they were dismantled in the 1950s and 1960s following the advent of electronic analog computers and digital computers ([10, 29]). Shannon (in [54]) showed that the GPAC generates the *differentially algebraic functions*, which are unique solutions of polynomial differential equations with arbitrary real coefficients. This set of functions includes simple functions like the exponential and trigonometric functions as well as sums, products, and compositions of these, and solutions of differential equations formed from them. Pour-El, in [48], and Graça and Costa, in [21], made this proof rigorous.

The fact is that, although the GPAC model is physically realizable and is an analogue model of some part of the Universe, inputting and outputting real numbers, it does not compute more that the Turing machine, in the sense of Computable Analysis.

Cooper and Odifreddi say that *The association of incomputability with simple chaotic situations is not new. For instance, Georg Kreisel sketched in [34] a collision problem related to the* 3*-body problem as a possible source of incomputability.*

We think that these ideas are indeed conceived in a few theoretical experiences like in [15, 63], although they qualitatively require an unbounded amount of energy [16], and for this reason, not for theoretical reasons, they are not implementable. Returning to Kreisel, the pure mathematical model of Newtonian gravitation is probably capable of *encoding the halting problem of Turing machines.* This hint is given by Frank Tipler, too, in [60], based on constructions similar to Xia's 5-body system (in [63]), were we have two parallel binary systems and one further particle oscillating perpendicularly to both orbits. This particle suffers an infinite number of mechanical events in finite time (e.g., moving back and forth with increasing speed). Can we encode a universal Turing machine in the initial conditions? This is an unsolved mathematical problem. Thus, it may well be that the system of Newtonian mechanics together with the inverse square law is capable of non-Turing computations. The hypercomputational power that this system may have is not coded in any real number but *in its own dynamics.* How do we classify such a *Gedankenexperiment*?

In the *Billiard Ball Machine model*, proposed by Fredkin and Toffoli in [18], any computation is equivalent to the movement of the balls at a constant speed, except when they are reflected by the rigid walls or they collide (preserving global kinetic energy) with other balls, in which case they ricochet according to the standard Newtonian mechanics. The *Billiard Ball Machine* is Universal. Moreover, the faster the balls move, the faster a given computation will be completed. Newtonian physical systems that perform an infinite number of operations in a finite time are well known. Specifically, we just have to consider 4 point particles moving in a straight line under the action of their mutual gravity. Mather and McGee have shown in [35] that the masses and the initial data of the particles can be adjusted to result in the particles having infinite velocity in finite time. Gerver in [19] published a paper reporting on a model where, using

[16] Although the total amount of energy involved does not change.

5 point particles in the plane moving around a triangle, all particles could be sent to infinity in a finite time.

Can these systems encode hypercomputational sets? We aim at obtaining *either a positive or a negative answer to this question*, i.e., (a) either we will be able to prove that *initial conditions do exist* coding for a universal Turing machine, (b) or we are not able to prove such a lower bound but, we will prove that *encoding of input and output exists, together with adjustable parameters coding for finite control* such that we will have an abstract computer inspired by Newtonian gravitation theory. This result, together with a non-computable character of the n-body problem as shown in [58] *inter alia*, will turn to be a *strong basis to discuss a possible Church-Turing thesis' violation*. In fact, the non-computable character of the n-body problem is close to Pour-El and Richards' results [49], and not so close to a mechanical computer rooted in the structure of the inverse square law.

6 Routes to hypercomputation

Martin Davis published a paper called *The Myth of Hypercomputation* ([14]) which fights against work on hypercomputation in [56]; the criticism seems to related with the *dream* of building a hyper-machine. In [56] two paths are started: first, the physical construction of a hyper-machine (that culminates in Siegelmann's controversial claims in *Science* [57] that, agreeing with Davis, can be misinterpreted; and, second, the theoretical study of models of hypercomputation, were one searches for neural nets with weights of different types computing diverse computational classes: integer nets are equivalent to finite automata, rational nets are equivalent to Turing machines, polynomial time real number nets are equivalent to polynomial size Boolean circuits, and so on.

