

Toward a formal theory for computing machines made out of whatever physics offers

Received: 9 October 2022

Accepted: 1 August 2023

Published online: 16 August 2023

 Check for updatesHerbert Jaeger ^{1,2} ✉, Beatriz Noheda ^{2,3} & Wilfred G. van der Wiel ^{4,5,6}

Approaching limitations of digital computing technologies have spurred research in neuromorphic and other unconventional approaches to computing. Here we argue that if we want to engineer unconventional computing systems in a systematic way, we need guidance from a formal theory that is different from the classical symbolic-algorithmic Turing machine theory. We propose a general strategy for developing such a theory, and within that general view, a specific approach that we call fluent computing. In contrast to Turing, who modeled computing processes from a top-down perspective as symbolic reasoning, we adopt the scientific paradigm of physics and model physical computing systems bottom-up by formalizing what can ultimately be measured in a physical computing system. This leads to an understanding of computing as the structuring of processes, while classical models of computing systems describe the processing of structures.

Digital computing technologies are accelerating into a narrowing lane with regard to energy footprint¹, toxic waste², limits of miniaturization³ and vulnerabilities of ever-growing software complexity⁴. These challenges have spurred explorations of alternatives to digital technologies. A fundamental alternative is neuromorphic computing⁵, where the strategy is to use biological brains as role model for energy-efficient parallel algorithms and novel kinds of microchips. We also see a reinvigorated study of other unconventional computing paradigms that search for computational exploits in a wide range of non-digital substrates like analogue electronics⁶, optics⁷, physical reservoir computing systems⁸, DNA reactors^{9,10}, or chemical reaction-diffusion processes¹¹. These approaches have become branded under names like natural computing, physical computing, in-materio (or in-material¹²) computing, emergent computation, reservoir computing and many more^{13–15}. These lines of study can be seen as belonging together in that they typically are interested in self-organization, energy efficiency, noise robustness, adaptability, statistical dynamics in large ensembles—foci that set these approaches apart from

quantum computing, a field that we rather see as a variant of digital computing¹⁶ and do not further address in this article.

A key objective in these fields is to understand how, given novel sorts of hardware systems made from “intelligent matter”¹⁷, one can “exploit the physics of its material directly for realizing its operations”¹⁸. A salient example is the realization of synaptic weights in neuromorphic microchips through memristive devices¹⁹. In digital simulations of neural networks, updating the numeric effect of a synaptic weight on a neuron activation needs hundreds of transistor switching events. In contrast, when a neural network is realized in a physical memristive crossbar array²⁰, one obtains an equivalent functionality through a single voltage pulse applied across the corresponding memristive synapse element.

This principle of direct physical mirroring is not limited to updating single numerical quantities. Complex information-processing operations—like random search in some state space, graph transformations, finding minima in cost landscapes, etc.—can be encoded in terms of complex spatiotemporal physical phenomena in

¹Bernoulli Institute, University of Groningen, 9700 AB Groningen, The Netherlands. ²Groningen Cognitive Systems and Materials Center (CogniGron), University of Groningen, 9700 AB Groningen, The Netherlands. ³Zernike Institute for Advanced Materials, University of Groningen, 9700 AB Groningen, The Netherlands. ⁴BRAINS Center for Brain-Inspired Nano Systems, University of Twente, 7500 AE Enschede, The Netherlands. ⁵MESA+ Institute for Nanotechnology, University of Twente, 7500 AE Enschede, The Netherlands. ⁶Institute of Physics, Westfälische Wilhelms-Universität Münster, Münster, Germany.

✉ e-mail: h.jaeger@rug.nl

many ways, for instance in oscillations, chaos and other attractor-like phenomena; hysteresis; many sorts of bifurcations and input-induced transits between basins of attraction; spatiotemporal pattern formation; intrinsic noise; phase transitions. In turn, these operations and their spatiotemporal encodings can be used to serve complex cognitive functions like problem solving, attention or memory management. The pertinent literature is so extensive that it defies a systematic survey. Other phenomena and their mathematical reconstructions are more specific, for instance heteroclinic channels and attractor relics^{21,22}, self-organized criticality^{23–25} or solitons and waves^{26,27}. Fabricating materials with atomic precision is today routinely done worldwide, exploring optical, mechanical, magnetic, spintronic or quantum effects and their combinations, for instance in nanowire networks^{24,28} or skyrmion-based reservoir computing²⁹. Physical materials and devices offer virtually limitless resources of physical phenomena for building unconventional computing machines. In contrast to the perfect precision of circuit layouts needed for digital hardware, many usable physical phenomena thrive on disorder and can self-organize into functional behaviour^{30,31}. An illustrative example from our own work is presented in Box 1.

It remains, however, unclear how one can make best use of these opportunities. We are lacking a general theory framework that would

inform us what specific sorts of computations can be served by different physical phenomena. Under the influence of cybernetics, computational neuroscience, systems biology, machine learning, robotics, artificial life and unconventional computing, our intuitions about computing have broadened far beyond the digital paradigm. Information processing in natural and artificial systems has been alternatively conceptualized and formalized in terms of analogue numerical operations^{6,32,33}, probabilistic combinations of bit streams^{34,35}, signal processing and control^{36–40}, self-organized pattern formation^{11,26,41,42}, nonlinear neural dynamics^{21,43,44}, stochastic search and optimization^{9,45}, evolutionary optimization^{46–48}, dynamics on networks^{49,50} or statistical inference^{51–53}. From this widened perspective, we investigate the need for, and the chances of, a formal theory for computing systems that directly exploit physical phenomena.

The algorithmic and cybernetic modes of computing

One might think that we already have a theory of general computing systems, namely what a Turing machine can do. Even philosophers, when they try to come to terms with the essence of computing, invariably orient their argumentation toward Turing computability^{54,55}. The mathematical theory of Turing computability constitutes the heart of modern computer science, unleashing a technological revolution on

BOX 1

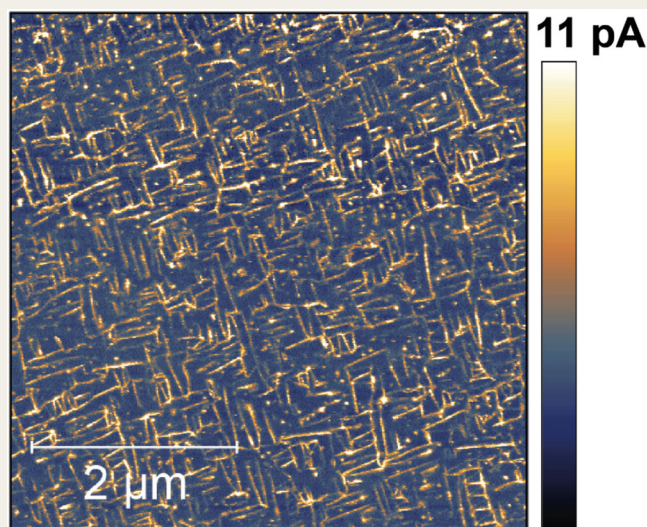
Complex spatiotemporal phenomena in thin film materials

One of us investigates ferroelectric and ferromagnetic effects in novel computational materials. These materials display an ordered phase, which is responsible for long-term bi-stability, and a disordered phase, in which these properties vanish. Ordering across macroscopic distances gives rise to strongly nonlinear responses to external stimuli. The complexity and sensitivity to external stimuli is maximized at phase transitions^{113,114}. Some of the novel materials that we synthesize combine multiple types of interactions (magnetic, electrical, mechanical, chemical) and thus display complex phase diagrams with multiple available phases. Using state-of-the-art thin-film deposition techniques, we can make materials that persist permanently at the edge between two phases, or close enough to a phase transition¹¹⁵, such that they can be brought from one phase to the other with low-energy external stimuli¹¹⁶.

The image (obtained using conducting atomic force microscopy at room temperature, taken from¹¹⁷) shows a network of conducting domain walls in a ferroelectric BiFeO₃ thin layer. The colour coding in

picoampere (pA) indicates the current measured vertically through the film to the ground plate at 3 V probing voltage. The domain walls (yellow) have a much higher conductivity than the regions in between the walls. The areas in between (blue) are insulating ferroelectric domains that can be switched into non-volatile memory states. Structures like this might become used, for example, to encode and dynamically switch the large random bit vectors which are the main representational elements in the hyperdimensional computing paradigm³⁵.

