The coordinates of isolated accumulations are exactly computable real numbers

Jérôme Durand-Lose*

LIFO, Université d'Orléans,
B.P. 6759, F-45067 ORLÉANS Cedex 2.

Abstract. In Abstract geometrical computation, Turing computability is provided by simples machines involving drawing colored line segments, called signals, according to simple rules: signals with similar color are parallel and when they intersect, they are replaced according to their colors. These signal machines also provide a very powerful model of analog computation following both the approaches of computable analysis and of Blum, Shub and Smale. The key is that accumulations can be devised to accelerate the computation and provide an exact analog values as limits in finite time.
In the present paper, we show that starting with rational numbers for coordinates and speeds, the collections of positions of accumulations in both space and time are exactly the computable real numbers (as defined by computable analysis). Moreover, there is a signal machine that can provide an accumulation at any computable place and date.

Key-words. Abstract geometrical computations; Accumulations; Computable analysis; Computable number; Signal machine.

1 Introduction

Starting from a few aligned points, lines are initiated. When they intersect, they end and new line segments start. Each line is given a color and lines with the same color should be parallel. The new line segments are colored according to the colors of the removed ones.

What can kind of figure can one build with finitely many colors? Is this system computing in some way?

Such a system computes. It does so in the understandings of both Turing computability [Durand-Lose, 2005], the original Blum, Shub and Smale model [Blum et al., 1989, Durand-Lose, 2007, 2008a] and computable analysis [Weihrauch, 2000, Durand-Lose, 2009b, 2010]. Even the Black hole model of computation can be embedded there [Etesi and Németi, 2002, Hogarth, 2004, Lloyd and Ng, 2004, Andréka et al., 2009, Durand-Lose, 2006a, 2009a].

* `http://www.univ-orleans.fr/lifo/Members/Jerome.Durand-Lose`, `Jerome.Durand-Lose@univ-orleans.fr`

Given that the underlying space and time are Euclidean, thus continuous, can there be any accumulation? What can be said about them?

Yes this is easy to achieve (as on Fig. 1(a), time is always elapsing upward) and with a simple shift and rescaling, this *isolated accumulation* could happen anywhere.

In the present paper, we show that if the system is *rational* (*i.e.* the coefficients of the lines and the initial positions are rational numbers) then the coordinates of any isolated accumulation are *computable* real numbers. From now on, computable means according to computable analysis. We also prove that for any computable coordinates, there is a machine and initial configuration producing an isolated accumulation exactly at the point. Moreover, there exists a machine such that the coordinates of possible isolated accumulation are exactly the computable positions.

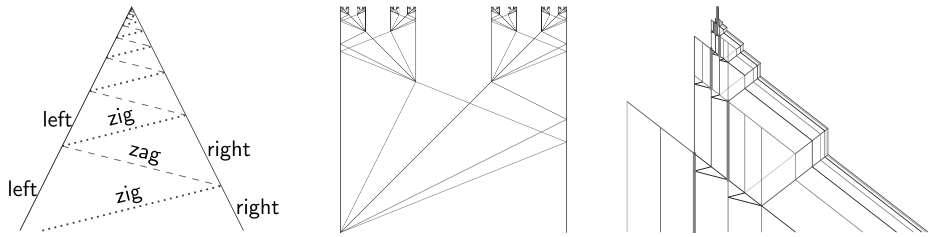
The system described above is a *signal machine* and the context is *abstract geometrical computations*. This was inspired by a continuous time and space counterpart of cellular automata [Durand-Lose, 2008b] similar to the approaches of Jacopini and Sontacchi [1990], Takeuti [2005], Hagiya [2005] and also as an idealization of collision computing [Adamatzky, 2002].

A *signal machine* gathers the definition of meta-signals (colors) and collision rules. An instance of a meta-signal is a dimensionless point called a *signal*. Each signal moves uniformly, its speed only depends on the associated meta-signal. The traces of signals on the space-time diagrams are then lines and as soon as they correspond to the same meta-signal, they are parallel. When signals meet, they are destroyed and replaced by new signals. The nature of emitted signals only depends on the nature of colliding ones.

