— Computability in Europe, Virtual conference, 6.07.2021 —

ANALOG COMPUTATION: CONTINUOUS VS EMPIRICAL

Paula Quinon

Paweł Stacewicz

Standard computation = digital computation

1. **Digital** computation, which provides a hard reference point, for an **analog** (non-standard) computation is:

   - **dominant** in contemporary IT practice,
   - described theoretically (modeled) by the Universal **Turing Machine**,
   - numerically describable by **natural** numbers ($\rightarrow$ *computable numbers in Turing sense*).

COMMENTS TO THE PREVIOUS SLIDE

Before we move on to a more detailed discussion of the various meanings of analogicity, it is worth emphasising that analog computing is usually contrasted with digital computing. Since this contrast will be the basis for one of the two meanings considered further on, let us briefly recall what the essence of digital computing is …

Digital computation, which is dominant in contemporary IT practice, is theoretically most often described by the Universal Turing Machine understood as an abstract machine. The structure and functions of the UTM are also extremely “close” to the practice of engineers and programmers. Most importantly, this model shows (one might even say: “explains”) at the most basic level of abstraction, regardless of the concrete representation of data and programs, what data processing by a programmable digital device consists in (Harel 1987).

So… it can be said that UTM somehow captures the essence of digitality, which is that data and its processing schemes (programs) are in the form of a symbolic code, consisting of distinguishable discrete elements.
In the case of UTM (itself), discreetness refers to the division of abstract machine tape into separate frames, a finite (though arbitrarily large) number of separate machine states, step-by-step head movements, and the symbolic alphabet used for writing data and commands.

Similarly, at the level of numerical description of data processing, digitality is contained in the fact that computations are performed on representations of numbers from a certain countable subset of the set of real numbers, today called *computable numbers* in Turing’s sense (Turing 1936). This applies to all aspects of the computing process, the numerical representation of input data, output data and program encoding.

Summarising the above remarks and returning to the notion of analogicity, it must be said that if analog computing is indeed something different (not necessarily opposite) to digital computing, then its theoretical models have to be significantly different from the UTM model.

Two basic meanings of analogicity

2. Analysis of the computations called analog allows us to distinguish two **basic meanings** of **analogicity**:

(AN-C) analog, i.e., **continuous** (non-discrete)

- *these are computations that allow the processing and generation of **continuous** data/signals (described at least by real numbers).*

(AN-E) analog, i.e., **empirical** (natural)

- *these are computations consisting in the realisation of physical processes which are **natural analogons** of certain mathematical operations.*

COMMENTS TO THE PREVIOUS SLIDE

When talking about analog computing, i.e. a kind of non-standard computing, there are two different (yet not necessarily separate) ways of understanding analogicity.

The first meaning, we shall call it AN-C, refers to the concept of *continuity*. Its essence is the generalisation (broadening) of digital methods in order to make not only discrete (especially binary) but also continuous data processing possible. On a mathematical level, these data correspond to real numbers from a certain continuum (for example, an interval of a form [0,1]), yet on a physical level – certain continuous measurable variables (for example, voltage or electric potentials).

The second meaning, we shall call it AN-E, refers to the concept of *analogy*. It acknowledges that analog computations are based on natural analogies and consist in the realisation of natural processes which, in the light of defined natural theory (for example physical or biological), correspond to some mathematical operations. Metaphorically speaking, if we want to perform a mathematical operation with the use of a computational system, we should find in nature its *natural analogon*. It is assumed that such an analogon simply exists in nature and provides the high effectiveness of computations.

In a short comment to this distinction, we would like to add that the meaning of AN-E has, on the one hand, a historical character because the techniques, called *analog*, which consisted in the use of specific physical processes to specific computations, were applied mainly until the 1960s. On the other hand, it looks ahead to the future – towards computations of a new type that are more and more often called *natural* (for example, quantum or computations that use DNA).
The meaning of AN-C, by contrast, is more related to mathematical theories of data processing (the theoretical aspect of computations) than to their physical realisations.

The categories AN-C and AN-E are not disjoint, as there are empirical computations that consist in processing continuous quantities. As such, they are AN-E, but also fall into the AN-C category.

AN-C and AN-E are not disjoint

- Categories AN-C and AN-E are **not disjoint**.
  They simply concern **different meanings** of the term "analog computing”.

- Putting it theoretically there are **empirical** computations that consist in **processing continuous** values.

- From an implementational point of view, it cannot be excluded that there are **actually continuous** quantities, using which we can actually compute **empirically**.