The properties of these materials—fine-grained conductivity pathways, local multi-stability with resistive properties that are switchable with minimal energy, many timescales, hierarchical topological structuring—hold many promises for computation in materials. Spatiotemporal processes in a regime close to criticality near a phase transition have been described as an enabling condition for complex information processing dynamics^{23,25}.



a par with the invention of the wheel. The Turing machine is simulation-universal in the sense that every physical system whose defining equations are known can be simulated on a Turing machine. Given these powers of the Turing paradigm, why should one wish to develop a separate theory for computing with general physical phenomena at all? What could such a theory give us that Turing theory cannot deliver?

We begin with the historical roots of the Turing machine concept. By inventing this formal concept, Turing set the capstone on two millennia of inquiry which started from Aristotle’s syllogistic logic and continued through an uninterrupted lineage of scholars like Leibniz, Boole, Frege, Hilbert and early 20th century logicians. The original question asked by Aristotle—what makes rhetoric argumentation irrefutable?—ultimately condensed in the *Entscheidungsproblem* of formal logic: is there an effective formal method which, when given any mathematical conjecture as input, will automatically construct a proof for the conjecture if it is true, and a counterproof if it is false? While all the pre-Turing work in philosophy, logic and mathematics had finalized the formal definitions of what are conjectures, formal truth, and proofs, it remained for Alan Turing (and Alonso Church⁵⁶) to establish a formal definition of what is an effective method for finding proofs. Turing’s formal model of a general mathematical proof-searching automatism is the Turing machine.

In his ground-breaking article *On computable numbers, with an application to the Entscheidungsproblem*⁵⁷, Turing modelled proof-finding mechanisms as an abstraction of a human mathematician doing formal calculations with paper and pencil. The Turing machine consists of a tape on which a stepwise moving cursor may read and write symbols, with all these actions being determined by a finite-state switching control unit. The tape of a Turing machine models the sheet of paper used by the mathematician (a male in Turing’s writing), the machine’s read/write cursor models his eyes and hands, and the finite-state control unit models his thinking acts. Importantly, Turing speaks of “states of mind” when he refers to the switching states of the control unit—not of physiological brain states. The Turing machine models reasoning processes in the abstract sphere of mathematical logic, not in neural electrochemistry. Students of computer science must do coursework in formal logic, not physiology; and their theory textbooks speak a lot about logical inference steps, but never mention seconds. When one understands the Turing machine as a general model of rational human reasoning, it becomes clear why computers can be simulation-universal: everything that physicists can think about with formal precision can be simulated on digital computers, because they can simulate the physicist’s formal reasoning.

The royal guide for shaping intuitions about the physical basis of computing are biological brains, especially human brains. We will now take a closer look at what aspects of physical neural dynamics are not captured by Turing computability.

We first admit that one specific aspect of brain-based information processing is indeed homologous to Turing computing. Humans can carry out systematic, logically consistent sequences of arguments and planning steps. Let us call this the algorithmic mode of computing (AC). After all, Turing shaped his model of computing after a human mathematician’s reasoning activity. However, our biological brains fall short of the mathematical perfection of Turing machines in many ways. Our ability to construct nested symbolic structures is limited⁵⁸; the concepts that our thinking is made of are not as clean-cut and immutable as mathematical symbols are, but graded, context-varying and incessantly adapting^{59,60}; our reasoning often uses non-logical, analogy-based operations⁶¹. The Turing machine is a bold abstraction of a particular aspect of a human brain’s operation.

Most of the time, however, most parts of our brain are not busy with logical reasoning. In our lives, most of the time we do things like walking from the kitchen table to the refrigerator, without clean logical

thinking. Yet, while walking to the refrigerator, the walker’s nervous systems is thoroughly busy with the continual processing of a massive stream of sensor signals, smoothly transforming that input deluge into finely tuned, uninterrupted signals to hundreds of muscles. Let us call this sensorimotor flow of information processing the cybernetic mode of computing (CC). For the largest part of biological history, evolution has been optimizing brains for cybernetic processing—for “prerational intelligence”⁶². Only very late, some animals’ brains acquired the additional ability to detach themselves from the immersive sensorimotor flow and generate logico-symbolic reasoning chains. Several schools of thinking in philosophy, cognitive science, AI and linguistics explain how this ability could develop seamlessly from the cybernetic mode of neural processing, possibly together with the emergence of language^{63–68}.

Aligning and contrasting algorithmic and cybernetic computing

In order to get a clearer view on these two modes of computing, we place two schematic processing architectures side by side (Fig. 1). Each of them is shown with three levels of modeling granularity, from a fine-grained machine interface level $L^{(1)}$ via an intermediate level $L^{(2)}$ to the most abstract task specification level $L^{(3)}$. To make our discussions concrete, we exemplify them with the elementary algorithmic task of multiplying 6 with 5 on a pocket calculator⁶⁹, and the paradigmatic cybernetic task of regulating the speed of a steam engine with a centrifugal governor^{70,71}.

We begin with the algorithmic dissection of the multiplication task. Digital computations are modelled (or, equivalently, programmed) by breaking down a user’s task description through a cascade of increasingly finer-granular formalisms down to a low-level machine interface formalism—think of compiling a program written in the task-level machine learning toolbox TensorFlow⁷² down to machine-specific assembler code, passing through a series of programs written in languages of intermediate abstraction like Python or C/C++. In our arithmetics example we condense this to three modeling levels (Fig. 1a–c). On the top level $L^{(3)}$, the user specifies task instances by typing the input string $u^{(3)} = 6 * 5$, or $17 * 4$ etc. In our graphics the structure of this input is rendered by the cells in the vertical state bars. The user knows that there is a mathematical function $f^{(3)}$ (the multiplication of integers) that is evaluated by the calculator, but the user is not further concerned about how, exactly, this computation is effected. This is taken care of by the designer of the calculator, who must hierarchically disassemble the input $6 * 5$ down to a machine-interface level $L^{(1)}$.

On an intermediate modeling level $L^{(2)}$ one might encode the original input $u^{(3)} = 6 * 5$ in a binary representation (for instance $6 \rightarrow 110$, $5 \rightarrow 101$, $*$ \rightarrow `binarymult` (leftmost green cell bars $u^{(2)}$ in b)). This new input encoding is then processed stepwise with functions $f^{(2)}$ (green arrows), possibly in parallel threads, through sequences of intermediate binary representations $x_i^{(2)}$, until some binary string representation $y^{(2)}$ of the result is obtained, which then can be decoded into the top-level representation $y^{(3)} = 30$. One of these functions $f^{(2)}$ could for instance be an addition operation $x_i^{(2)} + x_j^{(2)} = x_k^{(2)}$, like `binaryadd(110, 100) = 1010` (inner grey rectangle).

In a final compilation, this instance of the `binaryadd` operation might become encoded in a sequence of transformations of 8-bit strings (bytes), whose outcome $y^{(1)}$ decodes to $x_k^{(1)} = 1010$. This encoding format can be directly mapped to the digital hardware by an experienced engineer. In our schematic diagram we assume that this level specifies binary (Boolean) functions $f^{(1)}$ (violet arrows in c) between 8-bit re-writable register models (vertical violet bars).

We now turn to our view on hierarchical models of cybernetic information processing systems (Fig. 1d–f). On all modeling levels $L^{(m)}$, inputs $u^{(m)}$ and outputs $y^{(m)}$ are signal streams that are continually received or emitted. These signals can be composed from subsignals—think of a robot’s overall sensory input which might comprise