One key feature of AGC is that space and time are continuous and Zeno effects can be implemented (as on Fig. 1(a)) in a way to provide unbounded acceleration. This has been used provide infinitely many discrete transitions during a finite duration so as to be able to decide the halting problem by implementing the so-called Black hole model of computation [Durand-Lose, 2009a]. This has also been used to carry out exact analog computations [Durand-Lose, 2008a, 2009b].

All these were achieved with rational signal machines: speeds as well as initial positions are rational numbers. Since the positions of collisions are defined by linear equations in rational numbers, the collisions all happen at rational positions. This is interesting since rational can be handled exactly in classical discrete computation.

One early question in the field was whether, starting from a rational signal machine, accumulation could lead to an irrational coordinate. This was answered, as expected, positively in Durand-Lose [2007] by providing an accumulation at $\sqrt{2}$. The question was then to characterize the possible accumulation points. It should be noted that forecasting accumulation for a rational signal machine is as undecidable as the strict partiality of a computable function (Σ_2^0-complete in the arithmetical hierarchy [Durand-Lose, 2006b]).



(a) Isolated accumulation (b) Accumulating on a Cantor set (c) Accumulating on a curve

Fig. 1. Examples of space-time diagrams.

In the present article, we are interested in *isolated accumulations*: there is no accumulation point arbitrarily close to it in the space-time diagram as on Fig. 1(a). As pointed out by the various examples on Fig. 1, there are many types of accumulations. The ones on Fig. 1(b) form a cantor set and the one of Fig. 1(c) are on a curve and are almost all accumulations of signals but not collisions.

In Durand-Lose [2009b], computable analysis is implemented with real numbers implemented as distances and accumulations happening at the exact location. The construction shows that the accumulation can be any computable number but the accumulation is not isolated (because of the convoy construction that produces a curve made of accumulating signals as on Fig. 1(c)). The construction has been improved in Durand-Lose [2010] so as to have only one accumulation. This asserts that for any computable real number, there is one rational signal machine and one initial configuration such that an isolated accumulated happen at the location. This is achieve by a two folding structures (a folding structure provide the machinery to provide unbounded acceleration), the inner one being nested inside the outer one. The inner one is producing, one symbol at a time, the infinite string representing the output (the computable analysis representation of the computable real number). The outer one shifts and shrinks according to the symbol generated (the inner one is frozen while the symbol is processed).

With a slight modification, this can be improved so that any computable number can be the date of an isolated accumulation. The first step is to ensure that each symbol extraction, shift and scaling have the same duration independently of the simulated type-2 Turing machine and output symbol. Once this normalization is done, it remains to add extra delays according to the output symbol.

Then with a parallel execution of two type-2 Turing machines, construction for both space and time with any two given computable real numbers can be achieved. By considering a universal Turing machine, a signal machine able to produce an isolated accumulation at any computable date or position is generated.

To prove that isolated accumulations only happen at computable coordinates, we explain how to construct a type-2 TM that generates the infinite representation of the coordinate starting from the signal machine and a rational interval of space and time where the isolated accumulation is expected.

Definitions are gathered in Sect. 2. Section 3 deals with generating accumulations at computable coordinates. Section 4 shows that the coordinates of isolated accumulations are always computable. Section 5 concludes the paper.

2 Definitions

2.1 Abstract geometrical computation

A *signal machine* collects the definitions of available meta-signals, their speed and the collision rules. For example, the machine to generate Fig. 1(a) is composed of the following meta-signals (with speed): left ($\frac{1}{2}$), zig (4), zag (-4), and right (-$\frac{1}{2}$). Two collision rules are defined:

$$\{\mathsf{left}, \mathsf{zag}\} \longrightarrow \{\mathsf{left}, \mathsf{zig}\} \quad \text{and} \quad \{\mathsf{zig}, \mathsf{right}\} \longrightarrow \{\mathsf{zag}, \mathsf{right}\}$$

It could happen that exactly 3 (or more) meta-signals meet. In such a case, collisions rules involving 3 (or more) meta-signals are used. There can be any number of meta-signals in the range of collision rules.