In the same way that differential equations in $\mathbb{R}^n$ are used to model Newtonian gravitation, nets with real number weights are worthy of investigation, since for decades engineers have been using them to model learning. In the latter case, philosophical thinking leans towards Davis's considerations. We don't believe in a physical constant L with the value of the halting number[17]. Even if there was, without some reason why it should have that value, we could not use it to make a hypercomputation. Because, if such a constant existed, then we could apply Thesis 4 and see objects around us performing hypercomputation having no tool to reproduce it. That would be the case of having hypercomputation as Alchemy, observe Cooper and Odifreddi in [11]:

> *To the average scientist, incomputability in Nature must appear as likely as 'action at a distance' must have seemed before the appearance of Newton's "Principia". One might expect expertise in the theory of incomputability — paralleling that of Alchemy in the seventeenth century — to predispose one to an acceptance of such radical new ideas.*

[17] Let $\ldots, (e_n, x_n), \ldots$ be an enumeration of programs and natural number inputs. The n-th digit of L is 0 or 1 according to whether or not the program e_n halts on the input x_n.

Alchemy ended and Chemistry started when the *scale* was introduced in Alchemy, a quite good interpretation due to Alexander Koyré. How do we measure hypercomputational behaviour? Suppose we do have a physical constant L having the value of the halting number. Then, if we measure this constant up to, let us say, n digits of precision, for sufficiently large n, and become aware that the program of code e_m halts for input x_m, *how could we verify it?* This would work as a call for observational refutation, it would be, like for Leverrier and Adams, a matter of faith, but in this case without Roemer's telescope in Berlin. Siegelmann's paper in Science looks like Leverrier and Adams trying to convince the scientific community that there is an alien out there. Why was the community not convinced? Well, in first place it seems that nothing in computing escapes mathematical explanation, like Uranus escaped to his computed orbit. But this is not obvious, since sometimes the scientific community do not react as Airy did. Do you remember about the scandalous trial in London in 1877? (We learned this from Michio Kaku's *Hyperspace* in [33].)

A psychic from the USA visited London and bent metal objects at a distance. He was arrested for fraud. Normally, this trial might have gone unnoticed. But eminent physicists came to his defense, claiming that his psychic feats actually proved that he could summon spirits living in the fourth dimension. Many of defenders were Nobel laureates to be. Johann Zollner, from the University of Leipzig came in his defence; so did William Crookes, J J Thompson and Lord Rayleigh. Why this difference of attitude: Airy's reaction to the letters of Leverrier and Adams, with mathematical calculations; Thompson and Crookes reaction to the possibility of psychokinesis working with Zollner?

Newton's *Sensorium Dei* was a metaphysical tool to understand a system of the world that without the intervention of God would collapse in his center of gravity. Leverrier and Adams made people believe again in Newtonian's system of Physics. Departures of computed lunar orbit against observations were explained by Euler. The world is ready for a Laplacian demon to remove God from physical space since Mr. de Laplace *ne besoin pas de cette hypothese* to understand the merry-go-round of the heavenly bodies in the sky. However, what Laplace didn't know is that, most probably, although this system is deterministic, it encodes its own unpredictability and its own incomputability. Probably, not even Laplace's demon has such a computer. In a letter, Cooper observed:

> *... it seems to me that recursion theorists have not until recently really understood or cared what their subject is about, and most still resist even thinking about it (and maybe the same can be said about complexity theorists...). Actually, Gandy was interesting to talk to — as is Martin Davis, of course. We think it is hard for people of my generation and before to adjust to the new fluidity of thinking (or maybe we should say the old fluidity of thinking of the inter-war years).*

The study of hypercomputation should be pursued with mathematics, as with any mathematical concept.

