

Paweł Stacewicz

Warsaw University of Technology
Faculty of Administration and Social Sciences
International Center for Formal Ontology
e-mail: pawel.stacewicz@pw.edu.pl
ORCID: 0000-0003-2500-4086

**ANALOGICITY IN COMPUTER SCIENCE.
METHODOLOGICAL ANALYSIS**

Abstract. Analogicity in computer science is understood in two, not mutually exclusive ways: 1) with regard to the continuity feature (of data or computations), 2) with regard to the analogousness feature (i.e. similarity between certain natural processes and computations). Continuous computations are the subject of three methodological questions considered in the paper: 1a) to what extent do their theoretical models go beyond the model of the universal Turing machine (defining digital computations), 1b) is their computational power greater than that of the universal Turing machine, 1c) under what conditions are continuous computations realizable in practice? The analogue-analogical computations lead to two other issues: 2a) in what sense and to what extent their accuracy depends on the adequacy of certain theories of empirical sciences, 2b) are there analogue-analogical computations in nature that are also continuous? The above issues are an important element of the philosophical discussion on the limitations of contemporary computer science.

Keywords: computer science, methodology of computer science, analogicity, analogue computations, analogousness, natural computations, hypercomputations.

Introduction¹

Although analogue techniques that are discussed in this paper are not part of the mainstream of contemporary computer science, they seem to deserve philosophers' attention for two reasons. Firstly, it is a meaningful historical fact that at the early stage of the development of computer science, analogue machines were perceived as a real alternative to digital machines, and they were also intensely developed. Suffice it to mention V. Bush's differential analysers (built already in the 1930s), or C. Shannon's theoretical GPAC model of analogue computations (introduced in 1941). It is interesting to notice, however, that a certain return to the idea of analogicity can be

observed in the latest research, for example, in such areas as continuous hypercomputing and certain types of natural computing.

Secondly, philosophers' interest in analogicity can be inspired by the methodology of computer science. Above all, the unquestionable ambiguity of the term "analogue" (for example, techniques) forces us to make a methodological effort to organise and elucidate meanings that appear in different contexts. Furthermore, when some meanings are recognised as the most relevant, it becomes necessary to ponder the practical performability and reliability of some or other analogue techniques.

In the following parts of my work, I will undertake both above mentioned *methodological* tasks. To begin with, I will define two main meanings of analogicity: the first referring to the concept of continuity, the second to analogousness; then I will move to analyse some examples of such techniques and, finally, to formulate philosophically important questions about their factual (and not only theoretical) performability.

1. The intuitive understanding of the term "analogue method"

There is no doubt that the issue of analogicity is especially applied in *technical* contexts, when referring to such artefacts as machines, devices, technologies, data storages or signals. Such specialist references, however, do not prevent ordinary people (also those that are not very tech-savvy) from understanding the term "analogicity" in an *intuitive*² way.

At the beginning, I suggest we look at the most popular colloquial meanings which reveal some crucial features of analogicity that are not necessarily related to computer science. We should take into consideration the term "analogue method," which is a very general and neutral term from the viewpoint of information processing technologies.

Frequently only roughly defined, the term is used to name past, outdated methods used *before the age of digital technology*. In other words, analogue and digital methods are juxtaposed, but at the same time either it is not specified what their distinctness consists in, or it is vaguely mentioned that such distinctness is expressed in the rule of processing *continuous signals* (that for some reasons cannot be expressed binarily, or in a broader sense: are not discrete).

The above-presented general understanding of the term harmonises with another colloquial meaning, according to which analogue methods are such methods that allow operating not on some artificially created codes, but on directly given *physical quantities*. Such quantities can be found directly

in nature; moreover, they are subject to the laws of nature that people can employ for their needs. For example: when a sound is transmitted via a traditional analogue communication system, it is not encoded with the use of some symbolic code (such as zero-one notation), but “taken directly from nature,” and then, according to the relevant laws of physics, it is consistently transformed into various types of waves propagated in respectively suitable mediums (such as air, a membrane or an electric wire).

Another colloquial meaning, strongly related to the previous one, refers directly to the notion of *analogy* – it is linguistically visible in the name “analogue method.” This notion seems to express accurately a technological fact that at least some methods in technology are based on certain physical analogies. For example: on a typical analogue phonographic record, sound is transformed into an adequately shaped surface of the plate (uneven grooves) that resembles the shape of a wave propagating the sound in the air; thus, there is a direct analogy between the shape of the wave and the physical recording (representation) of the sound.

