Spacetime computing: towards algorithmic causal sets with special-relativistic properties

Tommaso Bolognesi*

Abstract *Spacetime computing* is undoubtedly one of the most ambitious and less explored forms of unconventional computing. Totally unconventional is the medium on which the computation is expected to take place – the elusive texture of physical spacetime – and unprecedentedly wide its scope, since the emergent properties of these computations are expected to ultimately reproduce *everything* we observe in nature.

First we discuss the distinguishing features of this peculiar form of unconventional computing, and survey a few pioneering approaches. Then we illustrate some novel ideas and experiments that attempt to establish stronger connections with advances in quantum gravity and the physics of spacetime. We discuss techniques for building *algorithmic causal sets* – our proposed deterministic counterpart of the *stochastic* structures adopted in the Causal Set programme for discrete spacetime modeling – and investigate, in particular, the extent to which they can reflect an essential feature of continuous spacetime: Lorentz invariance.

1 Introduction

Most approaches in the broad field of unconventional computing are tightly related to structures and functions that can be observed in the natural world (natural computing).

On one hand, smart solutions that have emerged during the multi-billion-year evolution of life on Earth provide valuable inspiration for developing novel algorithms meant to run on traditional computers, or novel computing paradigms and architectures to be implemented by ad-hoc, human-designed electronic hardware (bio-inspired computing).

* CNR/ISTI, 1, Via Moruzzi, Pisa, Italy, e-mail: t.bolognesi@isti.cnr.it

On the other hand, in the last two decades interest has grown for experiments in which the 'hardware' itself is provided by nature. For example, Rubel's Extended Analog Computer (1993) takes advantage of materials that are unfit for conventional computation, but still contribute to the machine functionality, just based on the laws of nature that they follow [26, 22]. As another example, in 1994 Adleman successfully used DNA molecules to solve a graph theoretic, combinatorial problem [2], thus starting the field of biomolecular computing. In [1], computing in 'reaction-diffusion' excitable media is shown to involve new computational paradigms, advanced non-standard architectures and novel materials. 'Natural hardware' may indeed take several forms, including chemical soups, cellular systems, bacteria, ant colonies, or various other biological substrata, as documented elsewhere in this volume. One of the main challenges that these unconventional, often massively parallel systems pose is how to harness their computing capabilities by adequate programming paradigms and techniques.

How about conceiving, as 'natural hardware', the elusive, ultimate fabric of the universe, namely spacetime? This bold question immediately raises two problems.

First, spacetime, the mathematical structure defined by the Einstein equations, is classically conceived as a *continuous* entity (a pseudo-riemannian manifold), while we usually associate the concept of computation to *discrete* entities, such as the state or the tape of a Turing machine. This objection is easily answered. On one hand, some of the above mentioned examples of natural computing prove that computation with continuous media is indeed definitely feasible. On the other hand, several recent theories of quantum gravity (e.g. Loop Quantum Gravity [29], Causal Dynamical Triangulations [3], Causal Sets [7]) adopt discrete models of spacetime that appear as perfectly adequate for supporting computation, as we shall soon illustrate.

The second difficulty is severe. While the biosphere offers several examples of brilliant information processing activities whose operation and purpose we now understand well, from the processing of genetic information as encoded in DNA to that of sensory data by various receptors and organs (say, echolocation in bats), we currently have no *direct* clues about information processing activities and functions that can be attributed to the discrete texture of spacetime, and no idea of what type of algorithm, if any, might be working at those ultra-low scales.

Thus, it would be inappropriate, at least as of today, to talk about *spacetime-inspired computing* in the same way as we talk about bio-inspired computing. Similarly, it would be extremely hazardous to imagine that physical spacetime might one day become the ultimate 'natural hardware' for human-controlled computations at the Plank scale (10^{-35} m); and not just because current experiments in molecular electronics and DNA-based computer circuits still take place at a much higher scale (10^{-9} m, the nano-scale) [16], but because, under the conjecture that human actions are themselves ultimately emergent from computations at ultra-low scales, the idea that we could manipulate those levels appears as highly questionable, if not a loopy logical impossibility.

Then, does it make sense to talk about *spacetime computing*?

The main purposes of this chapter are to provide some arguments in favour of a positive answer, and to illustrate a few past and present research and experimen-

tation activities which, in our opinion, can be reasonably collected under this bold name.

Two warnings are in order. First, the field is still fuzzily defined and might still take advantage from several research tracks, across several disciplines. But it is not our aim to be exhaustive in this respect; rather, we shall mainly focus on our own approach to the matter, presenting some of our recent results as well as original material. Second, we insist on the highly speculative nature of this research and on the fact that we still lack experimental evidence for punctual connections between computational processes and spacetime features.

In Section 2 we mention some solid, general reasons for believing that spacetime computes, although we are still very far from understanding how it does it.

In Section 3 we mention some early steps in this research area, mainly centered on the model of cellular automata. In Section 4 we consider other models of computation and their possible use for modeling a discrete, algorithmic, evolving space.

In Section 5 we move from a netwtonian view at space and time, intended as absolute, independent entities, to a relativistic, integrated view at spacetime. This leads us to deal with *causal sets*, or *causets*. After recalling a standard technique for building these discrete, *stochastic* models of spacetime, we discuss two general methods for obtaining discrete, algorithmic, *deterministic* versions of them, and introduce EH-causets ('Event-History') and PA-causets ('Permutation-Ant'). We also introduce an automaton, that we call 'Ring Ant', which produces both types of causet.

In Section 6 we identify a key feature of continuous spacetime that any discrete model of a computational spacetime must cope with: Lorentz invariance. We then introduce a relatively rough but practical indicator meant to assess the 'Lorentzianity' of the investigated spacetime models.

In Section 7 we illustrate a number of concrete examples of EH-causets derived from computations of the Ring Ant automaton, and show how they perform in terms of our Lorentzianity indicator. Section 8 is devoted to the illustration of PA-causets from computations of the same Ring Ant automaton. We find that these causets can perform better than EH-causets in terms of Lorentzianity.

In Section 9 we summarise our viewpoints about spacetime computing and its possible developments.

2 An algorithmic bottom layer

A quick argument in support of the idea that nature is fundamentally algorithmic is offered by the 'typing monkeys' metaphor.

In its original version, used, among others, by Borel and Eddington in the context of statistical mechanics, the metaphor suggests that an *infinite* sequence of random characters must include with probability 1, say, all sonnets by Shakespeare. However, if the sequence is created progressively, one can easily calculate that the expected time for the first complete sonnet to appear is incomparably longer than the age of our universe.

A modern version of the metaphor combines the use of old-fashioned typewriters with that of computer terminals. Assume, for example, that eight monkeys are typing at random on eight special typewriters with only three characters. The characters have the same shape - a square - but three different colors: white, grey, black. Each monkey fills a 30-line sheet, where each line contains 30-characters. The eight completed sheets are shown in the upper row of Figure 1; they appear statistically indistinguishable from one another, and no clue of order or structure can be detected in any of them: no shakespearean sonnet in sight. If the origin and unfolding of our universe were based on this kind of mechanism, we would get a totally random, structureless world, unable to support particles, atoms, not to mention stars and life.

When randomness and computation are combined, the picture changes. Each of the lower diagrams in Figure 1 shows the result of feeding the corresponding upper sheet to a different Turing machine selected at random from the set of $(4 \times 3 \times 3)^{3\times 3} = 101,559,956,668,416$ two-dimensional, 3-state, 3-color Turing machines, running for 80,000 steps. The potential 'universe' picture is now different: in spite of the randomness in the inputs and in the choice of the machines, some order emerges, manifested as an unbalance among the three colors, a tendency to distinguish between background and foreground 'objects', some alignment, and repeated patterns.

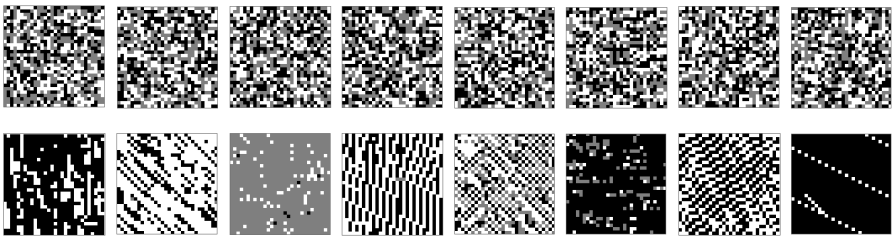

Fig. 1 Upper: random typing on eight paper sheets, using a three-character (three-color) typing machine. Lower: results of 80,000-step computations of eight randomly chosen 3-state, 3-color Turing machines, each running with the corresponding sheet above as input.

A more formal treatment of the typing monkey metaphor is possible via the notion of *algorithmic probability* of strings. Consider a string s of n bits. In the absence of any information on the origin of s, we usually assume that it was picked at random from the set of all 2^n bit strings of length n, thus we assign to it a probability 2^{-n}. If, however, we have reasons to believe that s was produced algorithmically, we can use the universal a priori probability, or Solomonoff-Levin algorithmic probability [14]:

$$m(s) = \sum_{p:U[p]=s} 1/2^{|p|}$$

where the summation involves all programs p of length $|p|$ such that a universal, prefix-free Turing machine terminates with output string s when running p. The use of this probability is legitimate regardless of the details of the algorithmic process

producing s, as guaranteed by Levin's coding theorem (see [14] for details and further references).

Consider the set S of all bit strings of length 5. In Figure 2-left we plot the individual algorithmic probabilities of the strings of S as estimated by the Online Algorithmic Complexity Calculator tool [2] [30]. This distribution is normalized - for each string s the plotted value is indeed $m(s)/\sum_{x\in S} m(x)$) - and compared with the uniform distribution of the 32 strings, each occurring with probability 1/32. Figure 2-right shows the 12 strings for which the (normalized) algorithmic probability is greater than or equal to 1/32.

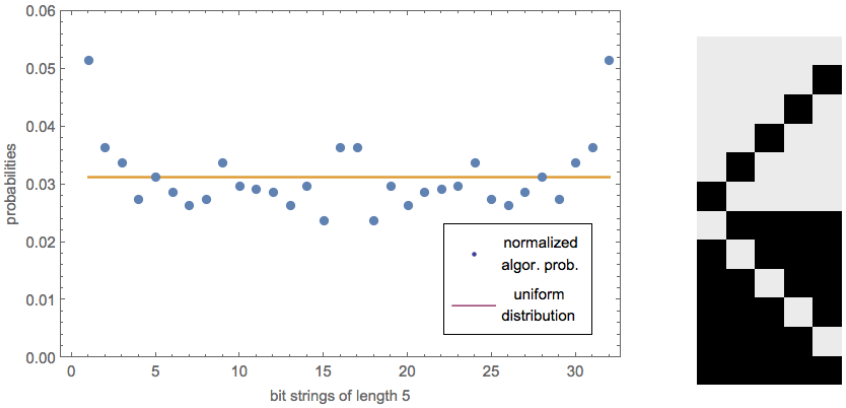

Fig. 2 Left: The algorithmic probabilities of the 32 bit strings of length 5. Right: The 12 strings with algorithmic probabilty greater than or equal to 1/32.

This simple example shows that algorithmic probability favours regular strings - in this case, those with all bits equal, with at most one exception. In an algorithmic universe order and disorder still coexist, but the former is given more chances to emerge.

3 Cellular automata: from Zuse to Wolfram

The idea of a physical space that computes is attributed to the german engineer Konrad Zuse (1910-1995), one of the fathers of the modern computer, although a related concept of a universe made of interacting elementary automaton-like entities - the *monads* - had been formulated much earlier by Gottfried Wilhelm von Leibnitz (1646-1716).

In his 1969 technical report *Rechnender Raum* (Calculating Space [40], re-edited in [39]) Zuse considered discretised versions of continuous fields in 1-D and 2-

[2] http://www.complexitycalculator.com/

D space, describing, in particular, molecule velocity and pressure in a gas-filled cylinder, and characterised their discrete dynamics in terms of higher order cellular automata rules directly derived from the differential equations of the continuous dynamics. Zuse's interest focused on what he baptized 'digital particles' (see [40], Sect. 3.1) - localised structures that emerge and possibly interact as the dynamics of the discretised field evolves. These 'particles' are abstract patterns that move in a spacetime diagram in which space extends horizontally and time flows downward; they should not be confused with the actual gas molecules, in the same way as an ocean wave is well distinguished from individual water molecules.