7 Final remarks

Science is a loose federation of different intellectual, experimental and material communities and cultures; the cultures are strong and are not confined to disciplines. We have reflected upon the task of combining our theoretical understanding of computation with that of the physical world. As working scientists, our view is limited to the problems of relating computability and complexity theory with mechanics (Newtonian, relativistic, ...). To us there are intriguing questions, observations, theorems and promising approaches.

However, the extraordinary development of the theory of computation since the 1930s has been based on its *mathematical abstractions from the physical world of machines and technologies.* These logical and algebraic abstractions have enabled the rise of digital computation, and have granted Computer Science its intellectual independence from electronics and physics. The mathematical maturity of our abstract theories of computation allow us to look at the physical foundations of computing in new ways. But it also makes our quest more controversial. Beautiful mature abstractions must be traded for clumsy new ill fitting physical notions.

All sorts of questions arise, from a fundamental curiosity about information processing in physical systems, and from a need to understand interfaces between algorithms and physical technologies (e.g., in new problems of quantum information processing, and in old problems of analogue computers). What is the physical basis of computation? Is there a theory of the physical process of making a computation?

But the task of unifying computability and mechanics involves wider issues. One discipline shapes the development of another. To take a example nearby, mathematical logic has had a profound influence of the development of programming languages. Priestley [47] has examined this process historically and to some extent methodologically using ideas of Pickering [45]. He has shown that there are exciting and rich philosophical phenomena to think about, involving concepts, theorems, practical problems, epistemics and sociology. Combining computability theory and mechanics is a tougher challenge. But it may some essential methodological structures, such as those of "bridging, transcription and filling" suggested by Pickering [45]. Likely it will have new ones, too.

Since the early seventies, we have seen the decline of the enthusiastic debate over what intellectual contribution has computability theory to offer science. The messy debate is back and there new people with a new agenda.

References

1. Edwin J. Beggs, José Félix Costa, Bruno Loff and John V. Tucker, Computational complexity with experiments as oracles, *Proceedings Royal Society Series A* 464: 2777–2801, 2007.
2. Edwin J. Beggs, José Félix Costa, Bruno Loff and John V. Tucker, The complexity of measurement in classical physics, in M. Agrawal, D. Du, Z. Duan, A. Li (eds),