Finally, let us move from technical contexts to mathematics, where we can also observe the use of phrases such as “analogue method.” Applying them, one has in mind *non-computational* (or: in-formal) methods that consist in the use of physical equivalents of mathematical objects (for instance graphic representations) that are not entirely accurate yet appeal to the human imagination. A good example is a graphic method of solving sets of linear equations with two unknowns, according to which to solve a set, one has to draw the graph for each of the equations and look for (identify in the graph) the point of their intersection (Although this method has its algebraical-arithmetical equivalent, its essence is different.).

If we treat the above-mentioned examples as an introduction to further clarification, it must be noted that within the issue in question two distinct “points of reference” of analogicity come to the fore: 1) the notion of continuity, and 2) the notion of analogy (compared additionally to physical reality).

2. Analogicity in computer science

In the context of computer science that we are discussing here, I will hereafter use the term “analogue data processing techniques,” or in short: *analogue techniques*,³ to name analogue methods. According to the convention observed in computer science – which takes into consideration the fact that, from a mathematical perspective, numbers can be ascribed to all data

and operations on numbers can be ascribed to all operations on data – I will define the aforementioned techniques as *analogue computations*.⁴

One more issue needs an introductory explanation. In my following deliberation, I will include two aspects of computations which are typical of computer science: a) *theoretical* – that manifests itself in the design and analysis of various mathematical models of computation (such as the Turing model) and b) *physical/implementational* – that is highlighted when examining physical systems used to carry out computations (such as electronic analogue systems) and their corresponding physical processes.

2.1. Two basic meanings of analogicity

Regardless of the (technical) aspect considered, in contemporary computer science there exist two different (yet not necessarily mutually exclusive) ways of understanding analogicity.

The first meaning, we shall call it AN-A, refers to the concept of *analogy*. It acknowledges that analogue computations are based on natural analogies and consist in the realisation of natural processes which, in the light of certain natural theory (for example physical or biological), correspond to some mathematical operations (Ifrah, 2006). Metaphorically speaking, if we want to perform a mathematical operation with the use of a computational system, we should find its *natural analogon* in nature. It is assumed that such an analogon simply exists in nature and provides highly effective computation. The initial examples of AN-A techniques (that will be expanded further on) are: the calculation of quotient using the Ohm's law (an illustrative example) or the integration of functions using physical integrators (a realistic example).

The second meaning, we shall call it AN-C, refers to the concept of *continuity*. Its essence lies in the generalisation (broadening) of digital methods in order to make not only discrete (especially binary) but also continuous data processing possible.⁵ On a mathematical level, these data correspond to real numbers from a certain continuum (for example, an interval of a form $[0, 1]$), while on a physical level – to certain continuous measurable variables (for example, voltage or electric potentials) (Mycka & Piekarz, 2006).

In a short comment to this distinction, I would like to add that the meaning of AN-A has, on the one hand, a historical character because the techniques, called *analogue*, which consisted in the use of specific physical processes to perform 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 computations or those that use DNA; see section 3.1).

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. Perhaps, it is solely a theoretical meaning that in practical applications is reduced to discreteness/digitality (I will develop this subject in section 4.2) due to physical features of data carriers.

I would also like to add that since analogousness does not exclude continuity, that is both continuous and discrete signals can be processed by applying analogy, the above-differentiated meanings are not entirely in opposition to each other.

2.2. Analogicity in relation to analogousness

The essence of AN-A analogue techniques can be called *analogousness* – that is the necessity to use some natural analoga/equivalents of the performed mathematical operations for computing purposes. Such computations are of definitely more *empirical* character than digital techniques which refer to extremely simple states/phenomena.⁶ Their specificity can be presented by the following points: a) find in nature a distinct process that “calculates something” (and is described by a certain mathematical formula), b) build a computational system that uses this process, c) initiate computations configuring the system, d) take measurements in the system and interpret the outcome as the results of computations (Ifrah, 2006, p. 656).

It should be underlined that computations arranged in such a way are always justified by a specific *physical theory*, which combines performed mathematical operations with phenomena used for their performance. Based on this theory, it can be acknowledged that a given phenomenon has such and such mathematical description; and to the contrary: that operations constituting such a description can be performed physically within the phenomenon (or more accurately: the results of these operations can be identified with the results of respective measurements). This can be explained in more details by the following illustrative (not realistic) example.

Example 1.

Let’s consider the calculation of 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, self-inductance (Krajewski, 1998)).

An analogue computation is performed in the following way: a) adjust the voltage V and the resistance R appropriately, b) initiate the

flow of the electric 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).

It should be provisionally noted (we will revisit this issue towards the end of the paper) that the validity and accuracy of AN-A analogue computations must depend on the level of the adequacy of the theory describing the process on which the computations are based (in this case: the flow of current through a conductor). If the theoretical mathematical formula ($I = V/R$) sufficiently characterises the above-mentioned process (it should be "good enough", and it must be kept in mind that such formulas refer to idealised situations), the computation can be acknowledged as accurate enough.