We have implemented some of the 1-D CA discussed in Section 3 of [40]; Figure 3 shows the associated emergent particles, that correspond to the computations carried out by Zuse - only manually, and to a rather limited depth - in Figures 15, 19, 20 and 26 of his paper.

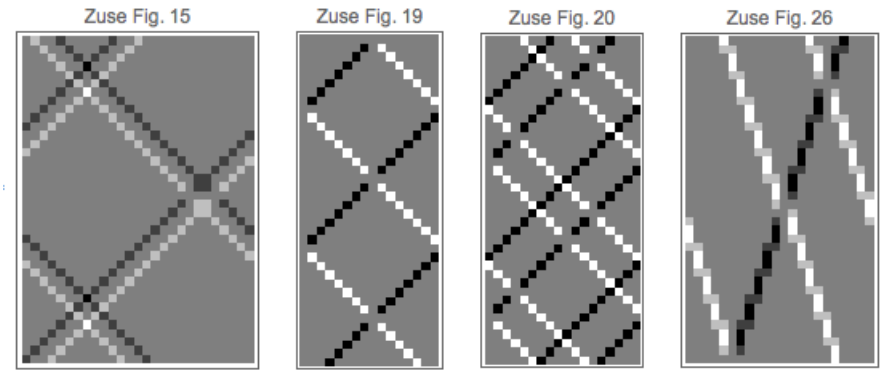

Fig. 3 Zuse's particles in higer-order cellular automata from discretised velocity-pressure 1-D fields. The diagrams show expansions of the computations illustrated in Figures 15, 19, 20, 26 of reference [40], using a circular topology for the CA cells. The integer values represented by the grey levels of the cells only refer to the *velocity* field; plots for the *pressure* field are similar.

Zuse observed that, as a consequence of a collision, particles may undergo a slight displacement of their initial trajectories, depending on their relative phase, and took this as a sign that '*a certain reaction process in particle interaction*' is possible. The phenomenon can be observed in the rightmost diagram of Figure 3.

The discovery of particles in CA computations and the intuition that their interactions resemble those of particle physics led Zuse to conjecture that CA might not just be useful discrete approximations of a supposedly continuous physical reality, but a perfect reflection of what reality ultimately is, and of how it operates: the physical universe as a giant Cellular Automaton.

Some limited experiments with particle emergence and interaction in 2D CA are also discussed by Zuse in [40], but it is only with Conway's Game of Life, divulged by Martin Gardner in 1970 [13] (the same year of the English translation of Zuse's *Rechnender Raum*) that the spectacular potentialities of these automata

became known to the wider public. *Blinkers*, *gliders*, *spaceships*, *pulsars* are just a few examples of the various localised structures that emerge in the Game of Life, and that are just elaborated instances of Zuse's digital particles. However, the large majority of the scientific community has constantly refused to attribute any deep theoretical meaning to these emergent phenomena, relegating them to the whimsical arena of recreational mathematics.

Very important contributions in support to the conjecture of a discrete, algorithmic spacetime have been given, after Zuse, by Ed Fredkin, with his work on Digital Philosophy. [3] Fredkin's Finite Nature assumption states that space and time are ultimately discrete, and that the number of possible states of any volume of spacetime is also finite [11]. Furthermore, information and computation assume, in Fredkin's work, a more fundamental status than matter, energy, and their transformations. As a consequence, Fredkin's interest, like Zuse's, focuses on cellular automata, in particular on second-order, Reversible Universal Cellular Automata (RUCA) and a model called SALT [21], in which two distinct components are arranged in a regular 3D lattice resembling *NaCl* crystals. According to Fredkin, RUCA reflect exactly and efficiently CPT symmetry, a fundamental property of physical laws, and can be regarded as the fundamental, information-processing mechanism at the roots of the physical universe.

Important theoretical and experimental developments in the field of cellular automata, reversible computation and their applications to the modelling of physical processes are due, among others, to Toffoli and Margolous [33, 18, 34, 35, 32] (see [8] for a survey).

Perhaps the strongest impulse to the investigation and divulgation of the computational universe conjecture is due, in more recent years, to Stephen Wolfram. With his monumental and controversial volume 'A New Kind of Science' [38], appeared in 2002, Wolfram has somehow reversed the approach by Zuse and Fredkin: rather than deriving a specific model of computation from the consideration of specific physical systems, he has undertaken a rather systematic, abstract exploration of the wide space of models of computation - from cellular automata to Turing machines, from register machines to string and graph rewrite systems, and more - with the objective to classify their emergent behaviours.

When a model of computation is sufficiently simple, it is possible to exhaustively explore all its instances: this is the case for Elementary Cellular Automata (ECA). An ECA is a linear arrangement of potentially infinite binary cells, typically represented as black (for '1') and white (for '0') squares, representing discrete, 1-D space. Time is also discrete, and, for any given ECA, the binary value $c_i(t+1)$ of cell c_i at time $t+1$ depends on the binary values at time t of c_i itself and of its immediate neighbors, namely $c_{i-1}(t), c_i(t), c_{i+1}(t)$. This dependency is expressed by a boolean function of three boolean arguments; since there are 256 such functions, we have 256 distinct ECAs, whose behaviours have been thoroughly studied by Wolfram, starting both from simple and from random initial configurations of the cells.

[3] http://www.digitalphilosophy.org/

ECAs are the simplest form of CA, considerably simpler than the higher-order CAs investigated by Zuse and Fredkin, and yet they turn out to be very effective in illustrating the creative power of 'spontaneous' computations. Emergent features observed in CA include self-similarity, pseudo-randomness (deterministic chaos), and digital particles, as illustrated in Figure 4.

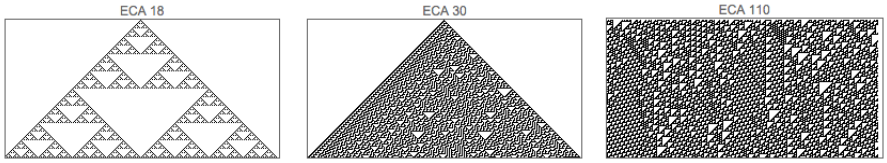

Fig. 4 Three ECAs illustrating self-simjilarity, pseudo-randomness, digital particles.

Interestingly, it is enough to move to a slightly more complex class of CAs - those that operate on ternary rather than binary cells - for finding a single CA in which the three mentioned features coexist, even when starting from an elementary initial condition. The automaton, discovered by Remko Siemerink, [4] is illustrated in Figure 5.

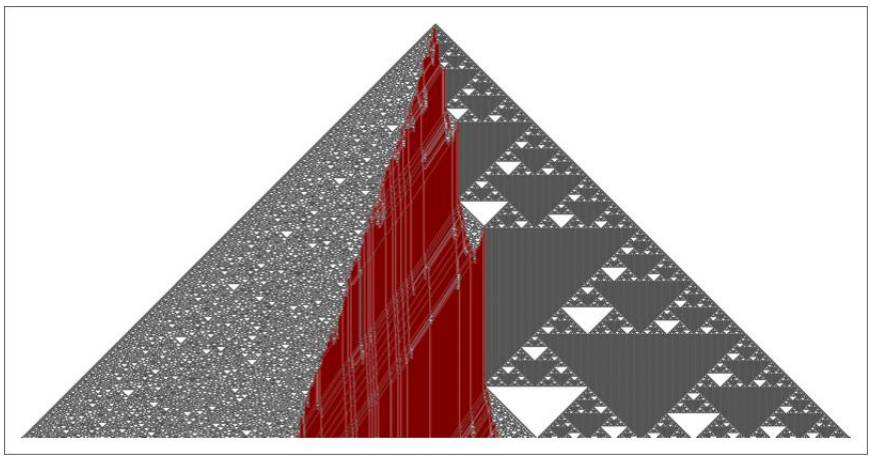

Fig. 5 A 3-color cellular automaton with elementary initial condition, exhibiting pseudo-randomness, particle trajectories and selfsimilarity.

The properties illustrated in Figures 4 and 5 appear to reflect some of the most fundamental, recurring patterns of Nature. Another crucial ingredient for the existence of what we regard as the most complex layer of our universe - the biosphere - is of course self-replication. In fact, CAs have been originally conceived by John

[4] http://www.wolframscience.com/summerschool/2009/alumni/siemerink.html

von Neumann, after a suggestion by Stanislaw Ulam, exactly for studying and simulating this phenomenon. The pervasive presence of these features both in Nature and in the abstract space of CAs provides further evidence that the universe is fundamentally algorithmic.

Beside this ability to replicate key aspects of Nature, two of the principles behind the operation of CAs appear, apriori, quite attractive to physicists: *uniformity* – the same boolean function is used for all cells, and *locality* – the cell transition function only involves neighbouring cells. Their relevance for applications to physics is rather obvious: they reflect the views that physical laws should not change with space or time, and that effects are transmitted by contact. However, a third principle behind CAs, namely *parallelism* - the synchronous operation of the unbounded set of cells - is much less appealing, since it hints at the idea of a global clock.

This reserve on parallelism, the fact that all Turing-universal models are equivalent, at least from the point of view of their computing power, and the observation that interesting emergent properties are found, more or less evident, also outside the realm of CAs, have provided momentum for the investigation of the wider space of non parallel, Turing-universal models.

4 Beyond CA: ant-based models of a dynamic space

The diagram of an elementary cellular automaton is a two-dimensional array of bits in which space extends horizontally and time flows vertically. Each row is a snapshot of space at a given time. Beside its 0/1 state, each cell in the array is characterised by the unique, absolute values of its space and time coordinates. Indeed, ECAs reflect exactly a newtonian concept of space and time.

But there are several other ways to conceive an algorithmic, netwtonian, evolving space. By dropping the requirement of parallel operation, one is led to consider models in which the data structure manipulated by the computation, for example a 1-D array of binary cells, is modified locally rather than globally, for example one cell at each step.

4.1 Turing machines

Elementary Turing machines (TM) are the most obvious example of this computational paradigm, that we shall call *ant-based*. The 'ant', in the case of a TM, is the *control head* - the finite state, read/write unit that reads the current cell and reacts, depending also on its own state, by writing a new a bit in the cell and moving one step left or right, as established by the state transition table.

A TM binary tape is analogous to the row of an ECA array, and can be interpreted as a snapshot of space. By packing the successive tape configurations of a TM we obtain a 2-D array conceptually equivalent to an ECA diagram - a discrete space-

time. However, in the TM case the change from one row to the next is confined to one location, all the rest of the tape being unaffected, so that in general one can trace the ant motion across spacetime: a TM-based toy universe evolves much more slowly than an ECA universe.

4.2 Turmites

Similar to CAs, TMs admit higher-dimensional variants. For example, in two-dimensional Turing machines, called 'turmites' [15], the control head moves on a two dimensional square array. The most famous turmite is *Langton ant* [37], a machine that gets trapped in a rather complex periodic behavior (a 'highway') only after over a thousand, pseudo-random initial steps, but many more behaviors are possible in this model, as illustrated in [15].

4.3 Network mobile automata

The cell array of a Turing machine is a rigid support similar to an immutable, infinite newtonian spatial background expected to exist before starting the computation. As an alternative, and being inspired by the Big-Bang concept, it is attractive to investigate algorithmic models in which space is not a predefined infinite rigid structure, but an emergent product itself of the computation.

Wolfram [38] has widely explored the idea of a graph-based computational Big-Bang, one in which space is modeled by a graph $G(N,E)$ - a set N of nodes interconnected, two-by-two, by a set E of edges. A *Network Mobile Automaton* is somewhat analogous to a Turing Machine, except that the ant does not move on a tape but on a graph, and modifies the latter locally, step by step. Space starts as a tiny graph, and evolves into a gigantic network of nodes due to the graph-rewrite rules applied at each step. These rules change the local topology of the graph and, most importantly, may introduce new nodes and edges: space evolves and grows with the computation. Trivalent (or 'cubic') graphs - ones in which each node has exactly three neighbours - are sufficient for 'implementing' spaces of any dimensionality, including 3D space ([38], Ch. 9).

In [4] we have explored variants of Network Mobile Automata for creating planar trivalent networks by using only two simple rewrite rules, namely the 2D Pachner rules, sometimes called *Expand-Contract* and *Exchange*. These rules have found application also in Loop Quantum Gravity [29], where they are used for the dynamics of spin networks. In spite of the planarity restriction, our experiments have yielded a wide variety of interesting regular 1-D ('polymer-like') and 2D networks, as well as oscillating rings, semi-regular hexagonal grids, up to totally irregular patterns.

