On Epistemically Useful Physical Computation

Tim Schmitz

timrschmitz@gmail.com

Abstract: Piccinini's Usability Constraint states that physical processes must have "physically constructible manifestation[s]" to be included in epistemically useful models of physical computation. But to determine what physical processes can be implemented on physical systems (as parts of computations), we must already know what physical processes can be implemented on physical systems (as parts of processes for constructing computing systems). We need additional assumptions about what qualifies as a building process. Piccinini implicitly assumes a classical computational understanding of executable processes, but this is an assumption imposed on physical theories and may artificially limit our picture of epistemically useful physical computation.

1 Introduction

This discussion emerges from two fundamental questions: What is physically computable? And what is the relationship between Turing computability and physical computability? As Turing computability is the central force of computability theory, the former question is often posed in terms of the latter (e.g. in Arrighi and Dowek 2012; Cotogno 2003; Hogarth 1994; Shagrir and Pitowsky 2003; Ziegler 2009 and countless others). Piccinini's discussion of the physical Church-Turing thesis in (Piccinini 2018) and (Piccinini 2011) follows this format. He argues that, if notions of computability are to be linked to what is epistemically useful to finite observers, a modest version of the physical Church-Turing thesis likely holds. This Modest Physical Church-Turing Thesis states that what is Turing computable acts as an upper limit for what is physically computable, given some constraints on what is considered a physical computation. These constraints are meant to restrict discussion to physical computations which could be epistemically useful to finite observers.

Though the Modest Physical Church-Turing thesis may seem plausible, we will see that the account of what counts as an epistemically useful physical computation that Piccinini uses to argue for this thesis requires more explicit conceptual grounding. In particular, I will argue that it begs the question regarding what physical processes one considers possible computational operations and implicitly fills in this gap with the assumption that physical computational operations correspond to processes that can be executed by physical systems that can be built using classical (Turing) computational processes on physical systems.

Trying to determine which physical processes can correspond to epistemically useful computational operations leads to a bootstrapping problem: We want to limit epistemically useful computations to those with a physically buildable manifestation for finite observers such as ourselves. However, our criteria for determining which physical processes finite observers can implement as part of a computational process require us to already know which physical processes finite observers can implement to build physical systems to implement that computation. This undermines the usefulness of buildability as a constraint for determining which physical computations are epistemically useful.

Piccinini implicitly assumes a classical (Turing) computation-based position in which finite observers can implement Turing-computable processes to build computing systems. While this classical starting points is not without merits – an assumed starting point is required and I will argue that a classical computation-based one is a good default position in 4.2 – it is also not the only possible position, and it might even be ill-suited at describing what is possible for finite observers in some physical theories. Additionally, this position may lead us to overestimate the likelihood that the Modest Physical Church-Turing thesis holds.

Section 2 discusses what is meant by physical computability in the context of this paper. Section 3 discusses what a model of physical computation is, in light of Piccinini's conditions for ensuring physical computational models are epistemically useful to finite observers. Section 4 argues that, given a physical theory, there is an inherent freedom for selecting physical operations to ground what counts as a model of computation in that theory. This results in a bootstrapping problem unless we impose a selection of what physically possible processes to consider possible building blocks for models of computation in a theory. The selection, implicitly made by Piccinini, derived from a classical computational understanding of operations, though a particularly compelling response to this freedom, is not without its weaknesses. Section 5 concludes.

2 Physical Computability Questions

Using Turing machines in classical universes whose spatial dimensions are described by hyperreal, rather than real, lines, Aitken and Barrett show that what is computable using a Turing machine depends on the physical theory in which it resides (Aitken and Barrett 2010; Aitken and Barrett 2009). They conclude that, generally, what is physically computable depends on the physical theory. Additionally, they propose that questions of physical computing power can only be answered relative to a physical theory and a computational model constructed relative to this theory. Thus, questions about physical computability should be answered relative to a particular computational model constructed relative to a particular physical theory.

On its own, this proposal tells us nothing about how we should select models of computation relative to physical theories. But determining the limits of physical computability and whether different versions of the physical Church-Turing thesis hold relies on some assessment of what counts as a model of physical computation in some physical theory. If we do not properly restrict how computational states and operations can be mapped to physical states and operations, then we can end up with some version of pancomputationalism where every physical system performs either some computation or every computation.[1]

One might reasonably hope that the selection of appropriate physical computational

[1]See (Piccinini 2017) for an overview, or (Müller 2014; Piccinini and Anderson 2018; Piccinini 2007) for some examples of critical responses to pancomputationalism.

models can be done in a principled way, ideally by catering our selection of computational model to leverage a physical theory's properties while reflecting what finite observers governed by this theory could possibly achieve. We will explore such an approach in Section 4.1 and find that it succumbs to a bootstrapping problem without supplementary assumptions that do not arise from the physical theory itself. Thus, Piccinini's Usability Constraint requires additional specification of what operations finite observers can utilize which do not arise from the physical theory or the Usability Constraint itself. The necessity of additional constraints that do not arise from a particular physical theory on allowed computational operations and arguments for and against Piccinini's position will be discussed in Section 4.2.