The most important characteristic of computations represented by the above example – a characteristic which accounts for the name "analogue computations" – is *analogousness*. Such a quality is manifested on two levels.

Firstly, as it is clearly presented by the above example, the process which performs a computation is a physical analogon of a certain mathematical operation. Therefore on this level we encounter correspondence [a formal operation – a physical phenomenon]. (In the example, this occurred between the calculation of quotient and the flow of current.)

Secondly, the *basic* process that is used to perform a computation can be applied to issues related to *similar physical processes*. The analogy between these two processes provides such analogy: they are analogous because an identical formal model describes both of them. To explain this, we should take into consideration an integrating physical system (an integrator) that is based on a physical process A . Due to the fact that this process is formally described by a theory of integration (for example, the Riemann's theory), it can be used in computations that concern a broad category of other processes (B, C, D, \dots) described by this or that integral (Fichtenholz, 1997).

Lastly, let us mention one more issue that leads directly to the second meaning of analogicity: *continuity* is very often perceived as a crucial feature of AN-A analogue computations. It is underlined because early (and specialised) analogue systems were used, above all, to solve analytical problems (concerning, for example, differential equations) that are defined

and described using continuous real numbers and mathematical structures (such as differentiable functions) “constructed” upon them (Ifrah, 2006, pp. 651–660). Moreover, processes that performed computations within the discussed systems were characterised mathematically with continuous (analytical) objects.⁷

Despite this, as explained earlier, the essence of AN-A analogue techniques does not come down to their continuity.

2.3. Analogicity in relation to continuity

The afore-mentioned concept of continuity constitutes a basis for determining the second meaning of analogicity, that is AN-C. From the point of view of modern computer science, this meaning should be recognised as dominant, which is manifested by a common tendency for identifying analogue computations with *continuous computations*⁸, or, in other words, defining analogue computations in opposition to digital computations. The essence of the former is sought in the fact that they make it possible to process and generate continuous (not only discrete) data represented in practice by continuous physical quantities. In short: AN-C analogicity is defined within the distinction *discrete-continuous*.⁹

Having in mind both above-analysed meanings, it should be stated that in the methodology of computer science as well as in the general understanding of IT users, a crucial shift in meaning has occurred: from analogicity understood as analogousness towards analogicity understood as continuity. It came about in the course of engineering practice, as more and more *universal* analogue machines were created and then those analogue devices gradually became supplanted by *digital* machines that were more reliable (Ifrah, 2006, pp. 651–662).

In an attempt to widen the scope of application of analogue machines, the first process consisted in searching for such a *minimal* set of processing components whose different arrangements (connections) would guarantee the performance of the broadest class of functions. The typical “minimal” components included amplifiers and integrators – suited for processing continuous signals. Because of the growing popularity of such solutions, step by step analogicity became identified with the *continuity* of processed data. The characteristic feature of analogousness (that is the use of natural analoga to computations) receded into the background because universal analogue devices performed certain combinations of several basic operations on continuous signals (Mycka, 2010).

As a counterbalance, the second process – which consists in the far-reaching universalisation of machines, interconnected with the digitisation

of signals and computations – caused analogue techniques, competitive to digital ones, to become associated with, above all, *non-digitality* (non-discreteness), that is with continuity.

The contemporary concept of analogicity in the sense of AN-C can be precisely expressed by *mathematical models* of analogue-continuous computations.¹⁰ Following [Mycka, Piekarcz 2004], they can be divided into: 1) continuous time models of continuous computation – for example, the GPAC model (Shannon, 1941) and 2) models of continuous computations performed in discrete steps – for example, the BSS model (Blum et al., 1989).

Because of both the idea of “full continuity” (that is the lack of any discretisation, also in the aspect of time) and the historical antecedence, the GPAC model seems to be the one worthiest of consideration.¹¹ This model describes the way of processing continuous signals (mathematically speaking: continuous functions) with a minimum number of functional operations that at the level of model constitute nodes/vertices of an oriented graph joining input and output signals. Such a graph presents the order in which an input signal is to be processed (sometimes simultaneously) by the components of the system that correspond to particular nodes. The said minimum number of operations includes: multiplying a function by a constant, adding a constant to a function, adding functions and integrating functions.

In engineering practice, the GPAC model is performed with the use of analogue electronic circuits (in short: AEC) that process data on the basis of adequately configured operational amplifiers. I will present how they work in example 2.

Example 2.