4.4 A multi-threaded universe

A typical objection against the ant-based computational universe conjecture is that we perceive the world as a concurrent, multi-threaded system, not as the single-threaded one that the ant-based view seems to imply.

Wolfram [38] has an appealing argument to dismiss this objection. In essence, viewing a sentient being itself as a bounded region R of the toy universe, and associating R's perception to a change of its internal configuration/state, an act of perception will only occur when the ant visits R. But during the inter-visit intervals the ant has the opportunity to modify many of the other regions, thus creating R's subjective illusion of multiple parallel changes. The sequential ant behaviour is detected by the external observer, not by the internal one.

5 From absolute space to relativistic spacetime: algorithmic causets

Although with graph oriented models we have gotten rid of the cumbersome, regular and rigid spatial background of CAs and TMs, we have still been reasoning in newtonian terms, dealing with a sequence, in *absolute time*, of snapshots of *absolute space*. But Minkowski and Einstein have taught us that space and time, taken separately, have no absolute value, since different inertial observers, say Bob and Alice, register different spatial distances and different time intervals between the same two events: in particular, Bob may perceive them as simultaneous when Alice does not, and vice versa. The only absolute distance between events - one on which all inertial observers agree - is *spacetime distance*, i.e. Lorentz distance.

To most physicists, no spacetime model should ignore the lesson of Special Relativity. Thus, let us briefly summarize some basic features of this integrated view at space and time, and the associated notion of causality.

5.1 Lorentz distance and lightcones

Let us consider Minkowski spacetime $M^{(1,3)}$, which describes the simplest form of a matter-free, flat universe. $M^{(1,3)}$ can be understood as Euclidean 4-D space E^4, with spatial dimensions w, x, y, z, in which one of the dimensions, say w, is interpreted as a time dimension t, and where the Euclidean distance is replaced by the Lorentz distance. While the Euclidean distance between two points $p(w,x,y,z)$ and $p'(w',x',y',z')$ is given by $d^2(p,p') = (w-w')^2+(x-x')^2+(y-y')^2+(z-z')^2$, their Lorentz distance is expressed by:

$$L^2(p,p') = +(t-t')^2-(x-x')^2-(y-y')^2-(z-z')^2.$$

Due to its + - - - signature, the squared Lorentz distance can be *positive*, *null*, or *negative*: correspondingly, the two points are said to be in *time-like*, *light-like*, or *space-like* relation.

The *lightcone* of point p is the set of all points q in light-like relation with p, including p itself. The *future* (resp. *past*) *lightcone* of p is the subset of the lightcone whose points have time coordinate larger (resp. smaller) than that of p. [5] A physical process taking place at p can only influence the processes taking place at points on or inside the future lightcone of p: causality is limited by the speed of light.

The Lorentz distance immediately induces a *partial order* '$\prec$' among spacetime points: $p \prec q$ whenever $L^2(p,q) \geq 0$ and q is on or inside the future lightcone of p. Given a set of points S with partial order '$\prec$', we can define intervals between points: the *order interval* $I[s,t]$ between points s and t is the set of points $\{p \in S | s \prec p \wedge p \prec t\}$, which includes s and t.

5.2 Stochastic causal sets from 'sprinkling'

If the fundamental structure of a *continuous* spacetime manifold is its causal, light-cone structure, it seems natural to conceive *discrete* spacetime as a partially ordered set of events, or a *causal set* ('*causet*') [7].

A *causet* is a set with a partial order relation. As such, it can be represented as a *directed acyclic graph* (DAG) $C(N,E)$, where N is the set of nodes and E is the set of edges that define the partial order among nodes. We shall assume causets to be *transitively reduced*, in which case their edges are called *links*. (The transitive reduction of a DAG G is the unique smallest graph that has the same transitive closure as G.)

The *sprinkling technique* is a stochastic method that allows one to directly derive such a DAG from a continuous, Lorentzian spacetime - one in which the Lorentz metric is defined. Consider 2-D Minkowski space $M^{(1,1)}$, the simplest toy model of a Lorentzian spacetime, with one time dimension (vertical) and one space dimension (horizontal). In Figure 6-left we show an interval of $M^{(1,1)}$, between the points labelled 0 and 9 - *source* and *sink* - and a set S of 8 points uniformly sprinkled in it. We also show all directed edges that connect point-pairs that are in time-like relation (the probability of finding two points in light-like relation is zero). In Figure 6-right we show, upside-down, the corresponding causet $C(S,E)$, obtained by taking the transitive reduction of the 'raw' graph on the left and disregarding node coordinate information.

In our opinion, one of the attractive efforts in the field of spacetime computing, that we begin to illustrate in the next subsection, is to reverse the above logic: under

[5] Note that the difference between the time coordinates of two points p and q that are in time-like or light-like relation depends on the frame of reference, and is affected by the Lorentz transformation between inertial frames, but only in its absolute value, not in its sign. If the points are in space-like relation, on the contrary, the sign itself may change, so that different observers may disagree on the time ordering of the events.

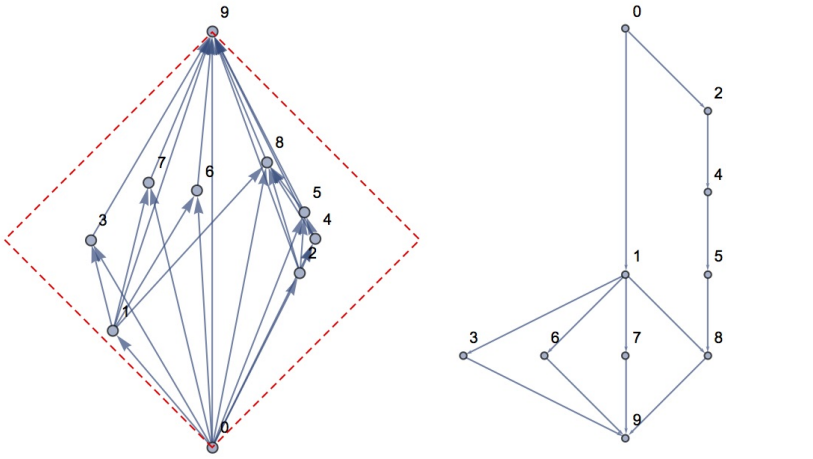

Fig. 6 Left: Sprinkling 8 points in an interval (red dotted lines) between two fixed points, labelled 0 and 9, of 2-D Minkowski space $M^{(1,1)}$. Right: Deriving a transitively reduced causet from the points, based on their mutual Lorentz distances; causal links here flow downward.

the assumption of a fundamentally discrete and algorithmic universe, the plan is to directly build a discrete, algorithmic model - say, an n-node causet - from scratch, without resorting to an underlying continuum, while expecting the familiar properties manifested by continuum models - e.g. dimension, curvature, Lorentz invariance - to emerge as $n \to \infty$. [6]

Note that this asymptotic perspective implies that those familiar properties might emerge only after some coarse-graining of the causet, e.g. by focusing only on a fraction of the available points. This view would leave room for 'wild' behaviours of the causet at its smallest scales; for example, the causet *as is* might turn out not to be faithfully embeddable in any manifold: "physics near the Plank scale need not be continuum-like" [25].

5.3 Algorithmic EH-causets ('Event-History')

In light of the variety of interesting emergent properties offered by the models of computation mentioned in the previous sections, and of the importance that we have come to attribute to causal sets for correct, post-newtonian spacetime modeling, we are interested in the possibility to *directly derive causal sets from the computations of those simple models*. The ultimate, ambitious goal of this approach would be

[6] Most physicists would favour the inclusion of a quantum-mechanical perspective to this effort, trying to handle collections of causets rather than individual instances, in the spirit of 'sum over histories', or 'path integrals'. We do not cover this aspect here, except for a few short comments in the conclusive section.

not only to obtain discrete spacetime models that exhibit the right properties of dimensionality, curvature, Lorentz invariance, but that also emergent properties such as fractals, or periodic localised structures analogous to CA 'particles', an effect that we certainly cannot expect from a purely stochastic approach! In essence, the plan is to merge two well distinct research efforts that are referred to as the 'Computational Universe Conjecture' and the 'Causal Set Programme'.

Can we recast the computations of, say, Turing Machines or Network Mobile Automata in terms of causal sets? Bizarre as the question may sound, the answer is definitely positive.

The idea of conceiving the steps of a computation as a set of causally related, partially ordered events was first explored in [12], but the purpose there was to characterise computable functions. It was Wolfram [38] who first proposed to view these graphs as instances of spacetime.

A general method for deriving a DAG from a sequential computation is easily defined [5], as long as we can represent the computation $\mathscr{C}$ as a sequence of steps that create, destroy, write and read state variables:

$$\mathscr{C} = ((-, W_0), (R_1, W_1), \ldots, (R_n, W_n), \ldots)$$

Each event (R_i, W_i) in the sequence reads the elements of some set R_i of state variables, and writes those of some set W_i. We conceive state variables, and associated read and write operations, in a rather broad sense: a state variable is not only a slot in some memory support, a cell on a tape, the state of a Turing Machine control unit; it can also be a node or an edge in a trivalent graph or, generally, any atomic component of some complex data structure. Then, read, write, creation or elimination operations are just manipulations of these items. The initial configuration of the system is created - written - by event 0, which does not read anything.

Once the above sequence $\mathscr{C}$ of computation steps is provided, a causet $C(N,E)$ is readily obtained: nodes $N = \{1, 2, \ldots n, \ldots\}$ are in one-to-one correspondence with the events, and an edge $i \rightarrow j$ is created in E whenever $W_i \cap R_j \neq \emptyset$: this means that some variable has been written (or created) by event i and read (or destroyed) by event j. State variables play the role of *causality mediators* between events, and organise events in a partial order which describes the history of the computation. For this reason we shall sometimes refer to these DAGs as *EH-causets*, for 'Event History'. (Note that in this model the actual values assumed by the state variables play no role.)

Before showing to the reader some examples of application of this general technique, we introduce a second approach for building algorithmic causets, more directly related to the stochastic, sprinkling procedure of Subsection 5.2. For doing this, it is convenient to represent sprinklings by permutations.

5.4 Correspondence between sprinklings and permutations

There exists a tight correspondence between a k-point sprinkling S in 2-D Minkowski space and a particular permutation π of the first k positive integers. All the information necessary for deriving a causet from S, via the Lorentz distance, can be compactly recorded in a permutation π.

To see this, we must view S under a different angle, literally. Without loss of generality, assume that the sprinkling - e.g. the one of Figure 6-left - has taken place in the interval of Minkowski space $M^{(1,1)}$ between points $s(0,0)$ and $t(0,\sqrt{2})$, so that the interval, identified by the red dotted lines, is indeed a square with sides of length 1.

Let us now rotate by $-\pi/4$ this 'diamond' and its content around the origin (0, 0), so that the points fall in the unit box between (0,0) and (1,1). Let now S_x be the list of the points, after rotation, sorted by ascending x-coordinate, and S_y be the list of the same points sorted by ascending y-coordinate. Finally, let $\pi = (\pi_1, \pi_2, ..., \pi_k)$ be the list of integers where π_i indicates the rank in S_y of the i-th point in S_x . Clearly π is a permutation of the first k positive integers. The patient reader may check that the permutation derived from the set S of 8 sprinkled points in Figure 6-left is (4, 8, 7, 6, 1, 2, 3, 5): the first element is 4 because the first point in S_x, labeled '1' in Figure 6-left, is the 4th point in S_y; the second element is 8 because the second point in S_x, labeled '3' in Figure 6-left, is the 8th point in S_y; and so on.

Consider now the following procedure for deriving a causet from a *generic* permutation π.

Permutation-based causet construction procedure

A causet $C_\pi(\Pi, F)$ is derived from a permutation π of the first k positive integers as follows:

- **Nodes**
$$\Pi = \{(i, \pi_i) | i = 1, 2 \ldots k\} \cup \{(0,0), (k+1, k+1)\}$$
The $k+2$ nodes are points with integer-valued coordinates. Of course these coordinates are not part of the causet structure: they are only used for defining the causet links.
- **Links**
A link $(i,h) \rightarrow (j,k)$ from node (i,h) to node (j,k) is created in F if and only if $i < j$, $h < k$, and the rectangle identified by the two points (as lower left and upper right vertices, respectively) is empty, i.e. no other node is found inside it. $\diamond$

Nodes labeled $(0,0)$ and $(k+1, k+1)$ are the source and the sink of the causet. Note that graph $C_\pi(\Pi, F)$ is acyclic and transitively reduced by construction.