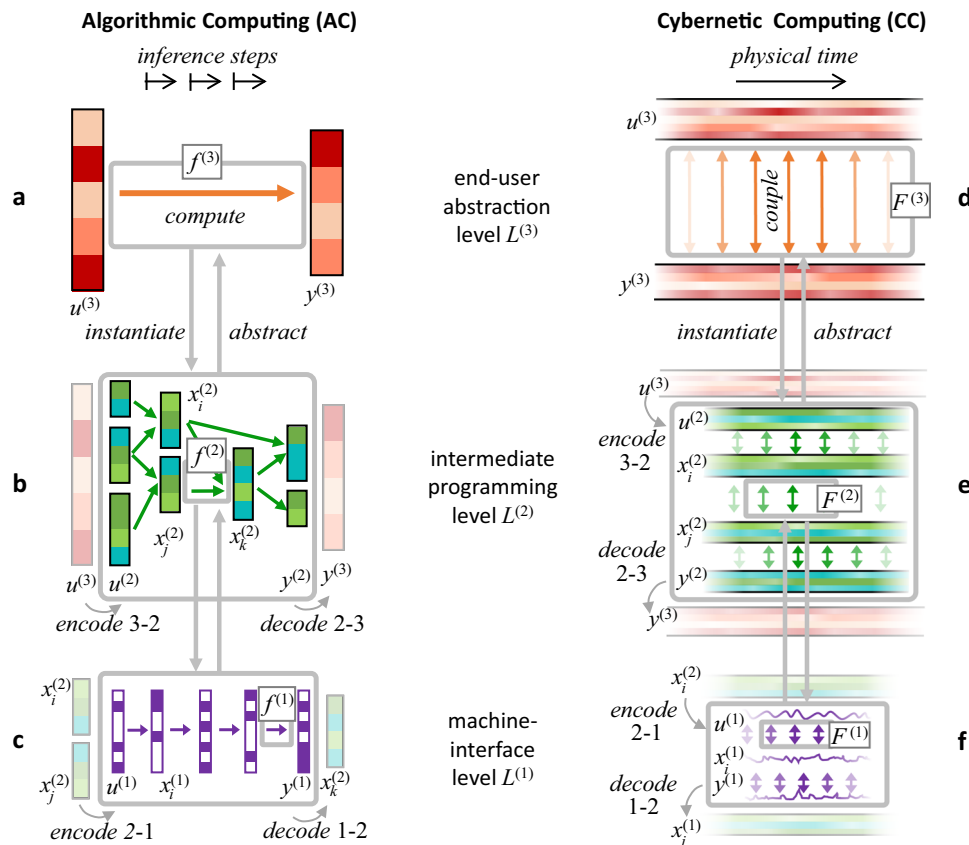


Fig. 1 | Modeling an algorithmic and a cybernetic computing system on three levels of abstraction. **a–c** Digital computing systems are typically modelled as algorithmic. The overall functionality of such a system is to transform input data structures $u^{(3)}$ into output data structures $y^{(3)}$. An algorithmic model thus represents, on the global task level, a mathematical function $f^{(3)}$ from inputs to outputs (**a**). To model and implement how this global task function is realized on a digital computing machine, it is stepwise broken down (compiled) into finer-grained models, where data structures (vertical colour bars) and functions (arrows) become hierarchically dissected until at a machine-interface level (**c**), both can be

straightforwardly mapped to the digital circuits of the underlying microchip. **d–f** Cybernetic computing systems, like brains or analogue processing chips for signal processing and control, transform a continually arriving input signal $u^{(3)}$ into an output signal $y^{(3)}$ by a continually ongoing nonlinear dynamical coupling $F^{(3)}$. Like digital data structures and processes, signals and their couplings can be hierarchically broken down from a global task-level specification (**d**) to machine-interfacing detail (**f**). Symbols x represent intermediate data structures (in algorithmic models) and signals (in cybernetic models). Three modeling levels are shown, but there may be more or fewer.

subsignals from cameras, touch sensors, the battery and joint angles. This multi-subsignal makeup is reflected in Fig. 1d–f by the stripes inside the $u^{(3)}$, $y^{(3)}$, $u^{(2)}$, $x_i^{(2)}$ and $y^{(2)}$ bands. The time-varying subsignal values, which we call their activations, are indicated by changing colour intensity. The decomposition into subsignals may be hierarchically continued. In Fig. 1 only the first-level subsignal decomposition is shown.

At the lowest machine-interface level $L^{(1)}$ (**f**), input/output signals $u^{(1)}$ and $y^{(1)}$ as well as intermediate processing signals $x_i^{(1)}$ are modelled as evolving in real time $t \in \mathbb{R}$. At higher modeling levels, more abstract mathematical models t of temporal progression may be used, allowing for increasing uncertainty about precise temporal localization (detailed in Section 2.4 in⁷³). We reserve the word signal for one-dimensional, real-time signals, and use the word chronicles to refer to possibly multi-modal signals that are formalized with possibly abstract time models t .

In our steam engine governor example, the highest-level input chronicle $u^{(3)}$ could be composed of the measured current engine speed $s^{(3)}(t)$ and the desired speed $d^{(3)}(t)$. The output chronicle $y^{(3)}$ is the steam valve setting signal $p^{(3)}(t)$. The input and output chronicles are continually connected by a coupling law $F^{(3)}$, say by the simple proportional control rule $\dot{p}^{(3)} = K(d^{(3)} - s^{(3)})$. While in this example the coupling is unidirectional from $u^{(3)}$ to $y^{(3)}$, in general we admit bidirectional couplings. In the terminology of signals and systems, we admit

autoregressive filters for coupling laws. We note that the refference principle in neuroscience⁷⁴ stipulates that output copy feedback is common in biological neural systems.

An intermediate $L^{(2)}$ -model would capture the principal structure and dynamics of the governor, using chronicles that monitor speeds, forces, angles etc. of system components like masses, levers, axes, joints etc.

On the lowest level $L^{(1)}$, the dynamical couplings between $L^{(2)}$ -chronicles are concretized to match the specific design of an individual physical governor. For instance, a coupling $F^{(2)}$ between a centrifugal force and a compensating gravitational load force, which would presumably be formalized in level $L^{(2)}$ by a differential equation, would be resolved to the metric positioning of joints on lever arms, the weights and sizing and strengths of mechanical parts, etc., leading to fine-grained signals $u^{(1)}$, $x^{(1)}$, $y^{(1)}$ like the current force or velocity components on a specific joint, or—in a high-precision model—measures of temperature or vibration which have an impact on the part's functioning.

The algorithmic–cybernetic distinction is not a clear-cut either-or division. An intermediate view on computing is adopted in models of analogue computing^{6,32,33,75}. They follow the AC paradigm in that a high-level function evaluation task is hierarchically broken down into lower-level flowchart diagrams of sequential function evaluations exactly like in our diagrams Fig. 1a–c. At the same time, they also

appear as cybernetic in that their data structures are composed of analogue real numbers, and the functions $f^{(m)}$ are evaluated through the continuous-time, coupled evolution of differential operators. Furthermore, the input-output theory of algorithmic computing has been variously extended to sequential processing models, for instance through the concept of interactive computing⁷⁶ and (symbolic) stream computing⁷⁷.

The major similarities and differences between algorithmic and cybernetic models of information processing systems, so far as we have identified them until now, are summarized in Box 2.

The core challenge for a theory of physical computing

Working out the general intuitions about modeling computing systems into a concrete theory of physical computing must meet two conflicting demands. First, such a theory must be able to model systems that can exploit a rich diversity of physical phenomena. The ultimate goal is to model systems that are as complex as brains (but might be organized quite differently). Second, such a theory must give transparent guidance to engineers and users of computing systems. The enormity of this double challenge becomes clear when we contemplate how theoretical neuroscience models brains (trying to

BOX 2

How algorithmic computing is like, and unlike, cybernetic computing

Online, real-time, brain-like, cybernetic transformations from input signal streams to output signal streams is in some ways similar to, and in other ways fundamentally different from algorithmic (digital, symbolic) computing (AC). The similarities justify to classify cybernetic computing (CC) as computing in the first place, while the differences mandate the development of a new formal theory of physical computing. In this box we summarize basic similarities and differences.

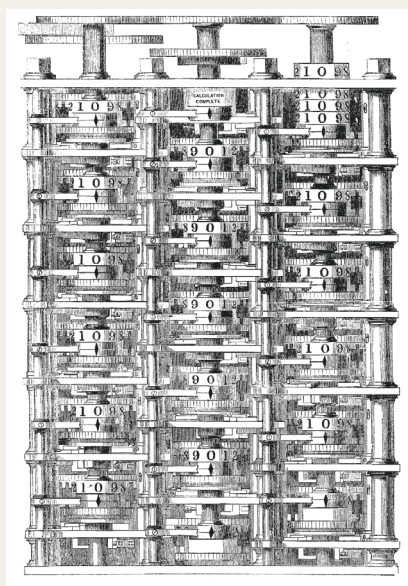
AC and CC are both based on compositional data models. All data items of AC, from integers to spreadsheet tables, come in the form of hierarchically organized, static symbol structures. The information-carrying, continuously varying signals in CC, which we call chronicles, are likewise hierarchically structured into subchronicles and sub-subchronicles, etc.

Static and discrete versus temporal and graded data models. The symbolic data structures in AC, once assembled in a computation, stay immutably the same until they are discretely updated to a new structure by a logical operation. Chronicles and their sub-chronicles

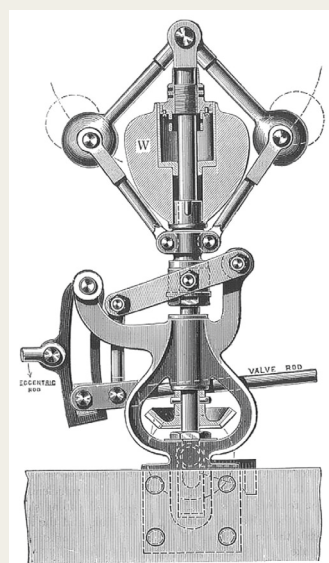
have continuously varying signal strengths (called activations) and may fade in and out.