In technical terms, every AEC is composed of a finite number of *basic* systems which create nodes of a net and electrical connections connecting nodes and conducting analogue signals. Every basic system is an adequately configured (by adding external components) *operational amplifier*, which physically performs one simple mathematical operation, for example, summing, multiplying, comparing, differentiating or integrating.¹²

Designed for a specific purpose (which can be, for example, finding solutions of a differential equation), an AEC functions in the following way: 1) the AEC input (which is one component or more) receives a particular *function* (for example, sinusoidal) in a continuous, real time fashion; 2) the continuously performed values of the function are car-

ried to the subsequent AEC components (sometimes in a parallel way) and modified there; 3) the result (for example, the results of integrating the following “fragments” of the input function) is also successively generated in real time at the output.

At any time, the user of AEC can *measure* the output signal and obtain a single desired result (for example, an integral result), observe the functions that are generated (for example, on the screen) and interpret them systematically as some functional results (which can be, for example, the results of differential equations).

Designing AECs that solve specific problems consists in combining adequately a number of freely chosen basic circuits (such as an adder or a comparator). Thus, the “programme” which performs a defined function is in fact the physical structure of the circuit.

Although the above-mentioned example (of an electronic realisation of a model) refers to specific technical solutions that were and still are used in practice, the existence of these solutions should not be treated as an argument for the actual continuity of computations described by the GPAC model and other related models, as there arise two instances of doubt. Firstly, in every electronic circuit, the measurement of resulting values is always taken with some finite accuracy, which can unquestionably discredit the result (described theoretically as a continuous object). Secondly, it is unknown whether basic “continuous” operations, performed by, for example, operational amplifiers, are not actually, at a sufficiently low level of description, discrete operations.

3. A new face of analogicity: natural computations

One could think that the “analogous” meaning of analogicity (AN-A), discussed in section 2.2, is only related to some outdated solutions that have definitively become a thing of the past with the invention of universal digital computers. However, this is not the case.

For in contemporary computer science, the current of *natural computing*,¹³ encompassing studies on computations that are in a vital way (either on the level of data storage or rules of data processing) dependent on processes occurring in nature, becomes more and more pronounced (Rozenberg et al., 2012). Even though computations of this type are not called “analogue”, they are consistent – as we will see below – with the old, naturalistic and empirical way of understanding analogicity.

3.1. Natural computing

Applied at the meeting point of computer science and natural science, the term “natural computing” defines collectively three types of processes: A) computations *inspired* by the observation of nature (for example, evolutionary computations); B) computations *realised* with the use of natural processes and/or data carriers (for example, quantum computations); and C) processes of (presumed) computational character that *occur in nature* (for example, intracerebral processes) (Kari & Rozenberg, 2008). Therefore, it has three different meanings that are connected by a shared reference to nature.¹⁴

Another thing they all share is the notion that at least some natural processes are of computational character, explicitly expressed in the meaning C. This realisation enables a step from the meaning C towards the remaining two: if some natural processes are computational by nature, we can exploit them computationally, devising such data processing techniques that are based either on these processes (moving from the meaning C towards B) or suitable computation rules (moving from the meaning C towards A).¹⁵

Among the above-described meanings, the meaning B remains consistent with the idea of analogicity (in AN-A sense) that is discussed in this paper. If computations performed with the use of natural processes are recognised as a type of natural computations, thus they are being identified with analogue computations without the term “analogue” actually being used. It can be said so, because, in their case, the reference to nature is made in order to find not an inspiration or a rule, but a real, specific process that computes something. Therefore, this is a search for a *natural analogon* of a mathematical operation, which has already been accepted as a distinguishing feature of the AN-A meaning. It is interesting to notice, however, that the vast majority of contemporary researchers seem to disregard the fact that old analogue computations (for example, the physically performed operation of integration) are indeed natural computations. The majority of practitioners and methodologists of today focus rather on new types of techniques (e.g. quantum), not recognising their methodological resemblance to traditional solutions.

When pondering on the similarities between a certain type of natural computations and the idea of AN-A analogicity, another rather general issue should be discussed. Defining analogue computations as procedures that must be performed with the use of natural processes and/or data carriers can be faulted for *non-constructiveness*. It seems obvious that the performance of all (not only analogue) computations without exception requires

something physical; thus, it is impossible to plausibly distinguish computations of type B.

