
From: jeffrey E. <jeevacation@gmail.com>
Sent: Monday, February 19, 2018 11:50 AM
To: Joscha Bach
Subject: Re:

Energy? Unlimited? Equal per computation ? =A0 Non local ? Two places at once? Distribution s. Field effec=s time to compute / all the same time ? Synchronized

=div class="gmail_quote">

On Mon, Feb 19, 2018 at 6:24 AM Joscha [REDACTED] <[REDACTED]> >=wrote:

As you may have noticed, my whole train of thought on computationalism is based on the rediscovery of intuitionist mathematics under the name "computation".

<http://math.andrej.com/wp-content/uploads/2014/03/real-world-realizability.pdf> <<http://math.andrej.com/wp-content/uploads/2014/03/real-world-realizability.pdf>>

The difference between classical math and computation is that classically, a function has a value as soon as it is defined, but in the computational paradigm, it has to be actually computed, using some generator. This also applies for functions that designate truth. For something to be true in intuitionist mathematics, you will always have to show the money: you have to demonstrate that you know how to make a process that can actually perform the necessary steps.

This has some interesting implication: computation cannot be paradoxical. In the computational framework, there can be no set of all sets that does not contain itself. Instead, you'd have to define functions that add and remove sets from each other, and as a result, you might up with some periodic fluctuation, but not with an illegal state.

Intuitionist math fits together with automata theory. It turns out that there is a universal computer, i.e. a function that can itself compute all computable functions (Turing completeness). All functions that implement the universal computer can effectively compute the same set of functions, but they may differ in how efficiently they can do it. Efficiency relates to computational complexity classes.

The simplest universal computers known are some cellular automata, with Minsky and Wolfram arguing about who found the shortest one. Boolean algebra is Turing complete, too, as is the NAND gate, the lambda calculus, and almost all programming languages. The Church Turing thesis says that all universal computers can compute each other, and therefore have the same power.

I suspect that it is possible that the Church Turing thesis is also a physical law, i.e. it is impossible to build physical computer that can calculate more than a Turing machine. However, that conflicts with the traditional intuitions of most of physics: that the universe is geometric, i.e. hypercomputational. The fact that we cannot construct a hypercomputer, not just not in physics, but also not mathematically (where we take its existence as given when we perform geometry), makes me suspect that perhaps even God can't make a true geometric universe.

How can we recover continuous space from discrete computation? Well, spacetime is the set of all locations that can store information, and the set of all trajectories along which this information can flow, as seen from the perspective of an observer. We can get such an arrangement from a flat lattice (i.e. a graph) that is approximately regular and fine grained enough. If we disturb the lattice structure by adding more links, we get nonlocality (i.e. some

information appears in distant lattice positions), and if we remove links, we get spatial superposition (some locations are not dangling, so we cannot project them to a single coordinate any more, but must project them into a region).

On the elementary level, we can define a space by using a set of objects, and a bijective function that maps a scalar value to a subset of these objects. The easiest way of doing might be to define a typed relationship that orders each pair of objects, and differences in the scalar are mapped to the number of successive links of that relationship type. We can use multiple relationship types to obtain multiple dimensions, and if we choose the relationships suitably we may also construct operators that relate the dimensions to each other via translation, rotation and nesting, so we derive the properties of Euclidean spaces.

To get to relativistic space, we need to first think about how information might travel through a lattice. If we just equalize value differentials at neighboring locations, we will see that the information dissipates quickly and won't travel very far. To transmit information over large distances in a lattice, it must be packaged in a way that preserves the value and a momentum (in the sense of direction), so we can discern its origin. A good toy model might be the Game of Life automaton, which operates on a regular two dimensional lattice and allows the construction of stable, traveling oscillators (gliders). In Game of life, only the immediate neighbor locations are involved, so gliders can only travel in very few directions. A more fine grained momentum requires that the oscillator occupies a large set of adjacent lattice locations. SmoothLife is a variant of Game of Life that uses very large neighborhoods and indeed delivers stable oscillators that can travel in arbitrary directions.