AC as well as CC are based on hierarchical system models. Designing and using complex computing systems requires theory-guided methods for breaking down a user's input-output task specification to a formal model that maps to a machine's hardware. This is achieved by an abstraction hierarchy of computational models, each of which has its own formal concepts.

Information processing by stepwise logical inference steps versus continual dynamical coupling. In an AC computational model, the transformation from input to output data structures occurs in sequences of logical update steps. In contrast, the dynamical coupling between input and output chronicles is inherently parallel and tied to continuous physical time on both ends of the system modeling hierarchy, through the external task source signals and the internal physical dynamics of the machine. In mathematical terms, an AC computation is the evaluation of a function, while a CC computation is the evolution of a dynamical system.



The paradigmatic algorithmic computing machine: Charles Babbage's difference engine No. 1 (Harper's new monthly magazine, December 1864)



The paradigmatic cybernetic information processing machine: the centrifugal governor (W. Ripper: Heat Engines. Longman, London 1909)

address the phenomenal diversity demand), and how theoretical computer science models symbolic computation (succeeding in mastering the transparent engineering demand).

We begin with neuroscience modeling. Biological evolution is apt to find and exploit any physiological-anatomical mechanism that adds competitive advantage. Brains appear as “giant ‘bags of tricks’” which integrate “a huge diversity of specialized and baroque mechanisms”⁷⁸ into a functional whole. Neuroscientists attempt to understand brains on increasingly abstract and integrative modeling levels⁷⁹, from the microscopic biochemistry of synapses to global neural architectures needed for learning navigation maps. Explaining how the phenomena described on some level of abstraction arise from the finer-grained dynamics characterized on the level below often amounts to major scientific innovations. For instance, the Hodgkin-Huxley model of a neuron⁸⁰ abstracts from a modeling layer of electrochemical processes and calls upon mathematical tools from electrical circuit theory; while on the next level of small neural circuits, collective voting phenomena may be explained by abstracting Hodgkin-Huxley neurons to leaky-integrator point neurons and using tools from nonlinear dynamical systems⁸¹. These ad hoc examples illustrate a general condition in theoretical neuroscience: the price that is paid for trying to understand the phenomenal richness of brains is a diversity of modeling methods.

In contrast, multilevel hierarchical modeling of digital computing processes is done with one single background theory that covers all phenomena within any modeling level as well as the exact translations between adjacent levels. This theory is mathematically rigorous, fits in a single textbook⁸², and lets an end-user of a pocket calculator be assured that their understanding of arithmetic becomes exactly realized by the bit-switching mechanics of their amazing little machine. The price paid is that digital machines can exploit only a single kind of physical phenomenon, namely bistable switching—a constraint that can be seen as the root cause for the problematic energy footprint of digital technologies.

We are certainly not the first ones to take up the challenge of modeling and engineering complex, hierarchically structured cybernetic computing systems. However, we find that none of the proposals that we are aware of fully meets the twofold demand of openness to a broad spectrum of physical mechanisms and unifying engineering transparency across modeling levels. We list four instructive examples. The Neural Engineering Framework⁸³, originally developed by Chris Eliasmith and Charles Anderson and used in a sizeable community of cognitive neuroscientists^{84–88}, provides mathematical analyses and design rules for modules of spiking neural networks which realize signal processing filters that are specified by ordinary differential equations. This framework does not provide methods for hierarchical model abstraction, and the range of supporting dynamical phenomena is restricted to spiking neurons. Youhui Zhang et al. present a method for engineering brain-inspired computing systems⁸⁹, programming them in a high-level formal design language, which is compiled down through an intermediate formalism to a machine-interface level, which can be mapped to the current most powerful neuromorphic microprocessors. This approach is motivated by practical systems engineering goals and in many ways follows the role model of AC compilation hierarchies. It is however limited to exactly the three specific modeling levels specified in this work, with different principles used for the respective encodings, and at the bottom end exclusively targets digitally programmable spiking neurochips. The Realtime Control System³⁷ of James Albus is a design scheme for control architectures of autonomous robotic systems, from the sensor-motor interface level to high-level knowledge-based planning and decision making. Like other models of modular cognitive architectures⁹⁰, it is taken for granted that they are simulated on digital computers. Exploiting general physical phenomena has not been a motivation for their inception. Johan Kwisthout considers Turing machines, which

upon presentation of a task input automatically construct a formal model of a spiking neural network that can process this task, and investigates the combined consumption of computational resources for such twin systems⁹¹. For the neural network model, he allows unconventional resource categories like the number of used spikes. This work makes spiking neural networks accessible to the classical theory of computational complexity, but does not specify how the neural networks spawned by the Turing machine are concretely designed, and the approach is only applicable to a specific formal model of neural networks, not to general physical computing systems.

We are not alone with our impression that we still lack a unifying theory for neuromorphic or unconventional computing. In a 1948 lecture, John von Neumann analysed how computations in brains differ from digital computing. Focusing on stochasticity and error tolerance, he concluded that “we are very far from possessing a theory of automata which deserves that name...”⁹². Similar judgements can be found in the contemporary literature: “The ultimate goal would be a unified domain of all forms of computation, in as far as is possible...”¹³; “As the domain of computer science grows, as one computational model no longer fits all, its true nature is being revealed... new computational theories ... could then help us understand the physical world around us”⁹³; “there is still a gap in defining abstractions for using neuromorphic computers more broadly”⁹⁴; “The neuromorphic community ... lacks a focus. [...] We need holistic and concurrent design across the whole stack [...] to ensure as full an integration of bio-inspired principles into hardware as possible”⁹⁵.

In the remainder of this perspective we outline our strategy for developing a formal theory of physical computing. Our aim is to reconcile the two seemingly conflicting modeling demands of capturing general physical systems with their open-ended phenomenology on the one hand, and of enabling practical system engineering on the other. Our strategy is to merge modeling principles that originate in algorithmic and in cybernetic modeling, respectively. From AC we adopt the hierarchically compositional structuring of data structures and processes, which is a crucial enabler for systematic engineering. From CC we take the perspective to view computing systems as continually operating dynamical systems, which enables us to model information processing as the evolution of a physical system. Our key rationale for working out this strategy is to start from physical dynamical phenomena and model computing systems in a hierarchy of increasingly abstracted dynamical systems models, starting from a physics-interfacing modeling level $L^{(1)}$. Our name for such formal model hierarchies is fluent computing (FC). This naming is motivated by Isaac Newton’s wording, who called continuously varying quantities “fluentes” in his (Latin) treatise⁹⁶ on calculus.

A phenomenon is what can be observed

All theories of physics are about phenomena that can be observed (measured, detected, sensed)—at least in principle, and possibly indirectly. In the followship of physics, we want to set up our FC theory such that the values of its state variables can be understood as results of observations (measurements, recordings), and the mathematical objects represented by the variables as observers. We mention in passing that this is the same modeling idea that lies at the basis of probability theory, where random variables are construed as observers of events, too. An observer responds to the incoming signals by creating a response signal. For instance, an old-fashioned voltmeter responds to a voltage input by a motion of the indicator needle. Following the cybernetic view in this regard, we cast observing as a temporal process whose collected observation responses are timeseries objects (chronicles). The time axis of these timeseries may be formalized with modes of progression t that abstract from physical time $t \in \mathbb{R}$. Above we distinguished between input, intermediate, and output variables $u^{(m)}, x_i^{(m)}, y^{(m)}$. For convenience we will henceforth use

the generic symbol $\nu^{(m)}$ for any of them. We will speak of observers when we mean the objects denoted by model variables $\nu^{(m)}$.

Interpreting model variables $\nu^{(m)}$ as formal representatives of observers is our key guide for the design of FC model hierarchies, which model a physical computing system in an abstraction sequence that starts with a model at a machine-interface level $L^{(1)}$ and ends with a model at a high abstraction level $L^{(k)}$, which is suitable to express task-specific conditions (as in Fig. 1). A detailed workout of our FC modeling proposal is documented in the long version of this perspective¹⁶. Here we give a summary of our main ideas.

An observer $\nu^{(m)}$ reacts to a specific kind of stimulus with an activation response (think of the readings of a voltmeter or the activation of a visual feature-detecting neuron). This activation $a_{\nu^{(m)}}$ may continuously change in time. We admit only positive or zero activation (no negative activations), following the leads of biology (neurons cannot be negatively activated; they can only be inhibited toward zero activation) and the intuition of interpreting activation as signal energy (energy is non-negative). We foresee that relaxed models a of real-number activations a will be needed, with the latter possibly being appropriate only in the physics-interfacing modeling level $L^{(1)}$. The general format of an activation at some time at some modeling level would thus be $a_{\nu^{(m)}}(t)$.