Despite the obvious character of this observation, I suggest looking at the matter more gently, paying attention to the *gradualness* of necessary references to nature. Some procedures can be performed with the absolute minimum of operations (which are, moreover, independent of the type of matter used), whereas other require considerably more operations. For example: in order to perform digital computations, it is enough to apply two distinguishable states and some very basic processes (such as: move the reader to the right or to the left), however, in the case of quantum computations (Deutsch, 1985) some very special quantum micro-circuits have to be used (if they were not taken from nature, the effect of breaking the exponential time complexity of some problems would not be achieved). In the case of biomolecular computations, the reference to nature runs even deeper. These examples prove that there is a *gradable scale* of references to nature. The closer the relationship between the effectiveness of computations and non-elementary properties of nature, the more natural a computation should be recognised.¹⁶

3.2. Two directions of extensions

If one views the discussed natural computations as contemporary *extensions* of AN-A analogue techniques, one will undoubtedly notice two directions in present research. Firstly, these are investigations set in modern *physics*, whose most tangible results seem to be, above all, quantum computations (although the quite daring ideas of infinitist hypercomputations derived from the general theory of relativity should not be forgotten; cf. (Shagrir, 2004)). Secondly, there are *biological* solutions (or referring to biology), increasingly often represented in contemporary computer science. Typical examples include, in the first place, biomolecular computations based on some qualities of DNA chains (also called DNA computations), but also more ambitious attempts at building biocomputers that process data using whole cells of living organisms.¹⁷

Thus, we can observe that the distinctness between new techniques and old analogue computations is expressed in the crucial extension of the range of references to nature. In the past, macroscopic physical systems (such as electrical circuits, see example 1) used to be the point of reference, while today they are both *micro* physical systems (i.e. quantum) and more complex *biological* systems (for example, living cells). The extension of the idea of past analogicity is indeed moving in two directions: *inwards* the traditional physical (inanimate) nature and *outwards* into the new area of biological (animate) nature.

4. Particular methodological issues

In the last chapter, I would like to highlight a few interesting issues (partly mentioned already) that emerge from the discussed distinctions and analyses. I regard these topics as an invitation to further discussion and offer the following thoughts as a brief introduction to deeper deliberations on the subject.

4.1. The empirization of computations and their reliability

The first issue is directly related to the first way of understanding analogicity (AN-A) and concerns the reliability (in a narrow sense: accuracy) of computations based on the principle of natural analogy. As I have already indicated in point 2.2., the mathematical reliability of procedures of this type (i.e. that they can efficiently be used to perform some mathematical operations) must depend on the level of *adequacy of a theory* that connects formulas and results of computations with the physical reality (more precisely: with the processes that perform these computations).

The mentioned theory – being the result of the *idealisation* procedure, typical of empirical sciences, which consists in examining phenomena while ignoring the factors that are presumed as unimportant – is never hundred percent exact (Krajewski, 1998). Thus, if the results of mathematical operations are sought directly in the reality that is described by the theory (for example, through experiment, measurement, etc.), these results must be distorted by the very same factors that have been omitted during idealisation. Metaphorically speaking: the procedure of idealisation is a double-edged sword. It allows to create a cognitively effective theory but when trying to carry out the theoretical computations by referring to (not idealised) reality, it will incorporate *mistakes* (imprecisions, errors that may result in a wrong outcome).

Performing computations with the use of empirical method requires an additional reflection on types of the theories that sufficiently *justify* the chosen method. In the light of the remarks made in point 2.2, the problem of choosing between physical and biological theories becomes prominent. That is to say: should natural computations, which are more and more frequently used, be justified only within *physics* (as it happens in the case of quantum computations and traditional analogue techniques), or is referring to *biology* equally valid?

On the one hand, there is no doubt that animate systems, examined by biologists in their natural environment, demonstrate a huge level of effec-

tiveness in solving problems (mainly adaptation problems). From the point of view of computer science, they can be treated as a “ready-to-use,” sophisticated product of natural evolution. On the other hand, biological theories are far less formalized than physical theories, hence they do not provide satisfactorily good justification of computational effectiveness of the systems used (i.e. living entities). Perhaps, in the case of computer science that is oriented towards biology, we are bound for creating solutions through the unreliable process of trial and error (which can be only partly justified by the fact that mechanisms and systems studied by biology simply work well in nature; see also (Stacewicz, 2019b)).

4.2. The physical realisation of continuous (hyper)computations

Another methodological issue is related to analogue computations in the sense of AN-C, that is continuous ones. Theoretical analyses indicate that computations of this type – described, for example, through the model of recursive real-valued functions – have the status of *hypercomputations*.¹⁸ It means that they allow solving problems that are out of reach of digital techniques formally expressed by the model of universal Turing machine (Mycka, 2010). An example of such a problem is the issue of solvability of diophantine equations (Harel, 1997).

Although the theory of continuous computations does predict that they have *higher computational power* than digital techniques, there is an important question of the practical *implementability* of continuous computations (Mycka, 2010). In other words, 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 analogue-continuous computations.