Analog as continuous

3. There are some intuitions leading to the idea of continuous computing:

- **continuous** quantities exist in **nature**, the real numbers (at least) correspond to them;
- they can be processed in a **continuous** manner, similarly as one operates on real numbers/functions.

digital	→	**N** ($\aleph_0$)
analog	→	**R** (**c**)

COMMENTS TO THE PREVIOUS SLIDE

AN-C computations, regardless of the model in which they are defined, allow the processing of continuous signals, mathematically described by real numbers (usually from a certain interval). Some of them also allow the processing of data in continuous-time, e.g., as specified by the GPAC model (proposed by C. Shannon in 1946).

Interestingly, if we refer again to the numerical description of data, AN-C computations also allow to operate on *uncomputable numbers* in Turing's sense (Turing 1936). This is because their set is the complement of the set of computable numbers to the set of real numbers. In other words, without the subset of uncomputable numbers, the set of real numbers would not be continuous. Recall that by definition, the domain of AN-C computation is continuous signals, and in mathematical terms: real numbers.

In consequence, we claim here, for theoretical (but not historical or technological) reasons, AN-C computations should not be seen as opposed to digital computations. Rather, they should be seen as their *extension* — just as the set of real numbers can be perceived as an extension of the set of natural numbers. The computational extension consists in the fact that any version of AN-C computation allows one to operate on a significantly wider set of data than any discrete set.

Models of AN-C computations

4. AN-C analog computations have many different **theoretical models** that define the types of operations (functions) performed on continuous data.

- For example, depending on how the computation is performed over time, we distinguish between:

  (**1**) models of AN-C computations, performed in **continuous** time [e.g., GPAC].

  (**2**) models of AN-C computations, performed in **discrete** time [e.g., BSS or continuous SSN].

COMMENTS TO THE PREVIOUS SLIDE

Depending on how the computation is performed over time, AN-C computation models fall into two categories (Bournez & Campagnolo 2008):

1) Models of AN-C computations performed *in continuous time*, and

2) Models of AN-C computations performed *in discrete time*.

The General Purpose Analog Computer (GPAC) model (Shannon 1941) and its various extensions, such as Extended Analog Computer model (EAC, see Rubel 1993), are examples of the first type of AN-C computation models, namely continuous-time AN-C models. Models of this type do not contain any kind of discretization (neither of the signals, nor of the modes of their transmission or processing), although it is possible in those models to process discrete signals that are distinguished quantities from a particular continuum.

In the case of continuous-time AN-C models, the data processing operations acceptable at the implementation level are described by a well-defined minimum of *continuous functions* and operations on these functions, such as addition, multiplication by a constant, addition of a constant to a function, differentiation or integration. Moreover, each model has a structural aspect that allows us to determine (e.g., in the form of a directed graph) the complex structure of operations leading from the data to the final result. Defining such a structure, i.e., designing a graph and assigning elementary functional operations to its nodes, can be treated as creating an algorithm for performing a certain task.

The second type of AN-C computation models, the discrete-time AN-C models, which include, for example, recursive neural networks (Siegelmann 1998) or the BSS model (Blum, Shub, Smale 1989), postulate discretization of time, and thus successive operations on continuous data in these models are strictly separated from each other. They are performed in discrete, clearly separated steps. For example, the aforementioned BSS model provides a discrete list of orders, each of which calls out separate operations executed at a specific time after the previous operation is completed. The property of continuity applies here to the individual operations — which, firstly, may be performed on data represented by real numbers (the data is assumed to be comprehensively recorded in special registers), and secondly, may consist in applying a rational function to a specific value from a given continuum. In short, computations are locally continuous.

Continuous computations as hypercomputations

5. At least some of the continuous computations, e.g., computations compliant with the model of real recursive functions, count as so-called **hypercomputations**.

- Their theory implies that they allow (upon an ideal implementation) to solve problems which are **not computable** in the Turing's sense, i.e., not solvable under the UTM model.

COMMENTS TO THE PREVIOUS SLIDE

Theoretical analyses indicate that some AN-C computations – described, for example, with the use of a model of recursive real-valued functions – have the status of *hypercomputations*. This means that they allow solving problems that are out of reach for digital computations which are formally expressed by the model of universal Turing machine. One of such problems is the issue of solvability of diophantine equations.

Although the theory of continuous computations does predict that they have *higher computational power* than digital computations (in theory!), the important question about practical *implementability* of continuous computations arises.

That is to say: if the physical world, the source of real data carriers and processes to process data, was discrete (quantised), we would never be able to perform any analog-continuous computations. And for this reason hypercomputability based on the processing of continuous quantities would be impossible to achieve.
This is an issue we will return at the end of the presentation.

Analog as empirical

6. Intuitively:

- Let's calculate, e.g., differentiate, integrate, solve NP-hard problems, by carrying out **dedicated physical processes**.