The specific stimulus part is harder to grasp. We call the specific kind of stimulus to which the observer is responsive, the quality of the observer. However, one cannot exhaustively characterize what a measurement apparatus responds to. Consider a thermometer. While a thermometer is engineered to specifically react to temperature, it will also be sensitive to other physical effects. For instance, it may also react (if only slightly) to ambient pressure, vibration or radiation. In neuroscience, attempts to characterize what exactly a neuron in a brain's sensory processing pathways responds to remains a conundrum⁹⁷. We do not want to become entangled in this question. Whatever an observer reacts to, we will call the quality of the observer, and we specify this quality by specifying the observer itself. While the activation value of an observer varies in time, its defining quality is unchangeable.

Observers can be composed of sub-observers, and sub-observers can again be compositional objects, etc. For example, a retina can be defined to be composed of its photoreceptor cells, or a safety warning sensor on a fuel tank might be combined from a pressure and a temperature sensor. In our proposed FC terminology, we say that component observers are bound in a composite observer. A plausible binding operator for retina observers would bind the photoreceptor cells through a specification of their spatial arrangement, while the pressure and temperature sensor values might be bound together by multiplication. Many mathematical operations may serve as binding operators. We mention that organizing complex systems models through hierarchically nested subsystem compositions has been a core rationale in complex systems modeling⁹⁸ from the beginnings of that field⁹⁹; that composition hierarchies for data structures and processes are constitutive for AC theory as well as in systems engineering and control^{37,100,101} and the mathematical theory of multi-scale dynamical systems^{40,42,102}; and that the so-called binding problem is a core challenge for understanding how cognition arises from neural interactions^{100,103–106}. Let us denote the set of observers that are (direct or deeper-nested) components of $\nu^{(m)}$ by $B_{\nu^{(m)}}^{\downarrow}$, and the set of observers of which $\nu^{(m)}$ is a component by $B_{\nu^{(m)}}^{\uparrow}$.

Compositional observer-observee hierarchies, where composite observers become tied together by formal binding operators, are the main structuring principle for FC models. They are what we carry over from AC modeling and add to CC modeling. Punchline: AC modeling is based on processing structures, while in FC modeling we structure processes.

Observers can have memory. Their current activation response may depend on the history of what they have observed before. In

simple cases, this amounts to some degree of latency needed before the observer's response settles. In more complex cases, the current activation response can result from an involved long-term integration of earlier signal input—at an extreme end, think of a human who, while reading a novel (observing the text signal), integrates what is being related in the story with their previous life experiences. There are several mathematical ways to capture memory effects. We opt for endowing observers $\nu^{(m)}$ with an internal memory state $\mathbf{s}_{\nu^{(m)}}(t)$, which evolves through a state update operator $\sigma_{\nu^{(m)}}$ via update laws of the general format $\mathbf{s}_{\nu^{(m)}}(\text{next-}t) = \sigma_{\nu^{(m)}}(\mathbf{s}_{\nu^{(m)}}(t), (a_{\nu^{(m)}}(t))_{\nu^{(m)} \in B_{\nu^{(m)}}^{\downarrow}})$. In words: the memory state of $\nu^{(m)}$ becomes updated by incorporating information from the activations of its component observers. The question how information about previously observed input is encoded and preserved in memory states has been intensely studied, under the headlines of the echo state property and fading memory, in the reservoir computing field^{107–112}.

A common theme in complex systems modeling is to capture how the dynamics of subsystems are modulated in a top-down way by superordinate master systems in which the subsystem is a component. This question arises, for instance, in neuroscience where top-down subsystem control serves functions like attention, setting predictive priors, or modulation of motion commands; or in AC programs when arguments are passed down in function calling hierarchies; or in hierarchical AI planning systems where subgoals for subsystems are passed down from higher-up planning systems; or in physics where the interaction of multi-particle systems is modulated by external fields. In our FC proposal we capture top-down control effects by modulating how the activation of a component observer $\nu^{(m)}$ is determined by its memory state: $a_{\nu^{(m)}}(t) = \alpha_{\nu^{(m)}}(\mathbf{s}_{\nu^{(m)}}(t), (a_{\nu^{(m)}}(t))_{\nu^{(m)} \in B_{\nu^{(m)}}^{\downarrow}})$, where $\alpha_{\nu^{(m)}}$ is the activation function of $\nu^{(m)}$.

The dynamics of the memory state $\mathbf{s}_{\nu^{(m)}}(t)$ and activation $a_{\nu^{(m)}}(t)$ are thus co-determined bottom-up by the components of $\nu^{(m)}$ and top-down by the master compounds in which $\nu^{(m)}$ is a component, respectively, via the state and activation update functions $\sigma_{\nu^{(m)}}$ and $\alpha_{\nu^{(m)}}$. These interactions are an explicit part of an FC model. They induce implicit interactions between observers, in that two observers may share components or be components in shared compounds. We use the word coupling for these lateral, indirect interactions. Coupling interactions are not explicitly reflected in an FC model. They can, however, have a strong impact on the overall emerging system dynamics. This poses a challenge for the

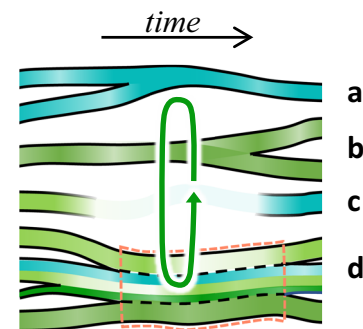


Fig. 2 | Dynamical re-configuration effects in FC modeling. Observers and their associated chronicles can dynamically bind and unbind in and from compounds during the execution of an FC model, leading to a variety of structural re-organization effects that have obvious analogues in algorithmic computing. From top to bottom: merging (a) and splitting (copying, b) of observers; termination and creation (c); binding and unbinding (d). The central segment in d (dashed orange outline) shows the temporary presence of a compound observer made from two primitive observers and one compound observer, which in turn is a binding of three primitive ones. The compound observers have activation histories of their own, which are not shown in the graphic. Coupling dynamics is symbolized by the circular arrow.

modeler, who must foresee these interactions. This challenge of identifying emerging global organization (or disorganization) phenomena in complex dynamical systems is a notorious problem in all complex systems sciences, and it is also a main problem when it comes to ensure global functionality in complex AC software systems. Two highlight examples: Slotine and Lohmiller¹⁰⁰ derive formal contractivity

conditions for the stability of subsystems such that global system stability (in the sense of recovery from perturbations) is guaranteed; and Hens et al.⁴⁰ analyse how a local perturbation from a globally stable system state spreads in activation waves through a network of coupled subsystems. In this regard, FC modeling faces the same challenges as other complex system modeling approaches.

BOX 3

Some observations about observing

In our project of developing a theory of fluent computing (FC), we interpret the information-carrying model variables as observers (detectors, sensors, measurement apparatuses). Here we point out some aspects of the observer concept that an FC theory should cover.

a. Localizing bit signals. An observer—whether a voltmeter or a human—does not observe the world at large but focuses on a specific signal source. One way to identify sources is by spatial localization. The binary bit switching signals in a digital microchip can be picked up at pointlike, non-moving localizations.

b. Observables can be spatially extended. In physical substrates (image insert shows the material from Box 1) one can observe objects or phenomena that are spatially extended, geometrically time-varying, and moving—fields, wave fronts, particles and more. An observer must have means to identify and track such objects.