This problem leads to the question of the separateness of the mind (or even the mind/brain understood as a biological system) from the physical world, to which real digital automata belong. Perhaps the fact that the mind’s computational power is higher than the power of digital machines – which, according to some, is proven by the observed ability of the mind to intuitively solve difficult mathematical problems (Marciszewski, 2013), (Stacewicz, 2019a) – can be explained by the *continuity* of mental sphere (or even the continuity of nervous system).¹⁹

4.3. Universal analogue machines

Probably the most crucial difference between analogue computations (of both types) and digital techniques consists in the fact that only in the case of the latter, there is a *universal executive programme* that allows to correctly execute programmes provided from the outside onto a digital ma-

chine. On a mathematical level, it corresponds to the universal Turing machine (Turing, 1936).

In the case of analogue-continuous computations, there are certain models of computations defined (such as the GPAC or EAC models), however, they assume that specific specialised analogue systems have to be constructed for different problems. Even if a model defines a minimal set of computations/elementary operations (whose different combinations are enough to realise any complex computation), there remains the question of a “universal” automaton that is capable of simulating any specialised circuit (treated as a programme provided from the outside).

If such an automaton existed, it would be an analogue equivalent of the *universal Turing machine* (UTM). The input would receive a signal encoding an analogue circuit U (its structure) and an input signal S of the circuit C . On this basis, the universal automaton would perfectly simulate the operation of the circuit C for the signal S . Thus for any possible circuit C and every possible input signals/data SD , the *universal circuit* UC would generate the same resulting signal SR as the circuit C would generate for the input SD .

In the theory of analogue-continuous computations, the described circuit is not defined, therefore there arises the question whether it is at all.²⁰ If it is not, then what is the theoretical argument behind it?

As regards analogue computations of the second type, referred to in this paper as *analogous* in short, the problem of the universal machine cannot be presented in the same way as it has been done above, because such computations are, by definition, of *local* character. It is worth remembering that in their case, natural analoga to particular types of mathematical operations are used. Nonetheless, it can be inquired whether a *maximalist* physical (more broadly: natural) theory exists, on the basis of which natural analoga of all mathematical operations (or at least those that can prove useful) could be found. However the existence of such a theory – especially in the light of the conclusions formulated by K. Gödel or, in our day, G. Chaitin (2003) – is indeed truly doubtful.

N O T E S

¹ The paper is based on the following text (published in Polish): Stacewicz, P. (2017). O różnych sposobach rozumienia analogowości w informatyce (On Different Meanings of Analogicity in Computer Science. *Semina Scientiarum*, 16, 121–137.

² Cf. post in the blog Cafe Aleph entitled “Analogowość, analogiczność, ciągłość (Analogicity, analogousness, continuity),” March 9, 2015, <http://marciszewski.eu/?p=8365>.

³ In such a narrow perspective, this concerns a processing technique that is a consequent conversion of some input quantities; in a broader perspective, however, it refers to connecting processing methods with the technique of data recording (for instance on vinyl records) and their transmission (as in analogue telecommunication). Further on in the paper I will focus on the narrow perspective.

⁴ In accordance with the mentioned convention, all *data processing* techniques are linguistically identified with some *computations*: analogue, digital, continuous, natural, etc. In line with this terminology practice, the elementary field of computer science that focuses on the most basic data processing models (such as, for example, the universal Turing machine model) is called the theory of computations.

⁵ Sometimes, especially in some informal contexts, *discrete vs continuous* (the mathematical aspect) and *digital vs analogue* (the computer science's aspect) distinctions are treated as oppositions. From a formal point of view, however, we cannot talk about opposition, but complementing or extension. A continuous domain includes a discrete domain as its subset, thus a discrete domain can be extended to a continuous domain. For example, a closed interval of a form $[0, 1]$ includes a set $\{0, 1\}$, thus this set $\{0, 1\}$ can be extended to the interval $[0, 1]$.

⁶ Obviously, in the case of digital computations two opposite states, such as voltage and the lack of voltage, are sufficient.

⁷ However, it is unknown whether continuity (of, for example, real numbers) is an actual feature of the above-mentioned processes as well as their results (results of measurements). Perhaps, this is only a feature of their mathematical description. Still, it is known that measuring instruments used in practice force discreteness of the obtained results just because they have finite accuracy. We will return to these issues at the end of the text.

⁸ See for example (Costa, Graça 2003).

⁹ It should be noticed that the distinction discrete-continuous (not being an opposition; see footnote 5) has its foundations in mathematics, in which discrete and continuous objects (for example discrete and continuous sets, discrete and continuous random variables, etc.) are distinguished in a rather standard way. The basis for the definition of the former are natural numbers (N), whereas of the latter – real numbers (R). Even at the level of naming the whole fields of mathematics, it is more and more common to differentiate discrete mathematics from continuous quantity mathematics (based on analysis).