A *configuration* is a function from the real line (space) into the set of meta-signals and collision rules plus two extra values: $\oslash$ (for nothing there) and ✺ (for accumulation). If there is a signal of speed s at x, then, unless there is a collision before, after a duration Δt, its position is $x + s.\Delta t$. At a collision, all incoming signals are immediately replaced by outgoing signals in the following configurations according to collision rules (it maps two or more meta-signals into a set of meta-signals). Moreover, any signal must be spatially isolated —nothing else arbitrarily closed—, locations with $\oslash$ value must form an open set and the accumulation points of non $\oslash$ locations must have ✺ value. (This is a spatial, static, accumulation like on Fig. 1(c)).

A *space-time diagram* is the collection of consecutive configurations, they form a two dimensional picture. It must also verify that any accumulation point in the picture have ✺ value. (This are dynamical accumulations like on Fig. 1(a).)

An accumulation is *isolated* when there is no accumulation arbitrarily closed in the space-time diagram (it is surrounded by $\oslash$ in its configuration). It is purely dynamical. Considering the definition of light cone as on Fig. 2, an accumulation at (x_0, t_0) is isolated if, sufficiently close to (x_0, t_0):

- the grey zone is empty (there is nothing but $\oslash$);
- there are infinitely many signals and collisions but no accumulation in the casual past.

A signal machine is *rational* if all the speeds are rational and only rational positions are allowed for signals in the initial configuration. Since the location of

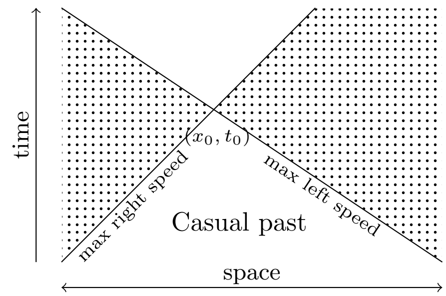


Fig. 2. Casual past of backward light cone.

collisions are solutions of systems of rational linear equations, they are rational. In any space-time diagram of a rational signal machine, as long as there is no accumulation, the coordinates of all collisions are rational.

Space and time are continuous; there is no absolute scale and the dynamics is uniform in both space and time. So that if the initial configuration is shifted or scaled so is the whole space-time diagram.

2.2 Computable number

Computable analysis uses type-2 Turing machines together with a valid representation of real numbers as an infinite sequence of symbols. A type-2 TM is nothing but a regular TM with an input stream (or read only tape) and an output stream (or write only tape). The TM should run forever and output the infinite representation of the computed real number. *Computable (real) numbers* are the ones generated by T2 TM with no entry.

The representation used is the *signed binary representation* [Weihrauch, 2000, Def. 7.2.4 p. 206], $\rho_{\mathrm{sb}} :\subseteq \{\bullet, \bar{1}, 0, 1\}^\omega \longrightarrow \mathbb{R}$, is defined only for infinite sequences with exactly one dot ($\bullet$) by:

$$w_0 \bullet d_1 d_2 d_3 \ldots d_n \ldots \longmapsto \nu_{\mathrm{sb}}(w_0) + \sum_{1 \leq i} \frac{d_i}{2^i}$$

where $w_0 \in \{\bar{1}, 0, 1\}^*$, $d_i \in \{\bar{1}, 0, 1\}$ and ν_{sb} is a naming of natural integers signed in base 2 ($\bar{1}$ stands for -1).

Since space-time diagram can be shifted and scaled just by doing it on the initial configuration and since rational shifts and coefficients preserve rationalness, generating (spatial) position in $[-1, 1]$ is enough.