Going back to our original sprinkling S and to the permutation π derived from it, we can now establish (without proof) the following simple fact.

Fact 1 *The causet $C(S,E)$ obtained directly from sprinkling S and the causet $C_\pi(\Pi,F)$ obtained from permutation π (in turn derived from S) are isomorphic. More precisely, there is a link $p_i \rightarrow p_j$ in E if and only if there is a link $(i,\pi_i) \rightarrow (j,\pi_j)$ in F.*

Additionally, it is easy to see that the process of obtaining causets from k-point random sprinklings in $M^{(1,1)}$ intervals, via the Lorentz distance, is *statistically equivalent* to that of obtaining causets from random permutations of the first k positive integers, via the permutation-based causet construction procedure above. The ultimate reason is that the x and y coordinates of the rotated sprinkled points are uniform and independent random variables.

In conclusion, one can safely and conveniently build sprinkled, 2-D interval causets by just using random permutations.

Let us now consider algorithmic techniques for generating and manipulating permutations, and therefore causets.

5.5 Algorithmic PA-causets ('Permutation Ant')

In a *Permutation Ant Automaton* [6] the 'ant' moves up and down a finite, one-dimensional array of cells $A(c_1,c_2,\ldots,c_n)$ by short steps or jumps, while performing operations such as reading, writing, swapping, creating or deleting cells. No matter how the array evolves, at any stage it stores a permutation $\pi = (\pi_1,\pi_2,\ldots,\pi_n)$ of the first n integers, one integer in each cell. Based on this permutation, at any step we can derive a *PA-causet* , for 'Permutation Ant' by the permutation-based construction procedure of the previous subsection.

The model can be enriched in various ways (see [6]). For example, the ant may be stateless, or follow a finite-state behaviour, like in Turing Machines. Furthermore, array cells may store bits, beside the elements of the permutation. The ant may read and write the bits in its neighborhood, as in CAs, and may react depending on these bits and on its current state (if any); it may move by unit steps, or may jump to other locations of array A, as addressed by the π_i of the current cell.

In the sequel we describe an algorithm that combines some of these features in a way that allows us to derive from the same computation *two distinct causets*: an EH-causet and a PA-causet. The advantage is to use a single model for illustrating the two causet construction mechanisms and for exploring two causet spaces.

5.6 Ring Ant

This model borrows and combines ideas from automata introduced in [6] and in [5]. The support of the computation is a circular tape with a distinguished first cell c_1. Each cell c_i stores a pair (b_i,π_i), where b_i is a bit and π_i is a positive integer

representing an element of the stored permutation π, which must be read starting from c_1.

The ant has 4 possible states $\{s_1, s_2, s_3, s_4\}$; its behavior depends on the current detected *situation* and is manifested by a set of possible *reactions*.

Situation. This is coded by 5 bits: 3 bits (b_{i-1}, b_i, b_{i+1}) are those found in the cell c_i where the ant is currently positioned, and in the two neighboring cells, with indices treated circularly. The two remaining bits code the state of the ant (i.e. $s_1 \leftrightarrow (0,0)$, $s_2 \leftrightarrow (0,1)$, $s_3 \leftrightarrow (1,0)$, $s_4 \leftrightarrow (1,1)$). Thus, there are 32 possible situations.

Reaction. This is also expressed by 5 bits $(b_1 \dots b_5)$ whose interpretation is as follows.

- b_1- insert a new cell $(b_1, n+1)$ at the current location, where n is the current tape length, so that $(n+1)$ is a fresh new element of the permutation.
- (b_2, b_3) identify 4 cases:
  - $(0,0)$: ant moves left one step;
  - $(0,1)$: ant moves right one step;
  - $(1,0)$: ant moves left by π_i steps, where c_i is the current cell;
  - $(1,1)$: ant moves right by π_i steps, where c_i is the current cell;
- (b_4, b_5) identify the new state of the ant.

The last two of the four ant moves are reminiscent of the GOTO command of various programming languages. Since for each of the 32 situations we can associate 1 out of 32 reactions, we can conceive 32^{32} distinct instances of the automaton - a huge space that we can only explore by random samplings or by a genetic algorithm approach. [7]

In our experiments we have started each computation with a two-cell circular tape storing permutation $\pi = (1,2)$, with bits set to 0, and with the ant in state s_1, and we have run the automaton for an arbitrary number of steps - typically a few thousands. Note that at each step the circular tape and the stored permutation grow by one unit.

The derivation of an EH-causet from a computation of this automaton, along the lines described in subsection 5.3, needs some clarification. We let the bits stored in the tape cells play the role of causality mediators among the events that write and read them; the permutation elements can't be used for this purpose, because they are never read. Since each event reads the bits of three cells, we obtain a 'raw' causet in which nodes - representing events - have in-degree 3. The out-degree of an event equals the number of times subsequent events have read the cell created by that event.

On the other hand, deriving a PA-causet from the final permutation is straightforward using the procedure described in subsection 5.4. Note that the final permutation, read from cell c_1, always starts with '1'. This means that the derived PA-causet

[7] Some of the computations presented here have been indeed selected by a genetic algorithm, using appropriate fitness functions. These aspects are not discussed here.

has always the node with integer coordinates $(1,1)$ as the root; this is also the only source node of the graph.

We ask again the reader to be patient: before providing examples of causets obtained by the above techniques we need to discuss an important criterion that we shall use for their assessment.

6 Causal sets and Lorentz invariance

We have mentioned before the requirement for discrete, algorithmic spacetime models to reflect as much as possible the features of continuous, physical spacetime. For example, we might attempt to reproduce the standard 4 dimensions of relativistic spacetime, or even aim at 10, 11 or 26 dimensions, as suggested by more recent theories such as String, Superstring and M-theory. In fact, one may expect spacetime dimensionality to depend on the observation scale. For example, two recent quantum gravity theories - Causal Dynamical Triangulations [3] and Quantum Einstein Gravity [24] - agree in picturing a four-dimensional universe which turns two-dimensional when observed at ultra low scales. A few techniques are available for estimating causet dimensionality [20, 23], and some applications to algorithmic causets have been investigated in [5].

However, the fundamental property of continuous spacetime on which we want to focus now is *Lorentz invariance*.

A physical entity, e.g. the distance between two spacetime events, or a physical law, e.g. the Maxwell equations, is Lorentz invariant if it does not change its value, or its form, under the Lorentz transformation, which describes the *change of spacetime coordinates* in passing from one inertial (non accelerating) frame of reference to another. According to the principle of relativity, the laws of physics must be invariant for all inertial frames of reference. It was the consideration of the Maxwell equations that led Einstein to abandon the galilean principle of relativity and to adopt the one based on the Lorentz transformation, since only the latter can account for the constant speed of light that comes with the Maxwell equations.

The issue of Lorentz invariance for causets is delicate, and we shall try to illustrate its essence without introducing excessive technicalities.

To begin with, a causet $C(N,E)$ is an abstract graph structure *without coordinates*, thus, applying a Lorentz transformation to it appears totally meaningless. For the idea to make sense we must still refer to an embedding of C in a manifold M, where each node has its own coordinates. The embedding must be consistent with the partial order expressed by the links E, that is, for any edge $p \rightarrow q$ in the transitive closure of E, p and q must be causally related also in M, via the Lorentz metric, and vice versa.

Embeddability is one of the hard problems studied in the field of causal sets. A generic DAG $C(N,E)$ with a sufficient number of nodes is very unlikely to be embeddable in a manifold. However, once C is embedded in some manifold M, it can be embedded in any other manifold M' obtained by applying a Lorentz transformation

to M: this is because the transformation drags, with M, the nodes N embedded in it, and does so while preserving their mutual Lorentz distances, so that consistency between the partial orders - in the discrete and in the continuum - is preserved. From the above remarks one might be tempted to conclude that as soon as a causet is embeddable in a Lorentzian manifold, it is also Lorentz invariant. But there is a complication.

A key point of Lorentz invariance is that all reference frames must appear equivalent - none must be singled out as preferred. If we decide to look at the pure causal structure of the graph, without coordinates, then we have no means to discriminate among Lorentz-interrelated reference frames: they all correspond to one and the same graph. But if we allow to access the coordinate information coming with the embeddings, then it turns out that for some graphs all embeddings are equivalent, while for other graphs they are not. In the latter case, a preferred frame may emerge.

The typical examples used for illustrating these two cases are a sprinkled 2-D causet and a regular square grid. A remarkable feature of a uniform Poisson distribution of points in a region of 2-D Minkowski space $M^{(1,1)}$ is that a Lorentz transformation will change the overall shape of the cloud of points, but will leave the local picture unchanged: an observer sitting on one of the points will notice no change in its neighbourhood - in statistical sense. No frame is special.

This is not the case for a regular, directed square grid embedded in $M^{(1,1)}$, one where each node has two incoming and two outgoing links at +45 or -45 degrees from the vertical, time axis - links that partition the plane into square tiles. Here a Lorentz transformation does induce local changes: as the frame speed increases, points get packed with increasing density along lines that get increasingly separated, breaking the symmetry of the original grid, and allowing us to single out the latter as the preferred, rest frame.

The notion of 'causet Lorentz invariance' or 'causet Lorentzianity' that uses embeddings and coordinates has, in our opinion, strong and weak aspects. It is strong because the original notion of Lorentz invariance does rest, crucially, upon that of reference frame - an embedding manifold. It is weak because causets were not conceived to inhabit a continuous background spacetime, but to be themselves the only existing, discrete spacetime - a concept known as *background independence*. Their 'Lorentzianity' should be directly manifested by their features - their nodes and edges - without need to refer to manifolds and their coordinate systems. Of course, when following this latter path we should be ready to give up the rich tool set that comes with continuous manifolds. For example, we immediately get into trouble when trying to define 'reference frame' purely in terms of DAGs.

What we show next is an alternative and simplified approach to causet Lorentzianity that avoids embeddings and reference frames, and only looks at DAG properties. This will be applied to algorithmic causets in the next section.

6.1 The interplay of longest and shortest paths

A rather counterintuitive feature of Minkowski space and its Lorentz distance L is the *reversed triangular inequality* - the fact that, given three points p, q and x, with q in the future lightcone of p and x in the order interval $I[p,q]$ (see subsection 5.1), we have:

$$L(p,q) \geq L(p,x) + L(x,q).$$

The Lorentz distance measures the time elapsed along a trajectory between two points, and the above inequality reflects the well known twin paradox of Special Relativity, by which the travelling twin, going through point x, ages more slowly than his sedentary brother. In fact, beside the longest path - a straight line from p to q - there is an infinite number of alternative paths that register shorter, or even *much* shorter time delays between those points, up to the limit case of two segments of light rays forming a $\pi/2$ angle at x, corresponding to a null time delay.

In [6] we have proposed a notion of causet Lorentzianity that takes into account, in a graph-oriented setting, this peculiar, wide range of path lenghts that characterises Lorentzian manifolds. The technique consists in collecting and aggregating statistical data about path lengths in the causet C under investigation. Given interval causet $C[s,t]$, we compute the lengths $lpl_s(x)$ and $spl_s(x)$ of, respectively, the longest and shortest paths from s to any given element x of the set of causet *Nodes*. We then aggregate the data into function $msp(l)$, which provides the *mean shortest path length* associated to each possible longest path length l:

$$msp(l) := Mean|\{spl_s(x)|x \in Nodes \wedge lpl_s(x) = l\}|$$

A very slow growth of this function reveals the presence of a wide gap between the lengths of the longest and shortest path from s to the other nodes. This is indeed what we observe in the *longest/shortest path plot* - the plots of the above *msp* function - for interval causets obtained from sprinkling in $M^{(1,1)}$. Instances of this plot will be included in many of the forthcoming figures (see, for example, the lowest function plots in the bottom row of Figure 7) as a benchmark for analogous plots of other causets.