¹⁰ Models of this type should be distinguished from models of analogue-analogous computations (in the sense of AN-A), which are in fact fragments of some physical theories. These theories connect defined mathematical structures (including: a mathematical operation performed by a given system) with defined physical processes. Due to the dedicated, hence non-universal character of computations of this type, the corresponding models can be called micro-models (they are not general).

¹¹ Its methodological significance is supported by the fact that it has some valuable contemporary expansions, for example, the EAC model based on recursive real-valued functions. On the EAC model, see (Rubel, 1993).

¹² Included in every basic system, an operational amplifier is a system with two inputs and one output. An ideal operational amplifier (a theoretical circuit) amplifies the signal to the extent that equals infinity (in a real circuit, it is about maximally high amplification being an attempt at the approximation of infinity). Without any external components, an amplifier works in the following way: the output signal is produced by multiplying the first input signal by “plus infinity” and adding it to the second input signal multiplied by “minus infinity.” If we insert resistors, the “infinity” will be replaced by a concrete quantity, which can be represented by a real number. Configurations that include capacitors (capacitance) realise the process of integration and differentiation. Cf. (Kulka & Nadachowski, 1982). I would like to thank Jarosław Sokołowski, a participant of philosophical

seminars organised at the Warsaw University of Technology, for valuable information on analogue electronic circuits.

¹³ Natural computing is an example of a phenomenon that is more and more often called an “empirization of computer science”. Cf. (Polak, 2016).

¹⁴ It is worth noticing that in the case of two first types of processes the term “natural computing” is not quite natural. Although in the process of designing/implementing these computations, rules or carriers/processes present in nature are used, they are still artificial computations (created by people). For this reason, it would be more appropriate to call them semi-natural or quasi-natural.

¹⁵ It is worth noting that both C and A meanings are additionally linked by an effective methodological scheme that is used, for example, in cognitive science. It is constituted by the following research directives: 1) assume that nature is partially of computational character (in accordance with the meaning C); 2) observing nature, create new computational techniques that allow computations of type A (for example, connection computations); 3) develop these techniques (for example, mathematically) to some extent regardless of further observations; 4) use the developed techniques to describe nature once more (for example, processes inside human brain).

¹⁶ Cf. a discussion on the subject in *Cafe Aleph* blog under the entry entitled “Obliczenia naturalne i quasi-naturalne” [Natural and quasi-natural computing], March 7, 2017, <http://marciszewski.eu/?p=8745>.

¹⁷ Also, we should not forget about computations that are biologically at the level of data processing (and not carriers or processes used) – with such a statement, however, we are moving beyond the sphere of pure analogicity. They include, among others, computations performed with the use of evolutionary techniques, cellular automata or artificial neural networks.

¹⁸ In the definitional (semantic) sense, continuous computations are not equal to digital computations. This is because their constitutive feature, that is *continuity*, constitute a vital extension of one of constitutive features of traditional Turing computations, that is discreteness (see also footnote 4).

¹⁹ The importance of the argument about “the continuity of nervous system” was also noticed by Alan Turing himself while he was writing about the hypothetical superiority of biologically established mind over digital machines. Cf. (Turing, 1950).

²⁰ In case of a negative answer, analogue-continuous computations would be characterised by irremovable weakness – especially in comparison to digital computations. Although analogue circuits would provide theoretically higher computational power than digital circuits, a programmable computer could not exist in their case. For every problem or a group of problems, a separate analogue circuit would have to be designed.

R E F E R E N C E S

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

Cafe Aleph (<http://blog.marciszewski.eu/>). An academic discussion blog ran by W. Marciszewski and P. Stacewicz.

Chaitin, G. (2003). *The Limits of Mathematics*. London: Springer-Verlag.