For the temporal coordinates it is enough to consider $(0, 1]$. The above encoding cannot be used directly since only (strictly) positive duration can be added. As presented in Sect. 3.2, the encoding is then biased by the addition of:

$$\sum_{1 \leq i} \frac{2}{2^i} = 2 \ .$$

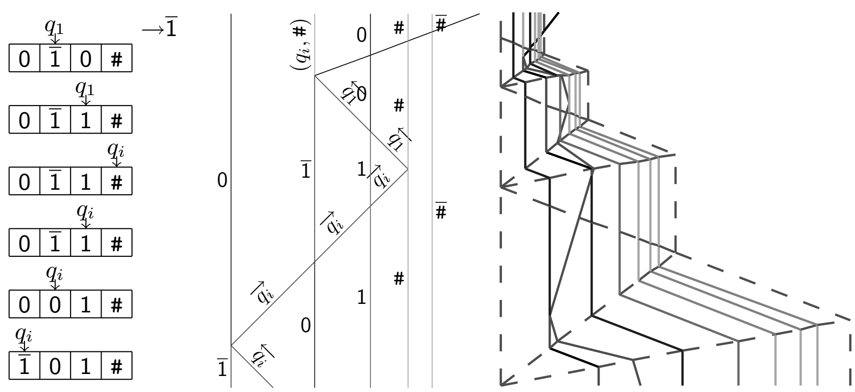
Adding or subtracting 2 does not change the computability of any number. But with the above increase, it is like working with values $\{1, 2, 3\}$ for "bits". The generated time is in $(2, 3]$, but again by scaling the initial configuration it can be anywhere.

3 Accumulating at computable coordinates

3.1 Spatial coordinate

In this section, the construction is only sketched. A more detailed construction can be found in [Durand-Lose, 2010].

Let x be any given computable number. It is represented by some TM M that, started on an empty tape and using no input stream, output its infinite signed binary representation. This machine can be implemented straightforwardly in abstract geometrical computation as shown on Fig. 3. The generated bits are signal leaving on the right.



(a) TM run (b) TM simulation (c) Embedded into the inner structure

Fig. 3. Simulating the T-2 Turing machine

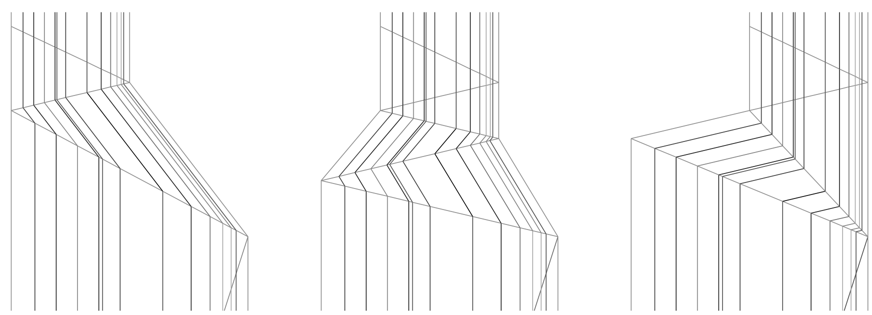
The construction is a two level structure made of shrinking structures: the *inner* and *outer* ones. Shrinking structures are construction that can be added to any signal machine in order to provide an unbounded acceleration ending in an isolated acceleration. These construction can be added to any signal machine: meta-signals and collision rules are added; existing ones are unaffected. The outer structure is in charge of driving the whole construction to accumulate at the right place, according of the generated (signed) bits produced by the inner structure.

The inner structure is in charge of *squeezing* (accelerating) M until a bit is output. When a bit is output, to avoid producing an unwanted accumulation, the inner structure and M freeze until another bit is asked for by the outer structure. To ensure that M is frozen, output is made blocking: M restarts on receiving an acknowledgement. One bit extraction is show on Fig. 3(c); it is the same iterations of M as on the rest of Fig. 3. The structure is indicated with the dashed lines. Squeezing ensures that the bit is obtained in bounded time. Moreover, when the inner structure (and the embedded M) is scaled down, so is the bound.

Basically the algorithm is:

1. The inner structure accelerates M until a bit is output.
2. The outer structure treats this bit by scaling down by one half and shifting the whole structure.
3. An acknowledgement is sent to the inner structure that restarts at first stage.

When the inner structure and M are frozen, only parallel signals are present. It is then easy to shift and scale them according to the bit received. The three cases are illustrated on Fig. 4.



(a) Case $\overline{1}$ (b) Case 0 (c) Case 1

Fig. 4. Outer structure dynamics.

Putting all together, the picture is like the one on Fig. 5. The accumulation at the top is exactly at the computable number. The inner structure is not provoking any accumulation anywhere else since it is always stopped and shifted.