3 Models of Physical Computation

For the purposes of this paper, a physical theory $\mathcal{T}$ will specify sets of objects J, possible physical states S_j for these objects, and dynamical rules $\mathcal{R}$ which describe how the states of these objects evolve and interact. For instance, the objects of J may be particles or perturbations of a field and will constitute the physical systems used to construct a model of computation. The states S_j will describe the properties of these objects, such as location or energy level. And the dynamical rules $\mathcal{R}$ determine how these objects and states can evolve. This definition is meant to be broad enough to encompass a wide variety of physical theories.

Given an appropriate physical theory $\mathcal{T}$, there are an immense number of ways to construct a model of computing in that theory. Any process can be considered to be a computation of almost anything if one is liberal enough with what one considers a computation, as the pancomputationalism literature demonstrates.[2] We will want to consider models of physical computation that follow the Aitken and Barrett's prescription in light of Piccinini's epistemological concerns. To give us a tractable starting point, I will define a model of computation as follows.

A model of computation relative to $\mathcal{T}$, $\mathcal{M}_{\mathcal{T}}$, will consist of finite numbers of units of information ("bits" of some sort) $\mathscr{B}$ and operations $\mathscr{O}$ that can be performed on these units of information; these will arise from information-carrying objects $B \subseteq J$ and operator systems $O \subseteq J$ which act on the elements of B according to dynamical rules $\mathcal{R}$ to change their corresponding physical states. There may not be a one-one correlation between units of information $\mathscr{B}$ and elements or even subsets of B. For instance, a unit of information may be carried by one particle or multiple particles, perhaps even within a single computer. Importantly, the physical states of B serve to transmit information, and the evolution of these states according to the dynamical rules $\mathcal{R}$ of the theory and through the influence of operator systems allows computation.

Piccinini proposes a usability constraint to restrict discussion of physical computation to those physical computational models that would be epistemically useful for finite observers:

Usability Constraint: *if a physical process is a computation, it can be used by a finite observer to obtain the desired values of a function*[3] (Piccinini 2011).

This constraint is broken down into four sub-constraints on physical processes:

[2] See Footnote 1.

[3] Here Piccinini only considers processes that fit this constraint to be computations. I will instead distinguish between usable physical computations that fit this constraint and physical computations more broadly, which may or may not fit this constraint.

> "An *executable* physical process is one that a finite observer can set in motion to generate the values of a desired function until it generates a readable result."
>
> "An *automatic* physical process is one that runs without requiring intuitions, ingenuity, invention, or guesses."
>
> "A *uniform* physical process is one that doesn't need to be redesigned or modified for different inputs."
>
> "[A] *reliable* physical process is one that generates results at least some of the time, and, when it does so, its results are correct." (Piccinini 2011, 741)

Our concern will be a particular aspect of the executability criterion. To be *executable*, a physical process must have inputs and outputs that are readable for a finite observer, the ability to solve problems that can be "defined independently of the processes that compute them", the ability to be repeated, the ability to set the computing system into a particular initial state, and *a physically constructible manifestation* (Piccinini 2011). The constructibility criterion implies a distinction between physically possible objects and processes and the physical processes and objects that could be constructed by some physically possible finite agent. This limits discussion of physical computation to things which can finites observers could execute and find epistemically useful.

The last component of the executability condition – that the physical process corresponds to some system that can be physically constructed – will be the main target of our concern, because an investigation of this criterion reveals an unrecognized assumption that guides the rest of Piccinini's work on the Modest Physical Church-Turing Thesis. What we now want to consider is how this restriction would play out when developing models of computation relative to a physical theory. For the sake of avoiding confusion with other concepts, we will use “build” where Piccinini uses “construct”.

4 Determining What is Buildable

We will see that Piccinini’s executability sub-constraint, due to ambiguities in how to determine what is buildable, can leave open a wider range of possibilities for computational operations than seems desirable and which seems to have been acknowledged in the literature. By investigating ways one can describe what counts as “physically constructible” (what we call “physically buildable” to avoid confusion with other ideas tied to the word “constructible”), we will see that Piccinini implicitly supplements this concept with a notion of what operations finite observers can use which is based in classical computation, rather than on the physical theory itself, in contrast with the prescription described by Aitken and Barrett. Nonetheless, I will argue that adapting the implicit classical notion underpinning Piccinini’s discussion of his usability constraint to a particular physical theory is a strong starting point for investigating physical computation in that theory, though we should be explicitly aware of that it is an assumption we are making.