Theory and Applications of Models of Computation, Lecture Notes in Computer Science, Volume 4978, Springer, 2008, 20–30.
3. Edwin J. Beggs, José Félix Costa and John V. Tucker. Oracles and advice as measurements, in C. S. Calude et al (eds), *Unconventional Computing 2008*, Lecture Notes in Computer Science, Volume 5204, Springer, 2008, 33–50.
4. Edwin J. Beggs and John 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.
5. Edwin J. Beggs and John V. Tucker. Embedding infinitely parallel computation in Newtonian kinematics, *Applied Mathematics and Computation* 178: 25–43, 2006.
6. Edwin J. Beggs and John V. Tucker. Can Newtonian systems, bounded in space, time, mass and energy compute all functions?, *Theoretical Computer Science* 371: 4–19, 2007.
7. Edwin J. Beggs and John V. Tucker. Experimental computation of real numbers by Newtonian machines, *Proceedings Royal Society Series A* 463: 1541–1561, 2007.
8. Edwin J. Beggs and John V. Tucker. Programming experimental procedures for Newtonian kinematic machines, A. Beckmann, C. Dimitracopoulos and B. Lowe (eds), *Computability in Europe, Athens, 2008*, Lecture Notes in Computer Science, Volume 5028, Springer, 2008, 52–66.
9. S. Barry Cooper. Clockwork or Turing universe? — remarks on causal determinism and computability, in S. B. Cooper and J. K. Truss (eds.), *Models and Computability*, London Mathematical Society Lecture Notes Series 259, Cambridge University Press, Cambridge, 1999, 63–116.
10. M. D. Bowles. U.S. technological enthusiasm and British technological skepticism in the age of the analog brain. *IEEE Annals of the History of Computing*, 18(4):5-15, 1996.
11. S. Barry Cooper and Piergiorgio Odifreddi. Incomputability in Nature. *Computability and Models*, in S. Barry Cooper and Sergei S. Goncharov (eds.), Kluwer Academic, 2003, 137–160.
12. Alain Connes, André Lichnerowicz and Marcel Paul Schützenberger. *Triangle of Thoughts*, American Mathematical Society, 2000.
13. David Deutsch. *The Fabric of Reality*, Penguin Books, 1997.
14. Martin Davis. The myth of hypercomputation, in Christof Teuscher (ed.), *Alan Turing: Life and Legacy of a Great Thinker*, Springer-Verlag, 2004, 195-212.
15. G. Etesi and I. Németi. Non-Turing computations via Malament-Hogarth space-times. *Int. J. Theor. Phys.*, 41:341-370, 2002.
16. Jens E. Fenstad. *General Recursion Theory*, Perspectives in Mathematical Logic, Springer-Verlag,1980.
17. Jens E. Fenstad. Computability theory: Structure or algorithms in *Reflections: A Collection of Essays in Honor of Solomon Feferman*, 2001.
18. E. Fredkin and T. Toffoli. Conservative Logic. *Int. J. Theor. Phys.*, 21:219-253, 1982.
19. Joseph L. Gerver. A Possible Model for a Singularity without Collisions in the Five Body Problem. *Journal of Differential Equations*, 52(1): 76-90, 1984.
20. Reuben L. Goodstein. On the restricted ordinal theorem. *Logic*, 9:33-41, 1944.
21. Daniel Graça and José Félix Costa. Analog computers and recursive functions over the reals. *Journal of Complexity*, 19(5): 644–664, 2003.
22. Morton Grosser. *The discovery of Neptune*, Dover, 1962, 1979.
23. Edward Harrison. *Cosmology, The Science of the Universe*, Cambridge University Press, Cambridge, 2000.

24. Stephen W. Hawking. Particle creation by black holes, *Commun. Math. Phys.* 43, 199, 1975.
25. Stephen W. Hawking. Information Loss in Black Holes, *Physics Reviews D* 72, 2005.
26. Gerald S. Hawkins. Stonehenge decoded. *Nature*, 202:1258, 1964.
27. S. J. Heims. *John von Neumann and Norbert Wiener: From Mathematics to the Technologies of Life and Death*, MIT Press, 1980.
28. Peter G. Hinman. *Recursion-theoretic hierarchies*, Perspectives in Mathematical Logic, Springer-Verlag,1978.
29. P. A. Holst. Svein Rosseland and the Oslo Analyser. *IEEE Annals of the History of Computing*, 18(4):16-26, 1996.
30. Fred Hoyle. *From Stonehenge to Modern Cosmology*, W. H. Freeman and Company, 1972.
31. Fred Hoyle. *On Stonehenge*, W. H. Freeman and Company, 1977.
32. Joe Killian and Hava T. Siegelmann. The dynamic universality of sigmoidal neural nets. *Information and Computation*, 128(1):48-56, 1986.
33. Michio Kaku. *Hyperspace, A scientific odyssey through the 10th dimension*, Oxford University Press, 1994.
34. Georg Kreisel. Church's Thesis: a kind of reducibility axiom for constructive mathematics. In A. Kino, J. Myhill, and R. E. Ṿesley (eds.), *Proceedings of the Summer Conference at Buffalo N. Y.*, North-Holland, Amsterdam, 1970, 121–150.
35. J. N. Mather and R. McGehee. Solutions of the collinear four body problem which become unbounded in finite time. *Dynamical systems theory and applications, Lecture Notes in Physics*, 38:573-597.
36. W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115-133, 1943.
37. C. Moore. Generalized shifts: unpredictability and undecidability in dynamical systems. *Nonlinearity*, 4:199-230, 1991.
38. C. A. Newham. *The Astronomical Significance of Stonehenge*, Coats and Parker Ltd, 2000 (first published in 1972).
39. J. M. Nyce. Nature's machine: mimesis, the analog computer and the rhetoric of technology. In R. Paton (ed), *Computing with Biological Metaphors*, 414-423, Chapman & Hall, 1994.
40. P. Odifreddi. *Classical Recursion Theory I*, Elsevier, 1992.
41. P. Odifreddi. *Classical Recursion Theory II*, Elsevier, 1999.
42. Roger Penrose. *The Emperor's New Mind*, Oxford University Press, 1989.
43. Roger Penrose. *Shadows of the Mind*, Oxford University Press, 1994.
44. Roger Penrose, Abmer Shimony, Nancy Cartwright, and Stephen Hawking. *The Large, the Small, and the Human Mind*, Cambridge University Press, 1997.
45. Andrew Pickering. *The Mangle of Practice*, Chicago University Press, 1995.
46. J. Pollack. *On Connectionist Models of Natural Language Processing*, PhD Thesis, Computer Science Dept., Univ. Illinois, Urbana, 1987.
47. M. Priestley. *Logic and the development of programming languages, 1930-1975*, PhD Thesis, University College London, 2008.
48. Marian B. Pour-El. Abstract computability and its relations to the general purpose analog computer. *Transactions Amer. Math. Soc.*, 199:1-28, 1974.
49. Marian B. Pour-El and I. Richards. Noncomputability in models of physical phenomena. *Int. J. Theor. Phys.*, 21(6/7):553-555, 1982.
50. Marian B. Pour-El and I. Richards. The wave equation with computable initial data such that its unique solution is not computable. *Advances in Mathematics*, 39:215-239, 1981.