3.2 Time coordinate

In the previous construction, special care has been taken to provide shrink and shift steps that have exactly the same height, *i.e.* take exactly the same amount of time. To master totally the duration, the delay in-between these steps has to be imposed by the construction and not depend on the output bit. Since the inner structure is accumulating, geometrically accelerating M, this duration is up bounded.

The next modification is:

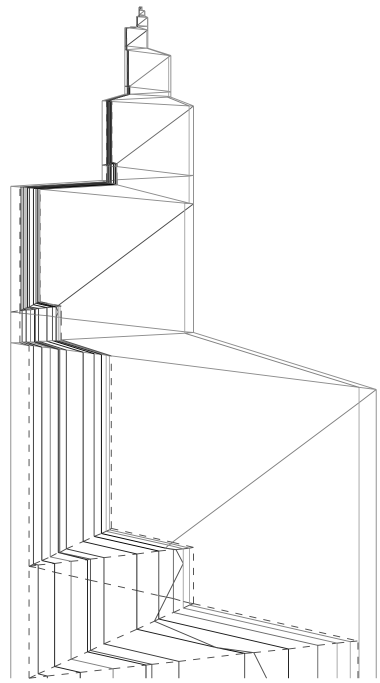


Fig. 5. The whole picture for spacial accumulation.

- the output bit is just collected and stored on the left, and
- the step is started by some signal emitted from the other side at the end of the previous step.

This starting signal is slow enough to ensure that it arrives after the generated bit.

At this point, the system accumulates at a date given as the sum of a geometric series (of rational factor). Considering the remark at the end of Sect. 2.2, to reach any computable number it suffices to add the same duration once for `0` and twice for `1`. Adding the same duration once or twice is very easy to achieve: once the shrink and shift step is done, one signal can go back and forth one or twice, just to let the corresponding amount of time elapse.

This way an accumulation can happen at any date that is a computable number.

3.3 Both coordinates

Starting from two computable numbers, x and t, it is possible to provide a signal machine and an initial configuration such that there is exactly one isolated accumulation, and it is located at (x, t).

Previous construction does not handle independent coordinates. This can be done very simply: a second inner shrinking structure and T2-TM is embedded inside the outer structure. The two inner structures are independent and each one squeeze its T2-TM to get bits. One is always received by the outer structure as an indication for the shift, the other always for the extra delays.

This ways, x and t are generated independently, each one operating on its coordinate.

4 Only computable coordinates

Let us consider a rational signal machine and an isolated accumulation at (x_0, t_0). There is a bounded portion of the space-time diagram such that all the collisions and signals are in the casual past of the accumulation.

More formally, there are rational numbers x^-, x^+, t^-, and t^+, such that: $x^- < x_0 < x^+$ and $t^- < t_0 < t^+$ and the grey zone of Fig. 2 contains only $\oslash$.

In the casual past, signals and collisions are only accumulating at (x_0, t_0) (since the accumulation is isolated there must exists such x^-, x^+, t^-, and t^+). At each date t, there is only finitely many signals and collisions. Moreover, if t is rational, then they are located at rational position.

Rational numbers can be implemented with exact precision so can rational signal machines (and it has been programmed in Java to generate all the space-time diagrams). So starting from x^-, x^+, t^-, t^+ and the signals and collisions present at t^- and simulating, the bounds x^-, x^+, t^-, t^+ can be improved and output. This generated an infinite sequence converging to (x_0, t_0), thus they are computable.

5 Conclusion

By considering a universal Turing machine and shift and scale operators, comes the following theorem:

Theorem 1. *There is a rational signal machine that can generate isolated accumulation at any computable coordinates. Moreover, the isolated accumulations of rational signal machine can only happen at computable coordinates.*

Bibliography

Andrew Adamatzky, editor. *Collision based computing.* Springer, 2002.

Hajnal Andréka, István Németi, and Péter Németi. General relativistic hypercomputing and foundation of mathematics. *Nat. Comput.*, 8(3):499–516, 2009.

Lenore Blum, Michael Shub, and Steve Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bull. Amer. Math. Soc.*, 21(1):1–46, 1989.