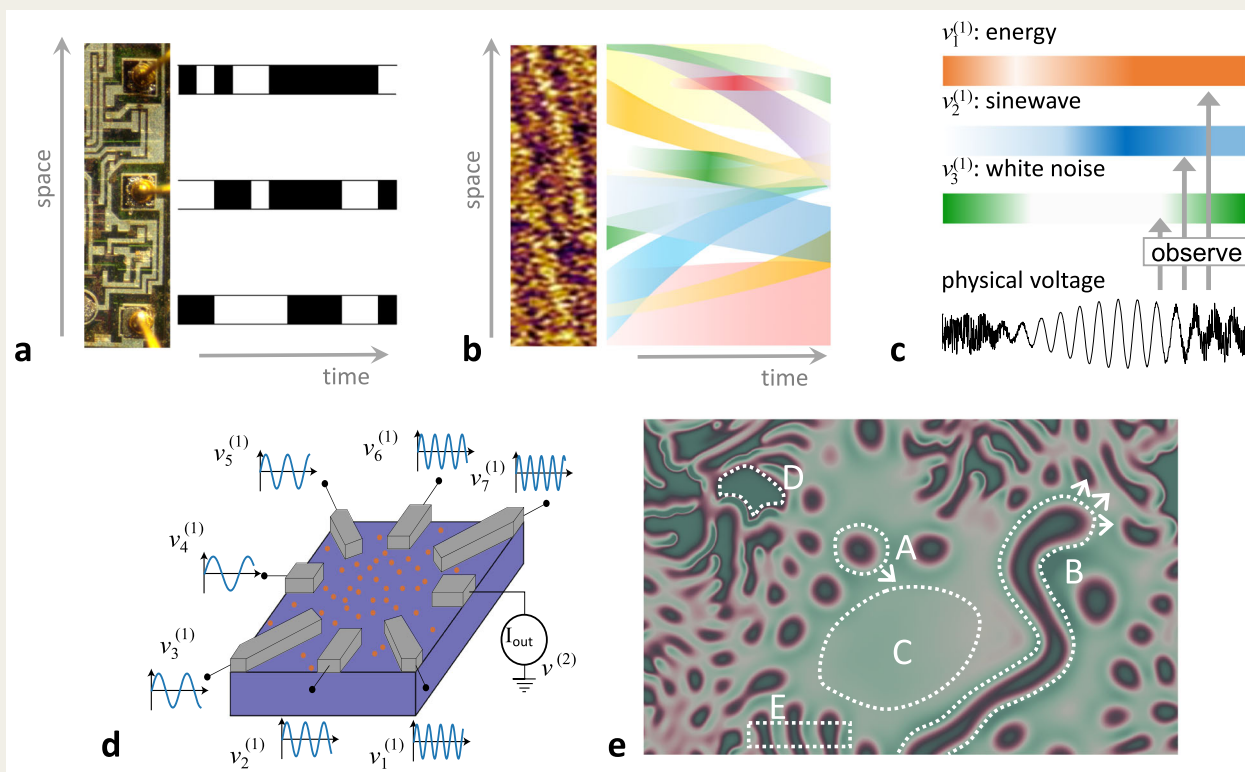
c. There are unlimited ways to define observables. The graphic shows response signals from three observers $v_1^{(1)}, v_2^{(1)}, v_3^{(1)}$ whose common source is the voltage of an electronic contact point. Their activation (indicated by colour intensity) respond to the short-term averaged signal energy, a specific sine frequency response, and the white noise component, respectively.

d. A complex physical observer. The drawing (adapted from ref. 118) shows a schematic of a dopant network processing unit

(DNPU)³¹, an unconventional nanoscale device developed in one of our labs. It consists of a doped silicon well which here is contacted by seven input and one output electrode. DNPU exhibit strongly nonlinear charge transport behaviour between the electrodes. The drawing shows an experiment where different input voltage signals lead to a nonlinear current response in the output. A neural network was trained to model this highly nonlinear 7-input, 1-output function. Using this model, modular DNPU architectures were mathematically optimized to yield very compact signal processing and pattern recognition systems, including handwritten digit recognition¹⁹, which were then tested with physical DNPU.

This neural network model of a DNPU can be formally cast as a level $L^{(2)}$ observer of seven signals $v_1^{(1)} - v_7^{(1)}$ on the modeling level $L^{(1)}$. The activation signal $v^{(2)}$ of the trained model output neuron is the response of this observer.

e. Observing spatiotemporal patterns. This graphic (background image created with the online tool²⁰) shows a snapshot from an evolving chemical reaction-diffusion system. We might consider as observable phenomena, for example, moving solitons (A), moving and growing filaments (B), neutral ground state areas (C), activated areas (D), periodic patterns (E). Arbitrarily more pattern categories can be defined.



Binding relations as well as state and activation update functions $\sigma_{\nu^{(m)}}$ and $\alpha_{\nu^{(m)}}$ can be time-varying, leading to dynamically changing system re-organizations (Fig. 2).

Finally, two models that are adjacent at levels $L^{(m-1)}, L^{(m)}$ in an abstraction hierarchy of FC models, are related to each other in that some observers $\nu^{(m)}$ in level $L^{(m)}$ can be declared as observing observers $\nu^{(m-1)}$ in the model below. This defines a grounding of the level- $L^{(m)}$ model in the level- $L^{(m-1)}$ model, and most importantly, a grounding of the machine-interfacing model at level $L^{(1)}$ in the physical reality of the underlying physical system. We must refer the reader to our long version¹⁶ of this perspective for a detailed discussion of consistency conditions for this level-crossing model abstraction and a comparison with model abstractions in traditional computer science and the natural sciences. In Box 3 we point out some facts about observations that are pertinent to building model abstraction hierarchies.

Outlook. Research in neuromorphic and other unconventional kinds of computing is thriving, but still lacking a unifying theory grounding. We propose to anchor such a theory in three ideas: viewing information processing as a dynamical system (adopted from the cybernetic paradigm), organizing these dynamics in hierarchical binding compounds (adopted from the algorithmic paradigm), and ground theory abstraction in hierarchies of formal observers (following physics). Surely there are many ways how these ideas can be tied together in mathematical detail. Our fluent computing proposal is a first step in one of the possible directions. We hope that this perspective article (and its long version¹⁶) gives useful orientation for theory builders who, like ourselves, are searching for the key to unlock the richness of material physics at large for engineering neuromorphic and other unconventional computing systems.

References

- Andrae, A. S. G. & Edler, T. On global electricity usage of communication technology: trends to 2030. *Challenges* **6**, 117–157 (2015).
- Zhao, H. et al. A New Circular Vision for Electronics: Time for a Global Reboot. Report in Support of the United Nations E-waste Coalition, World Economic Forum. <https://www.weforum.org/reports/a-new-circular-vision-for-electronics-time-for-a-global-reboot> (2019).
- Waldrop, M. M. More than Moore. *Nature* **530**, 144–147 (2016).
- Ebert, C. 50 years of software engineering: progress and perils. *IEEE Softw.* **35**, 94–101 (2018).
- Mead, C. Neuromorphic electronic systems. *Proc. IEEE* **78**, 1629–1636 (1990).
- Bournez, O. & Pouly, A. in *Handbook of Computability and Complexity in Analysis, Theory and Applications of Computability* (eds Brattka, V. & Hertling, P.) 173–226 (Springer, Cham, 2021).
- Huang, C. et al. Prospects and applications of photonic neural networks. *Adv. Phys. X* **7**, 1981155 (2022).
- Tanaka, G. et al. Recent advances in physical reservoir computing: a review. *Neural Netw.* **115**, 100–123 (2019).
- van Noort, D., Gast, F.-U. & McCaskill, J. S. in *DNA Computing Vol. 2340* (eds Jonoska, N. & Seeman, N. C.) 33–45 (Springer Verlag, 2002).
- Doty, D. Theory of algorithmic self-assembly. *Commun. ACM* **55**, 78–88 (2012).
- Adamatzky, A., De Lacy Costello, B. & Asai, T. *Reaction Diffusion Computers* (Elsevier, 2005).
- Ricciardi, C. & Milano, G. In materia should be used instead of in materio. *Front. Nanotechnol.* **4**, article 850561 (2022).
- European Commission Author Collective. Unconventional formalisms for computation: expert consultation workshop. Preprint at https://cordis.europa.eu/pub/fp7/ict/docs/fet-proactive/shapefetip-wp2011-12-05_en.pdf (2009).
- Adamatzky, A. (ed.) *Advances in Unconventional Computing Vol. 1: Theory, and Vol. 2: Prototypes, Models and Algorithms* (Springer International Publishing, 2017).
- Jaeger, H. Toward a generalized theory comprising digital, neuromorphic, and unconventional computing. *Neuromorph. Comput. Eng.* **1**, 012002 (2021).
- Jaeger, H., Noheda, B. & van der Wiel, W. G. Toward a formal theory for computing machines made out of whatever physics offers: extended version. Preprint at <https://arxiv.org/abs/2307.15408> (2023).
- Kaspar, C., Ravoo, B. J., van der Wiel, W. G., Wegner, S. V. & Pernice, W. H. P. The rise of intelligent matter. *Nature* **594**, 345–355 (2021).
- Zauner, K. P. From prescriptive programming of solid-state devices to orchestrated self-organisation of informed matter. in *Unconventional Programming Paradigms Vol. 3566* (eds Banâtre, J.-P., Fradet, P., Jean-Louis Giavitto, J.-L. & Michel, O.) 47–55 (Springer, Berlin, Heidelberg, 2005).
- Yang, J. J., Strukov, D. B. & Stewart, D. R. Memristive devices for computing. *Nat. Nanotechnol.* **8**, 13–24 (2013).
- Li, C. et al. Long short-term memory networks in memristor crossbar arrays. *Nat. Mach. Intell.* **1**, 49–57 (2019).
- Rabinovich, M. I., Huerta, R., Varona, P. & Afraimovich, V. S. Transient cognitive dynamics, metastability, and decision making. *PLoS Comput. Biol.* **4**, e1000072 (2008).
- Gros, C. Cognitive computation with autonomously active neural networks: an emerging field. *Cognit. Comput.* **1**, 77–90 (2009).
- Chialvo, D. R. Emergent complex neural dynamics. *Nat. Phys.* **6**, 744–750 (2010).
- Stieg, A. Z. et al. Emergent criticality in complex Turing B type atomic switch networks. *Adv. Mater.* **24**, 286–293 (2012).
- Beggs, J. M. & Timme, N. Being critical of criticality in the brain. *Front. Physiol.* **3**, article 00163 (2012).
- Lins, J. & Schöner, G. in *Neural Fields: Theory and Applications*, (eds Coombes, S., beim Graben, P., Pothast, R. & Wright, J.) 319–339 (Springer Verlag, 2014).
- Grollier, J. et al. Neuromorphic spintronics. *Nat. Electron.* **3**, 360–370 (2020).
- Kuncic, Z. & Nakayama, T. Neuromorphic nanowire networks: principles, progress and future prospects for neuro-inspired information processing. *Adv. Phys. X* **6**, 1894234 (2021).
- Lee, O. et al. Perspective on unconventional computing using magnetic skyrmions. *Appl. Phys. Lett.* **122**, 260501 (2023).
- Bose, S. K. et al. Evolution of a designless nanoparticle network into reconfigurable Boolean logic. *Nat. Nanotechnol.* **10**, 1048–1052 (2015).
- Chen, T. et al. Classification with a disordered dopant-atom network in silicon. *Nature* **577**, 341–345 (2020).
- Shannon, C. E. Mathematical theory of the differential analyzer. *J. Math. Phys.* **20**, 337–354 (1941).
- Moore, C. Recursion theory on the reals and continuous-time computation. *Theor. Comput. Sci.* **162**, 23–44 (1996).
- von Neumann, J. Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Autom. Stud.* **34**, 43–98 (1956).
- Kanerva, P. Hyperdimensional computing: an introduction to computing in distributed representation with high-dimensional random vectors. *Cognit. Comput.* **1**, 139–159 (2009).
- Ashby, W. R. *Design for a Brain* (John Wiley and Sons, New York, 1952).
- Albus, J. S. in *An Introduction to Intelligent and Autonomous Control* (eds Antsaklis, P. J. & Passino, K. M.) Ch. 2 (Kluwer Academic Publishers, 1993).
- Eliasmith, C. A unified approach to building and controlling spiking attractor networks. *Neural Comput.* **17**, 1276–1314 (2005).