51. Marian B. Pour-El and I. Richards. A computable ordinary differential equation which possesses no computable solution. *Amer. Math. Logic*, 17:61-90, 1979.
52. Pat Langley, Herbert Simon, Gary L. Bradshaw, and Jan M. Zytkow. *Scientific Discovery, Computational Explorations of the Creative Processes*, Anchor, 1997.
53. Hartley Rogers. *Theory of Recursive Functions and Effective Computation*, McGraw-Hill, 1968.
54. Claude Shannon. Mathematical theory of the differential analyzer. *J. Math. Phys., MIT*, 20:337-354, 1941.
55. Hava T. Siegelmann. *Neural Networks and Analog Computation: Beyond the Turing Limit*, Birkhäuser, 1999.
56. Hava T. Siegelmann and Eduardo Sontag. Analog computation via neural networks. *Theoretical Computer Science*, 131(2): 331-360, 1994.
57. Hava T. Siegelmann. Computation beyond the Turing limit. *Science*, 268(5210):545–548, 1995.
58. Warren D. Smith. Church's thesis meets the N-body problem. *Applied Mathematics and Computation*, 178(1): 154-183, 2006.
59. Viggo Stoltenberg-Hansen and John 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, 357–526.
60. Frank Tipler. *The Physics of Immortality: Modern Cosmology, God and the Ressurrection of the Dead*, Anchor, 1997.
61. John V. Tucker and Jeffrey 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: Logic Programming, Oxford University Press, 2000, 317 – 523.
62. John V. Tucker and Jeffrey I. Zucker. Abstract versus concrete computation on metric partial algebras, *ACM Transactions on Computational Logic*, 5(4): 611-668, 2004.
63. Jeff Xia. The existence of noncollision singularities in Newtonian systems. *Annals of Mathematics*, 135(3):411-468, 1992.
64. S. Wolfram. *A New kind of Science*, Wolfram Media, 2002.
65. D. Wolpert. A computationally universal field computer which is purely linear. *Technical Report* LA-UR-91-2937, Los Alamos National Laboratory, Los Alamos, NM, 1991.