Having introduced our loose but practical indicator of 'causet Lorentzianity', we are ready to examine some empirical results. In the next two sections we explore the two classes of causets that we can derive from the computations of the Ring Ant automata of subsection 5.6, namely the EH-causets (Event-History), in which computation steps are partially ordered through the mediation of write and read operations, and PA-causets (Permutation-Ant), in which a partial order is directly derived from the final permutation computed by the ant.

7 EH-causets from Ring Ant automata

By randomly sampling the huge space of EH-causets we could establish the following facts.

- All the EH-causets that we have examined - in the order of a few thousand - are planar. Recall that these causets are obtained by transitively reducing raw graphs: the latter in general turn out not to be planar.
- We find that around 70%-80% of the causets are linear paths - totally ordered sequences of nodes - or other slightly more elaborate periodic patterns that still grow, essentially, one-dimensional. Often the periodic phase is reached after an initial random-like transient phase. Emergent periodic patterns, called 'highways', are commonly observed in many other models of computation, e.g. in two-dimensional Turing machines [15, 37].
- In the remaining cases we find either random-like planar graphs, or regular planar graphs that we call *tiling causets*, for reasons to be clarified later, or intermediate cases in which randomness and regularity coexist.

7.1 Highways

In Figure 7 we show three examples of emergent highways in EH-causets from Ring Ant computations. Causets are shown in the upper row. In the lower row we present the corresponding longest/shortest path plots, each compared with the analogous plot for an interval causet obtained from sprinkling in $M^{(1,1)}$, which grows much more slowly. Eventually the functions for these causets grow linear: due to the periodic highway, the longest and (mean) shortest path lengths get coupled by a constant proportionality factor.

Under a spacetime perspective, these cases are not very interesting: establishing an analogy with the 'digital particles' that emerge in some cellular automata seems inappropriate, since in that case the localised structures move on a background structure, possibly interpreted as empty spacetime, which is missing here.

7.2 Random-like causets

The two rows in Figure 8 show two different, random-like EH-causets from Ring Ant computations. Each graph is shown in two alternative renderings (left and center). In each of the two diagrams on the r.h.s., the upper and lower functions represent, respectively, the longest/shortest path plots for the corresponding causet and for a sprinkled 2-D Minkowski causet (not shown in figure). The large gap between the two plots, in both cases, indicates that, in spite of their random-like character, these causets perform poorly w.r.t. our Lorentzianity criterion.

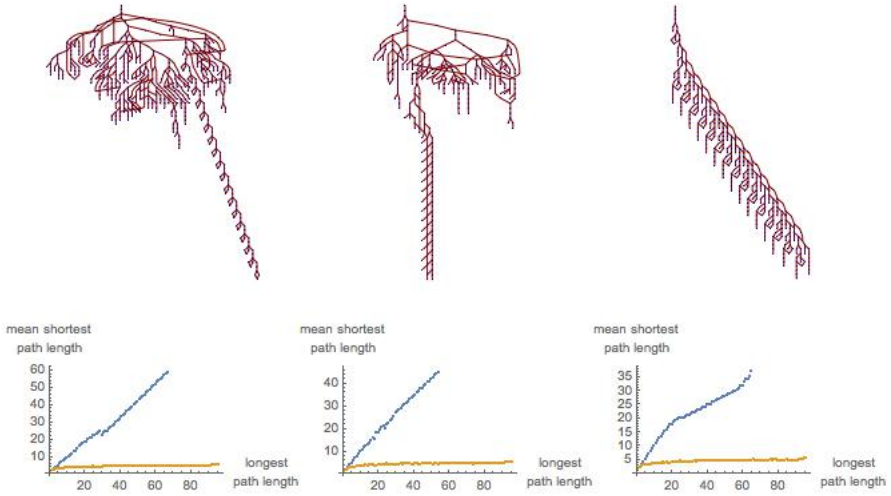

Fig. 7 Upper row: EH-causets from Ring Ant computations with emergent highways. Lower row: corresponding longest/shortest path plots, compared with the longest/shortest path plot of a 2500-node, sprinkled, 2-D Minkowski interval causet (lower function plots).

We may wonder whether better, i.e. lower longest/shortest path plots can be obtained by a randomised version of the EH-causet construction technique, one in which each time a new node n is added to the raw causet, three edges $p \to n$, $q \to n$, $r \to n$ are added to the graph, with p, q, r *chosen at random* among the existing nodes. The experiment is illustrated in Figure 9, where the longest/shortest path plot for a causet obtained by such a randomised procedure is compared, as usual, to that of a sprinkled causet. The randomisation yields an improvement over the random-like cases of Figure 8, but not enough to achieve the performance of sprinkled causets. [8] We note, incidentally, that these randomised causets are not planar.

7.3 Regular tiling causets

Figure 10 shows two regular EH-causets from Ring Ant computations, and their longest/shortest path plots. The graph on the left repeats the basic pattern of the graph on the right, and should not be likened to the periodic patterns - highways - of

[8] Due to computational bottlenecks, in Figure 9 and in subsequent analogous figures we tolerate possible differences between the maximum longest path length (about 100 links) achieved by the 2500-node, 2-D Minkowski sprinkled causet constantly used as a benchmark, and the maximum longest path lengths obtained for the various causets under scrutiny, as long as the growth trends for these functions are sufficiently clear.

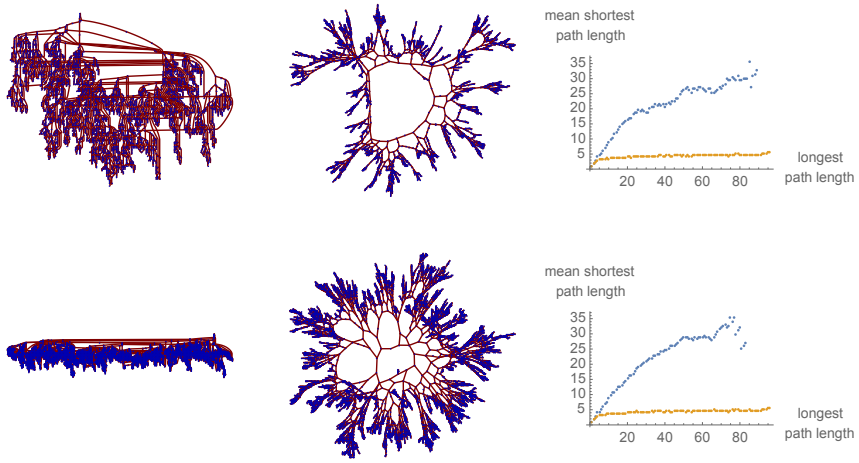

Fig. 8 Two random-like EH-causets from Ring Ant computations. At each row the same graph is shown with two different graph drawing algorithms (left and center). Right: the longest/shortest path plot for the causet at the left (upper function) is compared with the analogous plot for a 2500-node, sprinkled 2-D Minkowski causet, not shown (lower function).

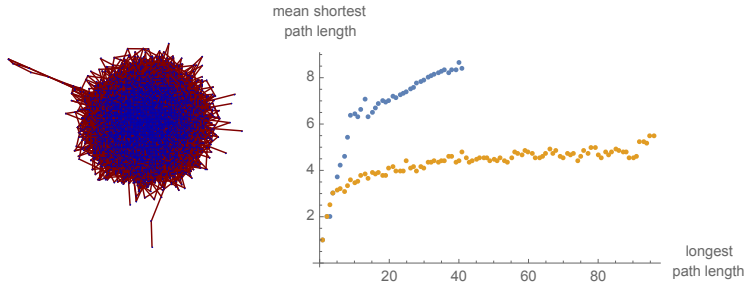

Fig. 9 Left: causet from randomised version of the Ring Ant EH-causet construction technique. Right: corresponding longest/shortest path plot (upper function), compared with the plot for a sprinkled causet (lower function).

Figure 7. Its corresponding longest/shortest path plot exhibits a slight improvement w.r.t. the linear growth of those periodic causets, but is still very far from the performance of random, sprinkled causets. The longest/shortest path plot for the graph on the right of Figure 10 performs better, with a considerably slower growth of the mean shortest path length.

These regular, 'tiling causets' are reminiscent of the tessellations of the hyperbolic plane, whose patterns are often represented on the Poincaré disc [9].

We recall that regular tessellations of the sphere, of the Euclidean plane and of the hyperbolic plane, can be represented by the Schläfli symbol $\{p,q\}$ which indicates

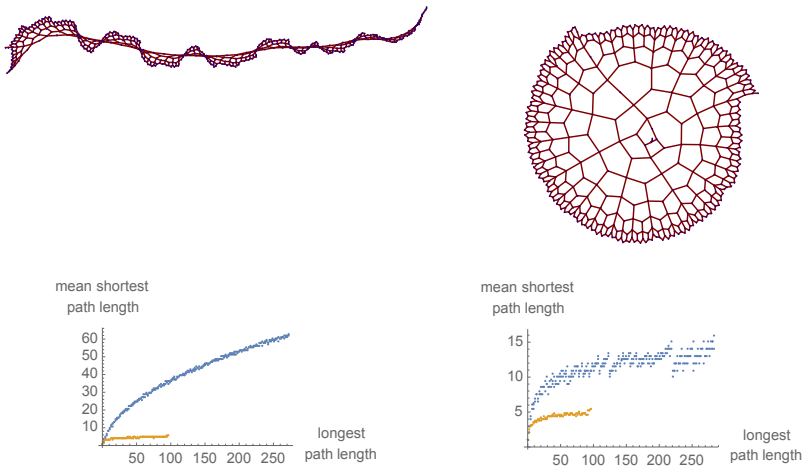

Fig. 10 Two regular EH-causets from Ring Ant computations (upper row), and their corresponding longest/shortest path plots (lower row). As in Figure 8, each plot is compared with the analogous plot for a sprinkled 2-D Minkowski causet, which appears, in both cases, as the lower function.

that q regular p-gons meet at each vertex. Based on the value of $1/p+1/q$ the following can be established:

- $1/p+1/q > 1/2$: the integer solutions are $\{\{3,3\},\{3,4\},\{3,5\},\{4,3\},\{5,3\}\}$, which are the Schläfli symbols for the five Platonic solids - tetrahedron, octahedron, icosahedron, cube, dodecahedron.
- $1/p+1/q = 1/2$: the integer solutions are $\{\{3,6\},\{4,4\},\{6,3\}\}$, corresponding to the familiar tilings of the plane by equilateral triangles, squares, hexagons.
- $1/p+1/q < 1/2$: there are infinite integer solutions, and as many regular tessellations of the hyperbolic plane.

Figure 11 shows two tessellations with regular pentagons meeting at vertices in groups of three (left) and four (right) yielding, respectively, positive and negative curvature. Curvature can indeed be defined also for planar graphs.

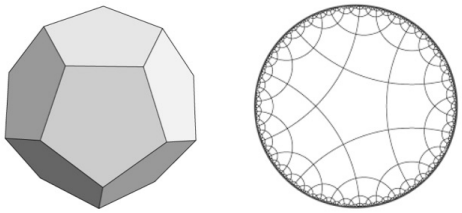

Fig. 11 Pentagonal tessellations $\{5,3\}$ and $\{5,4\}$.

Definition 1. (Combinatorial curvature)

For a planar graph $G(N,E)$, the combinatorial curvature cc of a node $x \in N$ is defined as:

$$cc(x) := 1 - degree(x)/2 + \sum_{f \sim x} (1/size(f)),$$

where summation is over all faces f incident with x. $\diamond$

Based on this definition, the dodecahedron $\{5,3\}$ has constant positive curvature 1/10, while hyperbolic tessellation $\{5,4\}$ has constant negative curvature -1/5.

We have mentioned regular tessellations and their curvature for comparisons with the causet in Figure 10-right. This planar graph has less symmetries than those of the tessellations in Figure 11, and is essentially formed by two concentric spirals of pentagonal faces; these faces meet at vertices in groups of three *or* four, with the exception of a pair of adjacent faces - a heptagon and an exagon - found at each round of one of the spirals. Following the spiral paths we find that nodes with degree 3 and degree 4 alternate, with the degree understood now as the sum of the in-degree and out-degree. Thus, roughly half of the nodes have positive curvature, and half have negative curvature. Link orientation is not shown in figure: both the radial links and those on the spiralling paths point outwards. Note that a hypothetical arrangement of the pentagons into a single spiral would yield a totally ordered, totally uninteresting causet: the presence of at least two spirals is essential for avoiding this collapse.