- appropriate computing systems exist in **nature** (animate and inanimate).

COMMENTS TO THE PREVIOUS SLIDE

Let us now turn to the second type of analog computing, which involves the implementation of certain mathematical (more broadly: formal) operations in an empirical way. We have called them AN-E (analog empirical). The empirically realised operations may apply to both continuous and discrete values.

Referring to some preliminary intuitions, this is how we can put the idea of AN-E computations:

- Let's calculate, e.g., differentiate, integrate, solve NP-hard problems, by carrying out **dedicated physical processes**.
- appropriate computing systems exist in **nature** (animate and inanimate).

Empirical computing (AN-E)

7.

The **computation** consists here in the carrying out of a **physical** process, which corresponds, according to some empirical theory, to a certain mathematical operation.

~ *empirical = physical, natural…*

COMMENTS TO THE PREVIOUS SLIDE

The physical implementation of specific AN-E computations consists in making use of *dedicated*, existing in *nature* and, so to speak, ready-made physical processes (Kari & Rozenberg 2008). For example, to differentiate a certain function, one can find in nature – according to a certain formalized theory – an effective differentiation process and apply it. To integrate a function, one simply uses some natural integrator. To solve a difficult optimization problem, one can rely on an existing, perhaps evolutionarily engendered, optimizing system. The efficiency of these processes is assumed to exceed that of traditional algorithmic solutions.

Generally speaking, AN-E type analog computing is performed according to the scheme:
a) find in nature a distinct process that "calculate something" (and is described by a certain mathematical formula),
b) build a computational system that uses such a process,
c) initiate computations by configuring the system appropriately,
d) take measurements in the system and interpret the outcome as the results of computations.

COMMENTS, AN EXAMPLE

Let us consider a simple illustrative example of calculating **a** quotient using the Ohm's law (I=V/R).

This law describes the flow of current in an electrical circuit (it must be added that this is an idealised circuit and its description does not take into considerations such factors as, for example, the self-inductance).

An analog computation is performed in the following way: a) adjust the voltage V and the resistance R appropriately, b) initiate the flow of current, c) take a measurement of current intensity I, interpreting the result as the value of the quotient.

A physical analogon of the computation is the flow of current in a circuit, which is initiated, controlled and observed with some intention, whereas the theory justifying the computation is the theory of current flow in a conductor (the idealising Ohm's law constitutes its element).

Empirical justification of AN-Es

Excerpt from our paper:

„ AN-E computations are closely related to the theories of **empirical** sciences (e.g., physics or biology). This means that specific computations of this type could neither be specified nor physically implemented without reference to a specific theory of this type.

Typically, such theories are treated as a tool for accurate **description** of physical reality in terms of mathematical structures and operations. Thus, their cognitive aspect is highlighted.

From the **computational** point of view (or more precisely: from the implementation one) they can be treated as a **basis** **for realizing** certain mathematical operations by means of physical processes described by these operations. With such an approach, a particular theory is treated as something that **justifies** the physical implementation of certain mathematical-algorithmic operations. It is therefore a justifying theory for a particular type of AN-E computation."

Models of AN-E computations?

8.

The computations are **dedicated**.

- there is no general model.
- a model may (eventually) be considered as a (empirical) **theory** that describes in mathematical terms the phenomenon being used for computing.

COMMENTS TO THE PREVIOUS SLIDE

Since empirical computations are dedicated, it is difficult to speak of the existence of *one universal* model that, like the theory of AN-C computation, would designate the elementary (mathematical) operations and structures that allow to describe any AN-E computation.

In their case, one can only speak of *local models*, which are actually fragments of theories that justify the attribution of a certain (simple or complex) mathematical operation to a certain physical process.

For example, a model of computation performed by a mechanical dedicated differential system would be the fragment of theoretical mechanics which describes the movement of bodies by means of differential calculus.

Some open questions…

- Are AN-C computations **physically feasible**?
  in particular: *are there in nature quantities* ***uncomputable*** *in Turing's sense (such quantities are a necessary component of any continuum)?*

- Does the **power** (including: speed) of some **AN-E** computations lies in the fact that they are also **AN-C** (i.e., they operate on continuous quantities)?

- Do analog continuous computations occur in **biological systems** (including: the human brain)?
  And if they do, are ***digital computations*** *sufficient for their effective simulation?*

Invitation to a discussion in an academic blog

We invite everyone to discuss presented topics via the **academic blog Cafe Aleph** (co-edited by Paweł Stacewicz)

Address of the blog entry and discussion:

https://marciszewski.eu/?p=10566

All comments can also be sent directly to our **e-mail** addresses

- Paula.Quinon@pw.edu.pl
- Pawel.Stacewicz@pw.edu,pl