- Costa, J. F., & Graça, D. (2003). Analog computers and recursive functions over the reals. *Journal of Complexity*, 19(5), 644–664.
- Deutsch, D. (1985). Quantum theory, the Church-Turing principle and the universal quantum computer. *Proc. Roy. Soc. Lond A* 400, 97–117.
- Fichtenholz, G. M. (1997). Rachunek różniczkowy i całkowy (Differential and Integral Calculus), vol. 2. Warszawa: PWN.
- Harel, D. (1997). *Algorithmics. The Spirit of Computing*. Reading, Massachusetts: Addison-Wesley.
- Hogarth, M. (1994). Non-Turing computers and non-Turing computability. *PSA* 1, 126–138.
- Ifrah, G. (2006). Historia powszechna cyfr (Universal History of Numbers), vol. 2. Warszawa: Wydawnictwo W.A.B.
- Kari, L., & Rozenberg, G. (2008). The many facets of natural computing. *Communications of the ACM*, 51(10), 72–83.
- Krajewski, W. (1998). Prawa nauki. Przegląd zagadnień metodologicznych i filozoficznych (Laws of Science. A Review of Methodological and Philosophical Issues). Warszawa: Książka i Wiedza.
- Kulka, Z., & Nadachowski, N. (1979). *Analogowe układy scalone (Analogue Integrated Circuits)*. Warszawa: WKiŁ.
- Kulka, Z., & Nadachowski, N. (1982). *Wzmacniacze operacyjne i ich zastosowanie (Operational Amplifiers and Their Applications)*. Warszawa: WKiŁ.
- Marciszewski, W. (2013). *Racjonalistyczny optymizm poznawczy w Gödłowskiej wizji dynamiki wiedzy (Rationalistic Cognitive Optimism in Gödel's Vision of Dynamics of Knowledge)*. In R. Ziemińska (Ed.), *Przewodnik po epistemologii (A Companion to Epistemology)* (pp. 423–456). Kraków: Wydawnictwo WAM.
- Marciszewski, W., & Stacewicz, P. (2011). *Umysł – Komputer – Świat. O zagadce umysłu z informatycznego punktu widzenia (Mind – Computer – World. On the Riddle of the Mind from the Computational Point of View)*. Warszawa: Akademicka Oficyna Wydawnicza.
- Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin, Heidelberg: Springer-Verlag.
- Moore, C. (1996). Recursion theory on the reals and continuous-time computation. *Theoretical Computer Science*, 162, 23–44.
- Mycka, J. (2010). Obliczenia dyskretne i ciągłe jako realizacje antropomorficznej i fizycznej koncepcji efektywnej obliczalności (Discrete and continuous computations as realisations of an anthropomorphic and physical concept of effective computability). In I. Bondecka-Krzykowska & J. Pogonowski (Eds.), *Światy matematyki. Tworzenie czy odkrywanie? (The worlds of mathematics. Creating or discovering?)* pp. 247–260). Poznań: Wydawnictwo Naukowe UAM.

- Mycka, J., & Piekarz, M. (2004). Przegląd zagadnień obliczalności analogowej (A Review of the Issues of Analog Computability). In S. Grzegórski, M. Miłosz, & P. Muryjas (Eds.), *Algorytmy, metody i programy naukowe (Algorithms, Methods and Scientific Programmes)* (pp. 125–132). Lublin: Polskie Towarzystwo Informatyczne.
- Pour-El, M. B. (1974). Abstract Computability and its Relations to the General Purpose Analog Computer. *Transactions of the American Mathematical Society*, 199, 1–28.
- Polak, P. (2016). Computing as Empirical Science-Evolution of a Concept. *Studies in Logic, Grammar and Rhetoric*, 48(1), 49–69.
- Rozenberg, G., Back, T., & Kok, J. N. (Eds.). (2012). *Handbook of Natural Computing*. Berlin-Heidelberg: Springer-Verlag.
- Rubel, L. (1993). The extended analog computer. *Advances in Applied Mathematics*, 14, 39–50.
- Shannon, C. (1941). Mathematical Theory of the Differential Analyzer. *J. Math. Phys. MIT*, 20, 337–354.
- Shagrir, O. (2004). Super-tasks, Accelerating Turing Machines and Uncomputability. *Theoretical Computer Science*, 317, 105–114.
- Sieglmann, H. T. (1998). *Neural Networks and Analog Computation: Beyond the Turing Limit*. Boston: Birkhauser.
- Stacewicz, P. (2010). Umysł a modele maszyn uczących się, współczesne badania informatyczne w oczach filozofia (The Mind and Models of Learning Machines. Contemporary Computer Science Research from Philosophical Perspective). Warszawa: Akademicka Oficyna Wydawnicza.
- Stacewicz, P. (2017). O różnych sposobach rozumienia analogowości w informatyce (On Different Meanings of Analogicity in Computer Science). *Semina Scientiarum*, 16, 121–137.
- Stacewicz, P. (2019a). From Computer Science to the Informational Worldview. Philosophical Interpretations of Some Computer Science Concepts. *Foundations of Computing & Decision Sciences*, 44, 27–43.
- Stacewicz, P. (2019b). Uncomputable Numbers and the Limits of Coding in Computer Science. *Studia Semiotyczne*, 30, 107–127.
- Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, 49, 433–460.
- Turing, A. M. (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.*, 42, 230–265.