An interesting effect of the spiral arrangement is that a spiral path provides the longest path from the root, at the center of the graph, to any given node x, while an essentially radial path will provide a substantially shorter, alternative path to x. This 'trick' implemented by the graph explains the relatively good performance of the longest/shortest path plot. Note that a similar pair of paths - a long, spiralling route and a short, mainly radial one - can be found for any pair of nodes.

7.4 Mixed cases

Several cases were found in which elements of order - e.g. the tiling structure - are mixed with random-like components. One example is the EH-causet shown in Figure 12.

The mix of regularity and pseudo-randomness is one of our key motivations for investigating algorithmic causets. However, the performance of this causet in terms of our rough Lorentzianity indicator is quite poor, as revealed by the plot in the r.h.s. of the figure.

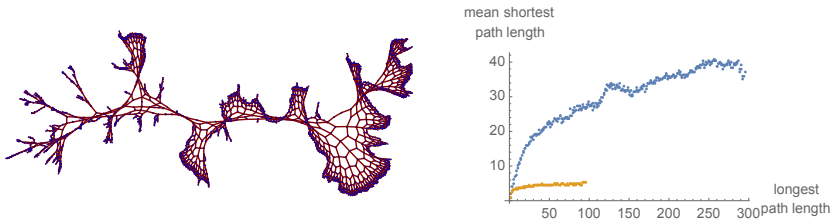

Fig. 12 Left: EH-causet from a Ring Ant computation, in which the pattern observed in the causets of Figure 10 appears mixed with random-like elements. Right: longest/shortest path plot for this causet (upper function) compared with the analogous plot of a sprinkled causet (lower function).

7.5 Causet from the fractal sequence

When discussing random-like EH-causets, we have introduced a randomised causet construction procedure in which each new node n is connected to previously created from-nodes p, q, r, chosen at random.

Now that we have seen examples of regular causets it is useful to explore the opposite solution and directly select those from-nodes by a completely deterministic procedure.

The fractal sequence [36] is a sequence of natural numbers defined as:

$$a(n) = k \qquad \text{if} \qquad n = (2k-1) * 2^m,$$

where $m = 0, 1\ldots$ and $k = 1, 2\ldots$ Its first 12 values are (1, 1, 2, 1, 3, 2, 4, 1, 5, 3, 6, 2); in Figure 13-left we plot the first 1000 values. The center of the figure shows the 502-node causet derived by splitting that 1000-element sequence into 500 consecutive pairs and using each pair as the from-nodes of each new node. [9] More precisely: we start with two nodes, labeled 1 and 2; the from-nodes of new

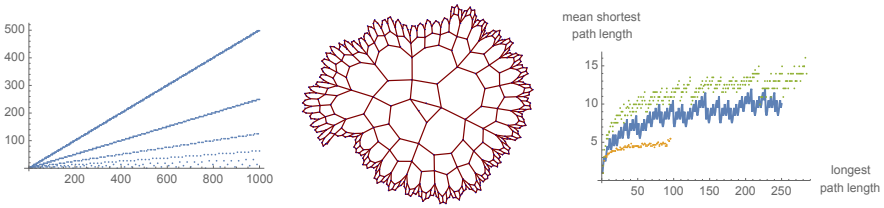

Fig. 13 Left: Fractal sequence. Center: Causet obtained from the sequence of pairs of elements of the fractal sequence. Right: longest/shortest path plot of the causet (intermediate solid line), compared with analogous plots for sprinkled causet (lower) and causet in Figure 10-right (upper).

[9] It turns out that segmenting the fractal sequence into triples, quadruples, etc., in place of pairs, does not yield interesting causets.

node 3 are (1, 1), since this is the first pair of the fractal sequence, thus parallel edges $1 \to 3$ and $1 \to 3$ are added; then node 4 is added, with edges $2 \to 4$ and $1 \to 4$, since (2, 1) is the second pair of the sequence; then edges $3 \to 5$ and $2 \to 5$ are added, and so on. The longest/shortest path plot of the causet is shown at the right: it is the solid line that appears in the middle, between the lower reference plot for a sprinkled causet and the upper longest/shortest path plot for the example of Figure 10-right, reproduced here for comparison.

The resulting causet is remarkably similar to the causet in Figure 10-right. The procedures for their construction are quite different, and the raw causets appear rather different too. But when transitively reduced, the two graphs reveal the same basic structure, formed by two concentric spirals of planar faces, although with the fractal sequence one of the spirals is formed by heptagons, not pentagons. Again the relatively good longest/shortest path plot is due to the simultaneous presence of long spiral and short radial paths between nodes. This appears to be a recurrent 'trick' in algorithmic causets, for keeping the growth rate of the longest/shortest path plots under control.

8 PA-causets from Ring Ant automata

As explained in Subsection 5.6, the deterministic Ring Ant automaton whose computations can be represented as partially ordered sets of events - yielding the EH-causets just discussed - also keeps a permutation π of the first n naturals, with n growing by one unit at each step. At any time the permutation can be readily turned into a PA-causet ('Permutation Ant'), as described in Subsection 5.4.

Analogous to the case of EH-causets, about 60% of these PA-causets are uninteresting, 1-D linear graphs. The remaining graphs split between regular structures, such as trees, and random-like structures.

8.1 Regular causets

In Figure 14 we show three very simple cases, meant primarily to further clarify the process of deriving the PA-causet (bottom row) from the final permutation (upper row). The elements of the permutation, intended as nodes of the causet, 'see' in their future lightcone only the elements/nodes that appear up-right to them in the permutation plot. Hence, with the permutation in Figure 14-left, that moves upward, all nodes are causally related with one another, thus yielding a linear path structure. With the two remaining permutations, that move downward, one or a few nodes near the origin of the plot 'see' almost all the remaining nodes in their future lightcone, while all these nodes are totally or largely causally unrelated, thus yielding a tree structure.

Two further regular cases are illustrated in Figure 15. The graphs look similar,

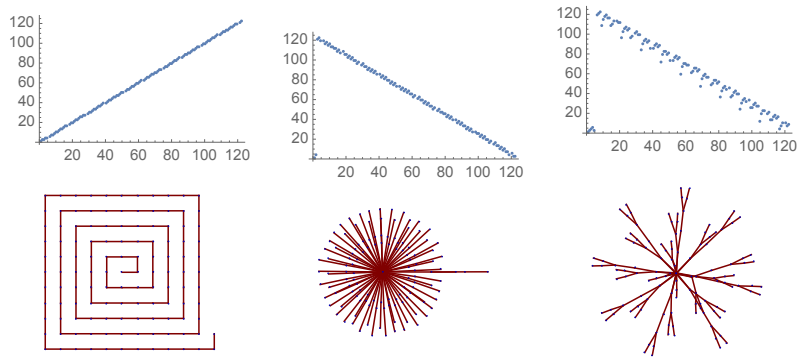

Fig. 14 Upper row: Final permutations computed by three runs of the Ring Ant automaton. Lower row: PA-causets derived from the permutations.

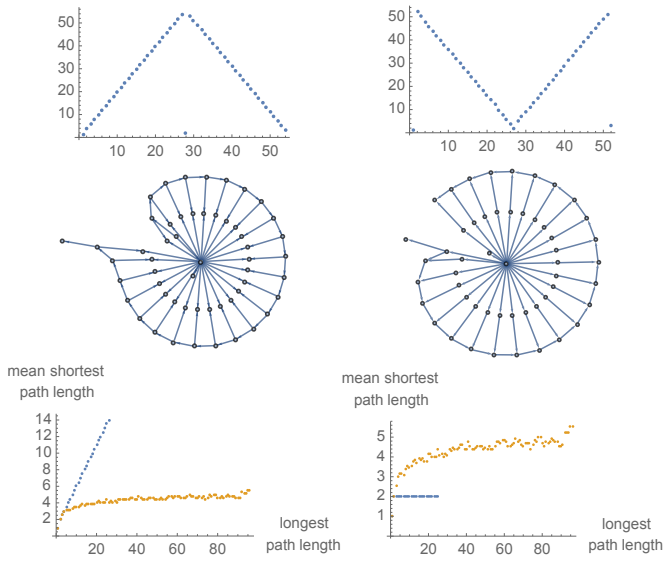

Fig. 15 Upper row: final permutations computed by two runs of the Ring Ant automaton. Central row: PA-causets derived from the permutations. Lower row: longest/shortest path plots. In both plots we compare the longest/shortest path plot of the corresponding causet - a straight line - with that for a sprinkled causet.

but differ in link orientation. In the graph on the right all radial edges point outward, while in the graph on the left the outer radial edges point inward. The reader may easily deduce the impact of this difference on the two longest/shortest path plots. In particular, the graph on the right is the first we have found whose longest/shortest path plot outperforms that of the sprinkled causet. The 'trick' is trivial, even more than that of the spiralling graphs: all nodes that are reached from the root - the

central node - by a longest path longer than 2 can be reached by a shortest path of length 2, thus the longest/shortest path plot is constant.

8.2 Random-like causets

Let us now consider some random-like cases. Two of these PA-causets are shown in the central row of Figure 16; the permutation from which each is derived appears in the upper row, and the corresponding longest/shortest path plots (dotted lines), compared with the analogous plot for sprinkled causets (solid lines), is shown in the lower row.

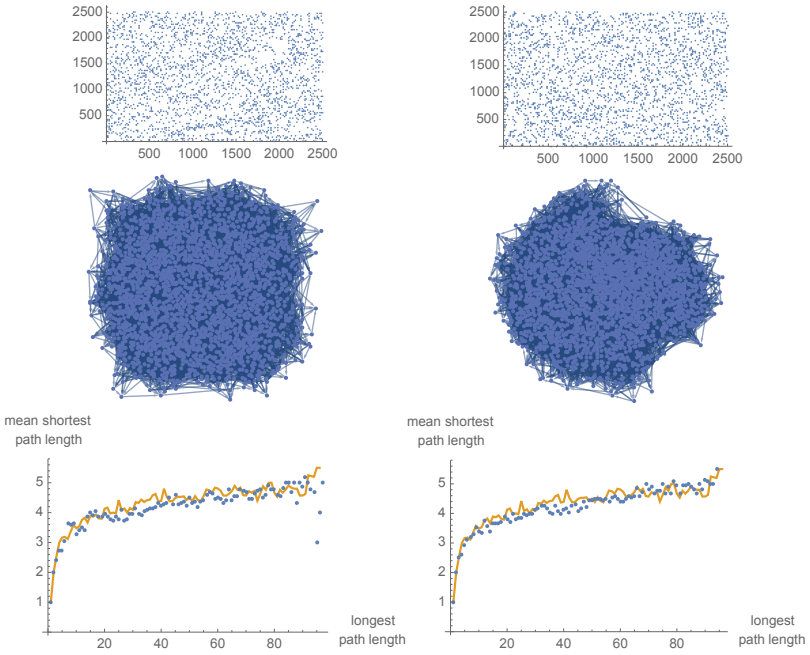

Fig. 16 Upper row: final, random-like, 2500-element permutations computed by two runs of the Ring Ant automaton. Central row: PA-causets derived from the permutations. Lower row: longest/shortest path plots. The dotted line is the longest/shortest path plot for the PA-causet, while the solid line is that of a sprinkled causet of the same size (2500 nodes).

At a simple visual inspection, these deterministic causets appear indistinguishable from the stochastic causets obtained by sprinkling. Most importantly, their longest/shortest path plots are equivalent to those from sprinkled causets. Thus, we have eventually found algorithmic causets that satisfy our test for Lorentzianity.

The significance of this success should be correctly assessed, and perhaps demystified: the result ultimately confirms that permutations are indeed equivalent to sprinklings, as discussed in Subsection 5.4, and that some instances of our Ring Ant automaton behave as 'good' pseudo-random number generators - an ability that, according to a conjecture proposed in [38], would indicate computational universality.

On the other hand, the crucial challenge - one that justifies our insistence on determinism - would be to find cases in which passing our Lorentzianity test combines with the presence of order, or of some mix of order and disorder in the causet, where 'order' is simply understood as a regularity that can be easily detected by visual inspection. Would this be possible?

Note that with PA-causets we can visually inspect two types of diagram - the permutation and the graph - with the idea that possible emergent order might be more apparent in one than in the other. By exploiting this advantageous circumstance, we have identified some additional interesting computations that seem to match, at least to some extent, our objective.