I think I have some idea how to extend this toy model towards oscillators with variable speed and more than two dimensions. It may also be possible to show that there are reasons why stable traveling oscillators can exist in 1d, 2d and 3d but not in 4d, for similar reasons why stable planetary orbits only work in 3d.

To give a brief intuition about a traveling oscillator as a wavelet: Think of a wavelet as two concentric circles, one representing the deviation above zero, the other one the deviation below zero. They try to equalize, but because the catch up is not immediately, they just switch their value instead (This is the discretized simplification.) Now displace the inner circle with respect to the outer one: the arrangement starts to travel. Making the pattern stable requires distorting the circles, and probably relaxing the discretization by increasing the resolution. The frequency of the wavelet oscillation is inversely related to how fast it can travel.

You can also think of a wavelet as a vortex in a traveling liquid. The vortex is entirely generated by the molecular dynamics within the liquid (which are our discrete lattice computations), and it does not dissolve because it is a stable oscillator. The vortex can travel perpendicular to the direction of the fluid, which is equivalent to traveling in space. It cannot go arbitrarily fast: the progression of the liquid defines a lightcone in which each molecule can influence other molecules, and which limits the travel of every possible vortex. Also, the faster the vortex moves sideways, the slower it must oscillate, because the both translation and state change depend on sharing the same underlying computation. It will also have to contract in the direction of movement to remain stable, and it will be maximally contracted at the border of the light cone. (The contraction of a vortex is equivalent to giving it a momentum.)

An observer will always have to be implemented as a stable system capable of state change, i.e. as a system of vortices that interact in such a way that they form a multistable oscillator that can travel in unison. From the perspective of the observer, time is observed rate of state change in its environment, and it depends on its own rate of change, which in turn depends on the speed of the observer. This gives rise to relativistic time. Also, the observer does not perceive itself as being distorted, but it will normalize itself, and instead perceive its environment around itself as being distorted. As a result, the observer will always have the impression to travel exactly in the middle of its light cone. This model seems to recover Lorentz invariance, but with a slight catch: it seems to me that while speed of light is constant and there is no preferred frame of reference wrt acceleration, the resolution of the universe changes with the speed of the observer. No idea if this is a bug or a feature, or if it will be neutralized by something I cannot see yet before I have a proper simulation.

Obviously, all of the above is just a conjecture. I can make a convincing looking animation, and I am confident that many features like simultaneity etc. will work out, but I don't yet know if a proper numeric simulation will indeed work as neatly as I imagine.

> On Feb 18, 2018, at 09:00, jeffrey E. <jeevacation@gmail.com <mailto:jeevacatio@gmail.com> > wrote:
>
> i want to hear more on your views on projection spaces. =C2 also feel free to put some more meat on the
bones of the thin=ing re lorentz transformations
>
> --
> please note
> The information contained in this communication is
> confidential, may be attorney-client privileged, may
> constitute inside information, and is intended only for
> the use of the addressee. It is the property of
> JEE
> Unauthorized use, disclosure or copying of this
> communication or any part thereof is strictly prohibited
> and may be unlawful. If you have received this
> communication in error, please notify us immediately by
> return e-mail or by e-mail to jeevacation@gmail.com <mailto:jeevacation@gmail.com> , and
> destroy this communication and all copies thereof,
> including all attachments. copyright -all rights reserved

--

=C2 please note

The information containe= in this communication is confidential, may be attorney-client privileg=d, may constitute inside information, and is intended only for the u=e of the addressee. It is the property of JEE Unauthorized use, disc=osure or copying of this communication or any part thereof is strictly =rohibited and may be unlawful. If you have received this communicati=n in error, please notify us immediately by return e-mail or by e-mail =o jeevacation@g=ail.com <mailto:jeevacation@gmail.com> , and destroy this communication and all copies thereof, =ncluding all attachments. copyright -all rights reserved

</=iv> --94eb2c0d4cf6bb414c05658f4a3b-- conversation-id 14861 date-last-viewed 0 date-received 1519040991 flags 8590195713 gmail-label-ids 7 6 remote-id 796234