39. Arathorn, D. W. A cortically-plausible inverse problem solving method applied to recognizing static and kinematic 3D objects. In *Proc. NIPS 2005 Preprint* at http://books.nips.cc/papers/files/nips18/NIPS2005_0176.pdf (2005).
40. Hens, C., Harush, U., Haber, S., Cohen, R. & Barzel, B. Spatio-temporal signal propagation in complex networks. *Nat. Phys.* **15**, 403–412 (2019).
41. Wolfram, S. *A New Kind of Science* (Wolfram Media, 2002).
42. Haken, H. Self-organization. *Scholarpedia* **3**, 1401 (2008).
43. Yao, Y. & Freeman, W. A model of biological pattern recognition with spatially chaotic dynamics. *Neural Netw.* **3**, 153–170 (1990).
44. Legenstein, R. & Maass, W. in *New Directions in Statistical Signal Processing: From Systems to Brain* (eds Haykin, S., Principe, J. C., Sejnowski, T. J. & McWhirter, J.) 127–154 (MIT Press, 2007).
45. Dorigo, M. & Gambardella, L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1**, 53–66 (1997).
46. Farmer, J. D., Packard, N. H. & Perelson, A. S. The immune system, adaptation, and machine learning. *Phys. D* **22**, 187–204 (1986).
47. Holland, J. A mathematical framework for studying learning in classifier systems. *Phys. D* **22**, 307–317 (1986).
48. Mitchell, M. *An Introduction to Genetic Algorithms* (MIT Press/Bradford books, 1996).
49. Arenas, A., Díaz-Guilera, A., Kurths, J., Moreno, Y. & Zhou, C. Synchronization in complex networks. *Phys. Rep.* **469**, 93–153 (2008).
50. Barabási, A. L. Network science. *Philos. Trans. R. Soc. Lond. Ser. A* **371**, 20120375 (2013).
51. Mumford, D. Pattern theory: the mathematics of perception. In *Proc. ICM 2002, Vol. 1*, 401–422 Preprint at <https://arxiv.org/abs/math/0212400> (2002).
52. Bill, J. et al. Distributed Bayesian computation and self-organized learning in sheets of spiking neurons with local lateral inhibition. *PLoS One* **10**, e0134356 (2015).
53. Friston, K. J., Parr, T. & de Vries, B. The graphical brain: belief propagation and active inference. *Network Neurosci* **1**, 381–414 (2017).
54. Harnad, S. What is computation (and is cognition that)?—preface to special journal issue. *Minds Mach.* **4**, 377–378 (1994).
55. Piccinini, G. Computing mechanisms. *Philos Sci.* **74**, 501–526 (2007).
56. Church, A. An unsolvable problem of elementary number theory. *Am. J. Math.* **58**, 345–363 (1936).
57. Turing, A. M. On computable numbers, with an application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.* **42**, 230–265 (1936).
58. Bickerton, D. & Szathmáry, E. (eds.) *Biological Foundations and Origin of Syntax* (MIT Press, 2009).
59. Bartlett, F. C. *Remembering: A Study in Experimental and Social Psychology* (Cambridge University Press, 1932).
60. Lakoff, G. *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind* (University of Chicago Press, 1987).
61. Hofstadter, D. *Fluid Concepts and Creative Analogies* (Harper Collins/Basic Books, 1995).
62. Cruse, H., Dean, J. & Ritter, H. (eds.) *Prerational Intelligence: Adaptive Behavior and Intelligent Systems Without Symbols and Logic* Vol. 3 Vol (Springer Science & Business Media, 2013).
63. Bradie, M. Assessing evolutionary epistemology. *Biol Philos* **1**, 401–459 (1986).
64. Greenfield, P. Language, tools and brain: The ontogeny and phylogeny of hierarchically organized sequential behavior. *Behav. Brain Sci.* **14**, 531–595 (1991).
65. Drescher, G. L. *Made-up Minds: A Constructivist Approach to Artificial Intelligence* (MIT Press, 1991).
66. Pfeifer, R. & Scheier, C. *Understanding Intelligence* (MIT Press, 1999).
67. Lakoff, G. & Nunez, R. E. *Where Mathematics Comes From: How the Embodied Mind Brings Mathematics Into Being* (Basic Books, 2000).
68. Fedor, A., Ittész, P. & Szathmáry, E. in *Biological Foundations and Origin of Syntax* 15–39 (MIT Press, 2009).
69. Schroeder, B. *Exercise Book for Elementary School: Deepen Addition, Subtraction, Multiplication and Division Skills* (Independently published, 2022).
70. Maxwell, J. C. On governors. *Proc. R. Soc. Lond.* **16**, 270–283 (1886).
71. van Gelder, T. What might cognition be, if not computation? *J. Philos.* **92**, 345–381 (1995).
72. Abadi, M. et al. TensorFlow: a system for large-scale machine learning. In *Proc. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, 265–283 (2016).
73. Jaeger, H. & Catthoor, F. Timescales: the choreography of classical and unconventional computing. Preprint at <https://arxiv.org/abs/2301.00893> (2023).
74. Jékely, G., Godfrey-Smith, P. & Keijzer, F. Reafference and the origin of the self in early nervous system evolution. *Philos. Trans. R. Soc. Lond. Ser. B* **376**, 20190764 (2021).
75. Blum, L., Shub, M. & Smale, S. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bull. Am Math. Soc.* **21**, 1–46 (1989).
76. Wegner, P. & Goldin, D. Computation beyond turing machines. *Commun ACM* **46**, 100–102 (2003).
77. Endrullis, J., Klop, J. W. & Bakhshi, R. Transducer degrees: atoms, infima and suprema. *Acta Inf.* **57**, 727–758 (2019).
78. Adolphs, R. The unsolved problems of neuroscience. *Trends Cognit. Sci.* **19**, 173–175 (2015).
79. Gerstner, W., Sprekeler, H. & Deco, D. Theory and simulation in neuroscience. *Science* **338**, 60–65 (2012).
80. Hodgkin, A. L. & Huxley, A. F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol.* **117**, 500–544 (1952).
81. Ermentrout, B. Complex dynamics in winner-take-all neural nets with slow inhibition. *Neural Netw.* **5**, 415–431 (1992).
82. Hopcroft, J. E., Motwani, R. & Ullman, J. D. *Introduction to Automata Theory, Languages, and Computation* 3rd edn (Pearson, 2006).
83. Stewart, T. C., Bekolay, T. & Eliasmith, C. Neural representations of compositional structures: Representing and manipulating vector spaces with spiking neurons. *Connection Sci* **23**, 145–153 (2011).
84. Bekolay, T. et al. Nengo: a Python tool for building large-scale functional brain models. *Front. Neuroinf.* **7**, 1–13 (2014).
85. Eliasmith, C. et al. A large-scale model of the functioning brain. *Science* **338**, 1202–1205 (2012).
86. Neckar, A. et al. Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model. *Proc. IEEE* **107**, 144–164 (2019).
87. Taatgen, N. A spiking neural architecture that learns tasks. In *Proc. 17th Int. Conf. on Cognitive Modeling (CCM 2019, held 2020)*, (ed. Stewart, T. C) 253–258 (Applied Cognitive Science Lab, Penn State, 2019).
88. Angelidis, E. et al. A spiking central pattern generator for the control of a simulated lamprey robot running on SpiNNaker and Loihi neuromorphic boards. *Neuromorphic Comp. Eng.* **1**, 014005 (2021).
89. Zhang, Y. et al. A system hierarchy for brain-inspired computing. *Nature* **586**, 378–384 (2020).
90. Samsonovich, A. V. Toward a unified catalog of implemented cognitive architectures. In *Proc. Int. Conf. on Biologically Inspired Cognitive Architectures (BICA 2010)*, 195–244 (BICA Society,