8.3 Mixed cases

In the case illustrated in Figure 17, the final permutation, shown on the left, is somewhat similar to the fractal sequence of Figure 13-left. A peculiarity of the corre-

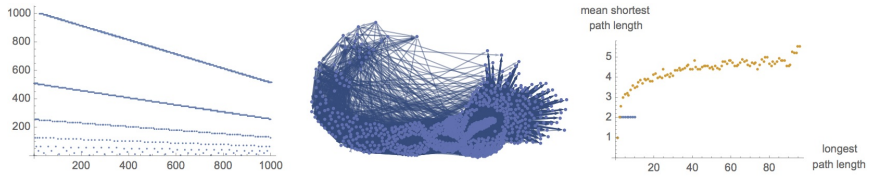

Fig. 17 A peculiar, regular Ring Ant computation. Left: final permutation. Center: PA-causet derived from the permutation. Right: longest/shortest path plot (the short horizontal segment), compared with the analogous plot for a sprinkled causet.

sponding causet, shown at the center of the figure, is that the shortest path from the root to any node is never longer than 2. For the 1000-node causet shown, the longest path has length at most 10. The resulting longest/shortest path plot is shown in Figure 17-right, where it is compared, as usual, with the analogous plot for a 2500-node sprinkled causet.

Given the high regularity detected in the permutation diagram - a regularity that goes unnoticed in the plot of the graph - we have reconstructed a version of the permutation by an ad-hoc algorithm, and plotted the graph with integer node coordinates as defined in Subsection 5.4, in order to better expose its structure. This is shown in Figure 18.

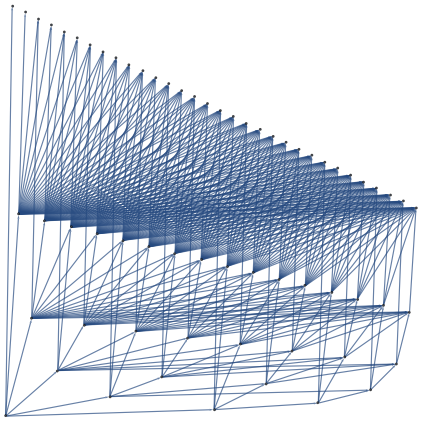

Fig. 18 The distilled structure of the causet in Figure 17.

The reader may easily check that the shortest path from the root - the leftmost node at the bottom - to any other node is never longer than 2, while the longest path does grow, but quite slowly, presumably as the logarithm of the number of nodes. This graph implements another brute-force 'trick' for keeping the growth of the longest/shortest path plot under control, alternative to the double spiral of Figure 10-right or the circular pattern of Figure 15-right.

Of course a flat longest/shortest path plot - one in which the mean shortest path length is constant, and independent from the longest path length - is as bad as one that grows linearly; our aim is to approximate the longest/shortest path plots of sprinkled causets.

It turns out that several permutations can be obtained in which aspects of regularity and of pseudo-randomness are mixed to varying degrees, with a beneficial effect on longest/shortest path plots. This phenomenon is illustrated in Figure 19, where three permutations of this kind and their corresponding longest/shortest path plots are presented.

A rather peculiar case of mix between order and randomness, in which the two components are sharply separated, is illustrated in Figure 20. As in the previous example of Figure 17, the plot of the final permutation is much more informative than the (default rendering of the) graph. This permutation appears as a mix of the descending 'lines' already seen in Figure 17, and a series of random-like slabs, and its structure appears to grow indefinitely. In Figure 21 we show the final permutation after 21,000 steps of the automaton. The appearance of these slabs plays an essential role in keeping the longest/shortest path plot close to that of sprinkled causets.

Do the 'lines' that appear in the permutation of Figure 17, or the remarkable mix of order and disorder in the permutation of Figure 21, correspond to properties of physical relevance for the associated causet/spacetime?

A first simple remark is that in both cases the 'lines' are formed by points that are in space-like relation with one another. However, these sets are not maximal space-

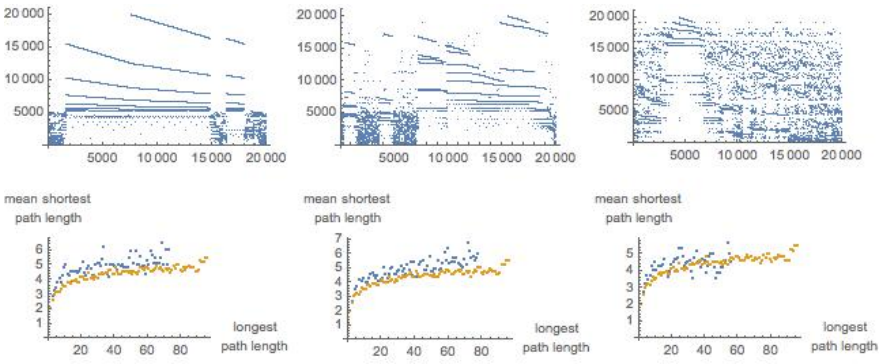

Fig. 19 Upper: permutations from 20,000-step computations of the Ring Ant automaton. Lower: longest/shortest path plots for the same computations, but limited to 2500 steps. These are compared, as usual, with the plot for a 2500-node sprinkled causet.

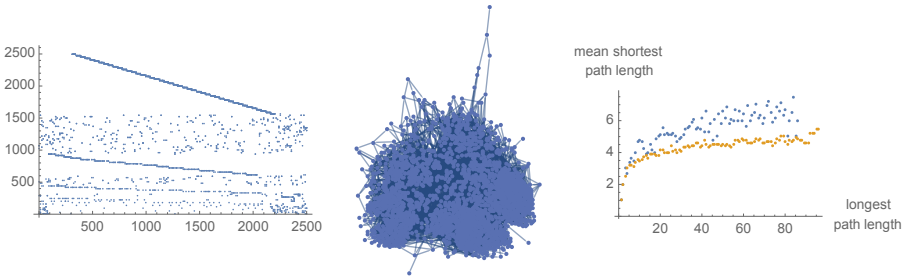

Fig. 20 A Ring Ant computation producing a mixed, ordered and random-like pattern. Left: final permutation. Center: PA-causet derived from the permutation. Right: longest/shortest path plot.

like regions, since each line has points that are space-like related also to additional, external points. Furthermore, it would be wrong to take these lines as separators between past and future, since links may well cross them, as documented in Figure 18. In conclusion, whether or not these lines might represent some meaningful 2-D spacetime pattern is still unclear to us.

On the other hand, in search for properties of physical relevance, we may wonder how these two cases perform under the Lorentz transformation.

In the left column of Figure 22 we show the sets of points of our two permutations after a 45 degree rotation, thus going back to their interpretation as events in Minkowski space $M^{(1,1)}$. In the column at the right we show their Lorentz transformations, relative to a reference system that moves at 1/3 of the speed of light c with respect to the system at rest.

Let us focus on the upper case. Consider the descending lines that form the original permutation π (Figures 17-left and 18), and assign them indices $i = 1, 2, ...$, starting from the top. Line i is formed by points whose y coordinates decrease by unit steps while x-coordinates are evenly spaced by steps of length 2^i. As a consequence, the angular coefficient of line i is $m(i) = -2^{-i}$. After the rotation by $\pi/4$,

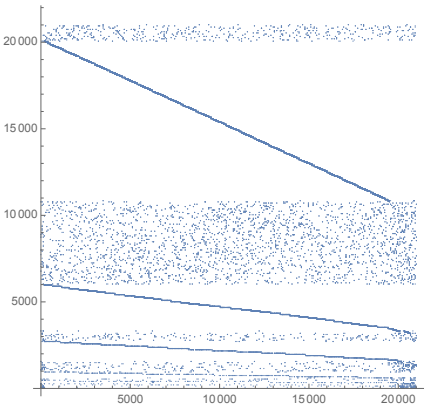

Fig. 21 Final permutation after 21,000 steps of the automaton of Figure 20.

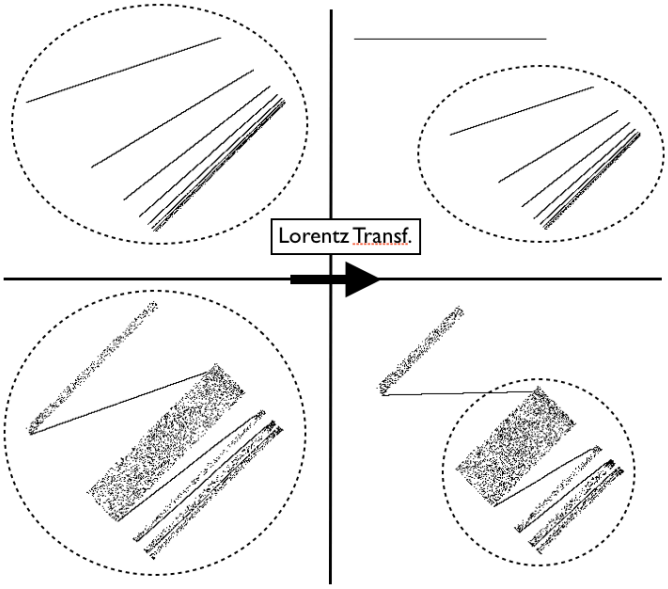

Fig. 22 Lorentz transformation of the two permutations of Figures 17 (32,767 nodes) and 21 (21,000 nodes). The points are plotted as seen by a system at rest (left column) and by a system moving at 1/3 of the speed of light c (right column). Selfsimilarity is manifested in the upper case, and approximated in the lower case, as highlighted by the dotted regions.

line i has angular coefficient $Tan(\theta(i)+\pi/4)$, where $\theta(i) = ArcTan(-2^{-i})$. After some manipulation, using:

$$Sin(ArcTan(-2^{-i})) = -2^{-i}/\sqrt{1+4^{-i}},$$
$$Cos(ArcTan(-2^{-i})) = 1/\sqrt{1+4^{-i}},$$

we obtain the values $mr(i)$ of the angular coefficients of the rotated lines in the upper-left diagram of Figure 22 :

$$mr(i) = \frac{2^i - 1}{2^i + 1}.$$

We wish now to apply the 2-D Lorentz transformation LT_v, for a frame moving at constant speed v, to the points (x,t) of the rotated lines:

$$LT_v(x,t) = (\gamma_v(x - vt), \gamma_v(t - xv)),$$

where γ_v is the Lorentz factor $1/\sqrt{1-v^2}$.

We can now compute the slope $mrLT_v(i)$ of the Lorentz-transformation LT_v of the rotated line i - another line - based on v and on the slope $mr(i)$ of the latter. Letting $LT_v(1, mr(i)) = (x_v(i), t_v(i))$, we have:

$$mrLT_v(i) = t_v(i)/x_v(i).$$

In general, a Lorentz transformation of the set of rotated lines, relative to a generic speed, will yield slopes that do not compare with those of the original line set. But if, for any integer j, we select speed $v(j) = (2^j - 1)/(2^j + 1)$, intended as a fraction of the speed of light c, then:

$$mrLT_{v(j)}(i) = \frac{t_{v(j)}(i)}{x_{v(j)}(i)} = \frac{mr(i) - v(j)}{1 - v(j)mr(i)} = mr(i - j).$$

The last equality indicates that the Lorentz transformation, for these specific $v(j)$ speeds, shifts the original lines so that they overlap with themselves.

For example, the set of lines with angular coefficients (1/3, 3/5, 7/9, 15/17, 31/33) after a Lorentz transformation $LT_{1/3}$ (i.e. for j = 1) become lines with coefficients (0, 1/3, 3/5, 7/9, 15/17). This is exactly reflected in the upper row of Figure 22, where the upper line on the r.h.s. is flat ($mrLT_{v(1)}(1) = 0$), while the lines below it repeat the slopes of the lines at the l.h.s.. After transformation $LT_{3/5}$ (j = 2), the new slopes are (-1/3, 0, 1/3, 3/5, 7/9), and so on.

The fact that this peculiar form of invariance is achieved only for a discrete set of inertial observer speeds is strongly reminiscent of the so called *Lorentzian lattices* [27], which are invariant only under a discrete subgroup of the Lorentz group.

In the case illustrated in the lower row of Figure 22, the overall structure of the cloud of spacetime points as seen from the system at rest (l.h.s. diagram) is only qualitatively repeated in a subset of the Lorentz-transformed set (r.h.s. diagram), due to the fact that the thickness values for the successive random-like slabs do not seem to follow a regular progression. Recall, however, that the Lorentz transformation leaves unaffected, in statistical sense, a cloud of points uniformly distributed in a region of $M^{(1,1)}$, such as these slabs.