Jérôme Durand-Lose. Abstract geometrical computation: Turing computing ability and undecidability. In Barry S. Cooper, Benedikt Löwe, and Leen Torenvliet, editors, *New Computational Paradigms, 1st Conf. Computability in Europe (CiE '05)*, number 3526 in LNCS, pages 106–116. Springer, 2005. doi: 10.1007/11494645_14. URL http://www.illc.uva.nl/CiE/CiE2005/.

Jérôme Durand-Lose. Abstract geometrical computation 1: Embedding black hole computations with rational numbers. *Fund. Inf.*, 74(4):491–510, 2006a.

Jérôme Durand-Lose. Forcasting black holes in abstract geometrical computation is highly unpredictable. In J.-Y. Cai, S. Barry Cooper, and Angshen Li, editors, *Theory and Applications of Models of Computations (TAMC '06)*, number 3959 in LNCS, pages 644–653. Springer, 2006b. doi: 10.1007/11750321_61.

Jérôme Durand-Lose. Abstract geometrical computation and the linear Blum, Shub and Smale model. In S. Barry Cooper, Benedikt. Löwe, and Andrea Sorbi, editors, *Computation and Logic in the Real World, 3rd Conf. Computability in Europe (CiE '07)*, number 4497 in LNCS, pages 238–247. Springer, 2007. doi: 10.1007/978-3-540-73001-9_25.

Jérôme Durand-Lose. Abstract geometrical computation with accumulations: Beyond the Blum, Shub and Smale model. In Arnold Beckmann, Costas Dimitracopoulos, and Benedikt Löwe, editors, *Logic and Theory of Algorithms, 4th Conf. Computability in Europe (CiE '08) (abstracts and extended abstracts of unpublished papers)*, pages 107–116. University of Athens, 2008a.

Jérôme Durand-Lose. The signal point of view: from cellular automata to signal machines. In Bruno Durand, editor, *Journées Automates cellulaires (JAC '08)*, pages 238–249, 2008b. URL http://www.lif.univ-mrs.fr/jac/.

Jérôme Durand-Lose. Abstract geometrical computation 3: Black holes for classical and analog computing. *Nat. Comput.*, 8(3):455–572, 2009a. doi: 10.1007/s11047-009-9117-0.

Jérôme Durand-Lose. Abstract geometrical computation and computable analysis. In José Félix Costa and Nachum Dershowitz, editors, *International Conference on Unconventional Computation 2009 (UC '09)*, number 5715 in LNCS, pages 158–167. Springer, 2009b. doi: 10.1007/978-3-642-03745-0_20.

Jérôme Durand-Lose. Abstract geometrical computation 5: embedding computable analysis. Submitted to Nat. Comput., special issue on Unconv. Comp. '08, 2010.

Gábor Etesi and Istvan Németi. Non-Turing computations via Malament-Hogarth space-times. *Int. J. Theor. Phys.*, 41(2):341–370, 2002. gr-qc/0104023.

Masami Hagiya. Discrete state transition systems on continuous space-time: A theoretical model for amorphous computing. In Cristian Calude, Michael J. Dinneen, Gheorghe Paun, Mario J. Pérez-Jiménez, and Grzegorz Rozenberg, editors, *Unconventional Computation, 4th International Conference, UC '05, Sevilla, Spain, October 3-7, 2005, Proceedings*, volume 3699 of *LNCS*, pages 117–129. Springer, 2005.

Mark L. Hogarth. Deciding arithmetic using SAD computers. *Brit. J. Philos. Sci.*, 55: 681–691, 2004.

G. Jacopini and G. Sontacchi. Reversible parallel computation: an evolving space-model. *Theoret. Comp. Sci.*, 73(1):1–46, 1990.

Seth Lloyd and Y. Jack Ng. Black hole computers. *Scientific American*, 291(5):31–39, November 2004.

Izumi Takeuti. Transition systems over continuous time-space. *Electr. Notes Theor. Comput. Sci.*, 120:173–186, 2005.

Klaus Weihrauch. *Introduction to computable analysis.* Texts in Theoretical computer science. Springer, Berlin, 2000.