- 2010). Preprint at https://www.researchgate.net/publication/221313271_Toward_a_Unified_Catalog_of_Implemented_Cognitive_Architectures.
91. Kwisthout, J. & Donselaar, N. On the computational power and complexity of spiking neural networks. In *Proc. 2020 Annual Neuro-Inspired Computational Elements Workshop (NICE 20)* (ACM, Inc., 2020).
 92. von Neumann, J. in *John von Neumann: Collected Works* Vol. 5, 288–328 (Pergamon Press, 1963).
 93. Horsman, D., Kendon, V. & Stepney, S. The natural science of computing. *Commun. ACM* **60**, 31–34 (2017).
 94. Schuman, C. D. et al. Opportunities for neuromorphic computing algorithms and applications. *Nat. Comput. Sci.* **2**, 10–19 (2022).
 95. Mehonic, A. & Kenyon, A. J. Brain-inspired computing needs a master plan. *Nature* **604**, 255–260 (2022).
 96. Newton, I. *De Analysi per Aequationes Numero Terminorum Infinitas* (Posthumously, 1669).
 97. Saal, H. P. & Bensmaia, S. J. Touch is a team effort: interplay of submodalities in cutaneous sensibility. *Trends Neurosci.* **37**, 689–697 (2014).
 98. Thurner, S., Hanel, R. & Klimek, P. Networks. (eds Thurner, S., Klimek, P. & Hanel, R.) In *Introduction to the Theory of Complex Systems*, Ch. 4 (Oxford University Press, 2018).
 99. Simon, H. A. The architecture of complexity. *Proc. Am. Philos. Soc.* **106**, 467–482 (1962).
 100. Slotine, J. J.-E. & Lohmiller, W. Modularity, evolution, and the binding problem: a view from stability theory. *Neural Netw.* **14**, 137–145 (2001).
 101. Antoulas, A. C. & Sorensen, D. C. Approximation of large-scale dynamical systems: an overview. *Int. J. Appl. Math. Comput. Sci.* **11**, 1093–1121 (2001).
 102. Kuehn, C. *Multiple Time Scale Dynamics* (Springer Verlag, 2015).
 103. Treisman, A. Feature binding, attention and object perception. *Philos Trans. R. Soc. Lond. Ser B* **353**, 1295–1306 (1998).
 104. Diesmann, M., Gewaltig, M.-O. & Aertsen, A. Stable propagation of synchronous spiking in cortical neural networks. *Nature* **402**, 529–533 (1999).
 105. Shastri, L. Advances in Shruti—a neurally motivated model of relational knowledge representation and rapid inference using temporal synchrony. *Art Intell* **11**, 79–108 (1999).
 106. Legenstein, R., Papadimitriou, C. H., Vempala, S. & Maass, W. Assembly pointers for variable binding in networks of spiking neurons. In *Proc. of the 2016 Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*. Preprint at <https://arxiv.org/abs/1611.03698> (2016).
 107. Jaeger, H. The “echo state” approach to analysing and training recurrent neural networks. GMD Report 148, GMD—German National Research Institute for Computer Science. Preprint at <https://www.ai.rug.nl/minds/pubs> (2001).
 108. Maass, W., Natschläger, T. & Markram, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* **14**, 2531–2560 (2002).
 109. Yildiz, I. B., Jaeger, H. & Kiebel, S. J. Re-visiting the echo state property. *Neural Netw.* **35**, 1–20 (2012).
 110. Manjunath, G. & Jaeger, H. Echo state property linked to an input: exploring a fundamental characteristic of recurrent neural networks. *Neural Comput.* **25**, 671–696 (2013).
 111. Grigoryeva, L. & Ortega, J.-P. Echo state networks are universal. *Neural Netw.* **108**, 495–508 (2018).
 112. Gonon, L., Grigoryeva, L. & Ortega, J.-P. Memory and forecasting capacities of nonlinear recurrent networks. *Phys. D* **414**, 132721 (2020).
 113. Langton, C. G. Computation at the edge of chaos: phase transitions and emergent computation. *Phys. D* **42**, 12–37 (1990).
 114. Kinouchi, O. & Copelli, M. Optimal dynamical range of excitable networks at criticality. *Nat. Phys.* **2**, 348–351 (2006).
 115. Everhardt, A. S. et al. Temperature-independent giant dielectric response in transitional BaTiO₃ thin films. *Appl. Phys. Rev.* **7**, 011402 (2020).
 116. Everhardt, A. S. et al. Periodicity-doubling cascades: direct observation in ferroelastic materials. *Phys. Rev. Lett.* **123**, 087603 (2019).
 117. Rieck, J. et al. Ferroelastic domain walls in BiFeO₃ as memristive networks. *Adv. Intell. Syst.* **5**, 2200292 (2023).
 118. Ruiz Euler, H.-C. et al. A deep-learning approach to realizing functionality in nanoelectronic devices. *Nat. Nanotechnol.* **15**, 992–998 (2020).
 119. Ruiz Euler, H.-C. et al. Dopant network processing units: towards efficient neural-network emulators with high-capacity nanoelectronic nodes. *Neuromorphic Comput. Eng.* **1**, 024002 (2021).
 120. Sims, K. *Artistic Visualization Tool for Reaction-Diffusion Systems*. <http://www.karlsims.com/rdtool.html> (2023).

Acknowledgements

H.J. acknowledges financial support from the European Horizon 2020 projects Memory technologies with multi-scale time constants for neuromorphic architectures (grant Nr. 871371) and Post-Digital (grant Nr. 860360). W.G.v.d.W. acknowledges financial support from the HYBRAIN project funded by the European Union’s Horizon Europe research and innovation programme under Grant Agreement No 101046878 and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through project 433682494—SFB 1459. B.N. acknowledges funding from the European Union’s Horizon 2020 ETN programme Materials for Neuromorphic Circuits (MANIC) under the Marie Skłodowska-Curie grant agreement No 861153. Financial support by the Groningen Cognitive Systems and Materials Center (CogniGron) and the Ubbo Emmius Foundation of the University of Groningen is gratefully acknowledged. We are immensely grateful for the extraordinary effort spent by the anonymous reviewers to help us in making this article understandable to an interdisciplinary readership.

Author contributions

H.J. developed the theoretical aspects of the fluent computing proposal. B.N. and W.G.v.d.W. provided the empirical and physical motivation and underpinnings. Draft writing, reviewing and editing was a collaborative effort.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Herbert Jaeger.

Peer review information *Nature Communications* thanks James Aimone, Hermann Kohlstedt and the other anonymous reviewer(s) for their contribution to the peer review of this work.

Reprints and permissions information is available at <http://www.nature.com/reprints>

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023