We believe that the examples illustrated in this subsection represent promising preliminary steps in the search for algorithmic causets that mix regular and pseudo-random features while attempting to match the requirement of Lorentz invariance.

9 Conclusions

Does it make sense to talk about *spacetime computing*? In this chapter we hope we have identified a few attractive research items that can be legitimately grouped under this name, and be seen as dealing with a rather extreme form of natural computing.

As anticipated in the introduction, our presentation has mainly focused on some specific issues related to the modelling of discrete, algorithmic spacetime by causal sets, while several other relevant aspects have been left uncovered. A much wider treatment of the relations between nature and computation can be found, for example, in [39].

One of the aspects we have largely ignored here is quantum mechanics.

Although causal sets reflect in their basic structure the quantisation of spacetime, each node of the graph corresponding to a 'quantum' (an 'atom') of the latter, much more would be needed, e.g. in terms of dynamical laws, in order to set up a fully blown quantum-mechanical, algorithmic, causet-based theory of the natural universe, one involving Lagrangians, amplitudes, path integrals or sums over histories and all the conceptual tools that make quantum mechanics and quantum field theory so powerful (and so difficult). It is fair to say that very little progress has been done so far by theoretical physicists in this direction: for our purposes here, any detailed discussion on these aspects would be inappropriate.

It is worth mentioning, however, that the related question of whether and how we can use computers to *fully* and *exactly* simulate Physics, *and quantum mechanical features in particular*, was already addressed by Richard Feynman in 1981, in a famous keynote speech at a conference on the 'Physics of Computation' [10]. In that speech, also due to previous interactions with Fredkin, Feynman suggested, for this simulation, a visionary computer architecture based on cellular automata enriched with quantum mechanical capabilities, thus promoting the development of quantum computing.

The influence of these ideas is particularly evident in the work of Seth Lloyd, where all the different physical phenomena of the quantum world - e.g. all subatomic particle interactions - are interpreted as different quantum information processing activities, and the universe is seen as a huge network of programs that collectively determine the evolution ... of the universe itself [17].

If this multiplicity of different quantum computing processes is a correct picture of our world, then we might hope to be eventually able to 'crack' the code of some of these programs and profit from their imitation, in the same way as we do with bio-inspired computing.

However, the boldest conjecture about spacetime computing hints at the existence of a *single* algorithm at the root of everything, with that multiplicity of computing

processes only emerging from this unique source, and Schmidhuber [28] goes as far as suggesting that it is cheaper for a Turing machine to compute *all* possible computable universes rather than just one (ours), thus outlining an ultra-concise, computational theory of the multiverse.

Whether *this* is instead the correct picture - whether there is indeed a single, possibly elementary, possibly immutable, perhaps even deterministic and non-quantistic algorithm at the bottom of the universe or multiverse [10] - is still completely unknown. What is certain is that its discovery would deeply revolutionise the landscape of theoretical physics, catapulting spacetime computing to its forefront.

In light of the above conjecture, a sensible research track in the spacetime computing agenda is, in our opinion, the exploration of *abstract* algorithms for building discrete models of spacetime - models that, in the present chapter, we have identified with causal sets. By 'abstract' we mean that we abstain from coding into these algorithms any knowledge from theoretical physics - e.g. constants such as the speed of light - since everything should emerge from the algorithm itself, a posteriori.

One may object that the space of algorithms is potentially infinite, and that searching it blindly is unreasonable. We believe that these difficulties can be in part mitigated. First, the notion of computational universality (or Turing-completeness) acts as a sort of unifying factor for all models above a certain threshold of sophistication. Second, the space of qualitative behaviours that characterise the computations of potentially all conceivable algorithms is much smaller than the space of those algorithms, as widely shown by Wolfram [38].

Furthermore, the search for 'promising' algorithms is not totally blind, but should be guided by the early appearance of interesting emergent patterns - patterns that we should be able to recognise as useful for setting up, in the long, or *very long* run, the features of our familiar physical world. Among the valuable clues we include periodicity, self-similarity, pseudo-randomness, and 'digital particles'.

All of these features have been already found in the artificial, computational universe, but mainly in models such as cellular automata and 2-D Turing Machines that, due to other intrinsic limitations, lack physical realism. In this chapter we have therefore devoted much attention to the alternative model of (algorithmic) causal sets, and have extended the list of desirable emergent features to one of great physical significance: Lorentz invariance. We have discussed an indicator - the longest/shortest path plot - meant to reveal the closeness of an algorithmic causet to the ideal (2-D) Lorentzian causet, and have shown the extent to which regular, random-like, or mixed causets perform against this benchmark. Finally, we have identified causets that satisfy a form of discretised Lorentz invariance while offering a remarkable mix of regularity and pseudo-randomness.

Causal sets represent a more promising and physically realistic model than cellular automata or Turing machines. Of course, several of the regular, often planar causets that we have introduced appear naively simple, and remote from the complexity of the 4-dimensional causets that we might expect to represent spacetime,

[10] The possible existence of non-quantum mechanical laws at the roots of reality, below the layer of quantum mechanics, has been recently envisaged by G. 't Hooft [31].

at least at sufficiently high scales of observation. Still, they have been helpful for elucidating the variety of emergent properties that the model can offer.

Cracking the code that animates the elusive, discrete texture of the physical universe is the most ambitious goal of spacetime computing. As pointed out in [19], this goal would greatly benefit from the cooperation of various research areas, most notably quantum gravity and cosmology, complex networks and the theory of computing. Algorithmic causal sets seem to represent an ideal choice also in this respect: causets have acquired enduring attention and esteem from various schools of thought in quantum gravity, and, given their high abstraction level, they also offer a relatively easy access point for investigations and simulations by the curious computer scientist!

Acknowledgements

I express my gratitude to Alex Lamb and Marco Tarini for useful technical discussions, and to Stephen Wolfram for having initially stimulated my interest in this area of reasearch.

References

1. A. Adamatzky. *Computing in Nonlinear Media and Automata Collectives*. IoP Institute of Physics Publishing, 2001.
2. L. M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(11):1021–1024, November 1994.
3. J. Ambjørn, J. Jurkiewicz, and R. Loll. The self-organizing quantum universe. *Scientific American*, July 2008.
4. T. Bolognesi. Planar trinet dynamics with two rewrite rules. *Complex Systems*, 18(1):1–41, 2008.
5. T. Bolognesi. Algorithmic causal sets for a computational spacetime. In *A Computable Universe*. H. Zenil (ed.), World Scientific, 2013.
6. T. Bolognesi and A. Lamb. Simple indicators for Lorentzian causets, 2014. http://arxiv.org/abs/1407.1649v2 [gr-qc].
7. L. Bombelli, J. Lee, D. Meyer, and R. D. Sorkin. Space-time as a causal set. *Phys. Rev. Lett.*, 59(5):521–524, Aug. 1987.
8. J. Brown. *Minds, Machines, and the Multiverse - The Quest for the Quantum Computer*. Simon & Schuster, 2000.
9. J. H. Conway, H. Burgiel, and C. Goodman-Strauss. *The Symmetries of Things*. A K Peters CRC Press, 2008.
10. R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6 - 7):467–488, 1982.
11. E. Fredkin. Five big questions with pretty simple answers. *IBM J. Res. Dev.*, 48(1):31–45, 2004.
12. P. Gacs and L. A. Levin. Causal nets or what is a deterministic computation? *Information and Control*, 51:1–19, 1981.
13. M. Gardner. Mathematical games: the fantastic combinations of John Conway's new solitaire game 'Life'. *Scientific American*, 223(4):120–123, October 1970.

14. M. Hutter, S. Legg, and P. M. B. Vitanyi. Algorithmic probability. *Scholarpedia*, 2(8):2572, 2007. revision 151509.
15. E. Pegg jr. Turmite. From MathWorld–A Wolfram Web Resource, created by Eric W. Weisstein. http://mathworld.wolfram.com/Turmite.html, Jan. 21, 2011.
16. G. I. Livshits et al. Long-range charge transport in single G-quadruples DNA-molecules. *Nature Nanotechnology*, 9:1040–1046, October 2014.
17. S. Lloyd. Universe as quantum computer. *Complexity*, 3(1):32–35, 1997.
18. N. Margolus. Crystalline computation. In Anthony J. G. Hey, editor, *Feynman and Computation*, pages 267–305. Perseus Books, Cambridge, MA, USA, 1999.
19. F. Markopoulou. The computing spacetime. In S. Barry Cooper, Anuj Dawar, and Benedikt Löwe, editors, *How the World Computes - Turing Centenary Conference and 8th Conference on Computability in Europe, CiE 2012, Cambridge, UK, June 18-23, 2012. Proceedings*, volume 7318 of *Lecture Notes in Computer Science*, pages 472–484. Springer, 2012.
20. D. A. Meyer. The dimension of causal sets, 1989. PhD Thesis, MIT.
21. D. B. Miller and E. Fredkin. Two-state, reversible, universal cellular automata in three dimensions. In *CF '05: Proceedings of the 2nd conference on Computing frontiers*, pages 45–51, New York, NY, USA, 2005. ACM.
22. J. W. Mills. The nature of the Extended Analog Computer. *Physica D Nonlinear Phenomena*, 237:1235–1256, July 2008.
23. T. Nowotny and M. Requardt. Dimension theory of graphs and networks. *Journal of Physics A: Mathematical and General*, 31(10):2447, 1998.
24. M. Reuter and F. Saueressig. Quantum Einstein gravity. *New Journal of Physics*, 14(5):055022, 2012.
25. D. Rideout and P. Wallden. Emergence of spatial structure from causal sets. *J.Phys.Conf.Ser.174:012017*, April 30 2009. Proceedings DICE 2008 - doi:10.1088/1742-6596/174/1/012017.
26. L. A. Rubel. The Extended Analog Computer. *ADVAM: Advances in Applied Mathematics*, 14:39–50, 1993.
27. M. Saravani and S. Aslanbeigi. On the causal set-continuum correspondence. *Classical and Quantum Gravity*, 31(20):205013, 2014.
28. J. Schmidhuber. A computer scientist's view of life, the universe, and everything. In Christian Freksa, Matthias Jantzen, and Rdiger Valk, editors, *Foundations of Computer Science*, volume 1337 of *Lecture Notes in Computer Science*, pages 201–208. Springer Berlin Heidelberg, 1997.
29. L. Smolin. Atoms of space and time. *Scientific American*, pages 66–75, January 2004.
30. F. Soler-Toscano, H. Zenil, J.-P. Delahaye, and N. Gauvrit. Calculating Kolmogorov complexity from the output frequency distributions of small Turing machines. *PLOS one*, 9(5):e96223, May 2014.
31. Gerard 't Hooft. The cellular automaton interpretation of quantum mechanics, June 2014. http://arxiv.org/abs/1405.1548 [quant-ph].
32. T. Toffoli. Non-conventional computers. In J. Webster, editor, *Encyclopedia of Electrical and Electronics Engineering*, pages 455–471. Wiley & Sons, 1998.
33. T. Toffoli and N. Margolus. Cellular automata machines. *Complex Systems*, 1, 1987.
34. T. Toffoli and N. Margolus. Programmable Matter: Concepts and Realization. *International Journal of High Speed Computing*, 5:155–170, 1993.
35. T. Toffoli and N. Margolus. Invertible Cellular Automata: A Review. In *International Symposium on Physical Design*, 1994.
36. E. W. Weisstein. Fractal sequence. From MathWorld–A Wolfram Web Resource, created by Eric W. Weisstein. http://mathworld.wolfram.com/FractalSequence.html, valid on Nov. 25, 2015.
37. E. W. Weisstein. Langton's ant. From MathWorld–A Wolfram Web Resource, created by Eric W. Weisstein. http://mathworld.wolfram.com/LangtonsAnt.html, valid on Nov. 26, 2015.
38. S. Wolfram. *A New Kind of Science*. Wolfram Media, Inc., 2002.
39. H. Zenil (editor). *A Computable Universe*. World Scientific, 2013.
40. K. Zuse. Calculating space. Technical report, Proj, MAC, MIT, Cambridge, Mass., 1970. Technical Translation AZT-70-164-GEMIT. Original title: "Rechnender Raum".