4.1 Buildable Physical Systems & a Bootstrapping Problem

If our concern is what a physical theory allows finite observers to do, not all physically possible system states may be useful for computation. For a model of physical computation to pass Piccinini’s executability condition, finite observers must be able to reliably build devices that manifest it. One then needs to understand what processes finite observers can reliably use to build physical systems to act as computers. If a finite observer cannot systematically put a system in (or extremely near to) a state, then that state cannot be part of a model of computation that is useful for finite observers, even if the state can be realized in a world governed by theory $\mathcal{T}$. This applies to both the information-carrying systems and the operator systems; we need systematic processes within the theory to build the elements of B and O and to set them up for a computation, otherwise these states cannot be epistemically usefully harnessed. These restrictions are motivated by epistemic considerations – they are meant to limit usable models of physical computation to those which are not just physically possible simpliciter but are physically possible ways for an observer within the physical reality to use other systems within the same physical reality to solve some problem (disregarding surmountable technological limitations). Ideally, we would be able to define a notion of "buildability" that emerges from an account of finite observers in a particular physical theory alone. However, we will see that the situation seems not to be so simple, and deciding what counts as buildable requires us to make stipulations that do not arise from the physical theory itself.

What can be built within different physical theories obviously varies; for instance, quantum mechanics allows the construction of Hadamard gates, while classical mechanics may not. Also, what building procedures are available may vary from one theory to another. One theory may allow infinite-step processes to be completed while another does not, or, more typically, one theory may allow access to different physical systems and different ways physical systems can be manipulated than another. If theories allow different processes to manipulate physical systems, they may also allow different operator and bit systems to be built and perhaps different $\mathscr{B}$ and $\mathscr{O}$ which can be used in computational models relative to the theories. But finite physical systems can implement many processes that should not be included in an account of epistemically useful physical computation. The task for understanding useful physical computing is determining which of the physical processes that finite systems can implement can be used to build operator and bit systems for use in physical computers.

Suppose we want to determine what models of computation finite observers can build relative to some physical theory. To determine what can be built for use in a model of computation relative to our physical theory, we must first determine what physical processes in the theory can be used for building computer components. But here we run into a problem – building some component for a computer requires a set of operations we can perform on systems to change their states in order to build that component. But this set of operations we can perform on systems is precisely what we are attempting to discover, as these are the operations which can be used as part of a model of physical computation. So we have a bootstrapping problem – we want to know which physical processes a physical theory allows finite observers to reliably implement (that would be used in computers), but to discover this we must already know what physical processes the physical theory allows finite observers to reliably implement (that would be used in building computers).

Let's make this more concrete. Imagine we have some physical theory $\mathcal{T}$ with a set of objects J which have possible physical states S_j and dynamical rules $\mathcal{R}$ which describe how these states evolve. We want to know what bits $\mathscr{B}$, corresponding with the physical states S_j, and operations $\mathscr{O}$, corresponding with dynamical operations on these states from $\mathcal{R}$, could be used in epistemically useful models of computation $\mathcal{M}_{\mathcal{T}}$ in this physical theory. To find this out, we must know what physical systems finite observers can build to act as bit systems B and operator systems O in computing systems. That is, to determine what B and O can be in an epistemically useful physical computer, we must know how finite observers can systematically manipulate the states S_j of physical systems J to build B and O. But knowing how finite observers can systematically manipulate physical states for use in computing *requires us to already know what operations these finite observer can perform on the states of physical systems.* We need a way to select which physical processes that finite observers can implement to include in our account of buildability in a particular physical theory.

Buildability needs supplementation to work as a useful constraint on what we consider epistemically useful physical computation. While epistemically useful physical computation must in fact fit the buildability criterion, this criterion itself has significant epistemic problems when we try to use it; we would need a full account of what physical operations and states finite observers can and cannot reliably use in a particular physical theory, but this list does not directly fall out of the laws of a physical theory, as finite observers (acting as finite systems) can implement all sorts of operations that we would not want to consider part of a computational model in that theory.[4] For instance, a system acting as a unitary operator in a real quantum computer that we build will, with probability one, correspond to some operator that is classically noncomputable. Yet

[4]Readers may find the work of (Curiel 2020) (following the work of (Stein 1995)) on why "schematizing the observer is required for an adequate philosophical analysis of the structure and semantics of theories" helpful both for orienting themselves in this discussion and for investigating concerns in philosophy of science that in many ways mirror the concerns of this paper.

these operations are not considered part of quantum computing; they can only be approximated, e.g. with the Solovay-Kitaev algorithm (Dawson and Nielsen 2006; Kitaev 1997). We will turn to Geroch's more thorough discussion (Geroch 2009) of this topic in the following section. *We need additional rules for determining which operations pass the buildability criterion.*

Can the rest of Piccinini's Usability Constraint free us from this problem? Let's first look at the rest of the executability sub-constraint. Again, this requires that there be readable inputs and output, the ability to solve problems defined independently of the processes that compute them, repeatability, and the ability to set computing systems into particular initial states, in addition to having a physically buildable manifestation. Whether a process is repeatable and whether a computing system can be set to a particular initial state are consequences of what is physically buildable – if we can build components to act out some physical process, then we can do so repeatedly and we can do so to initialize the state of a computing system. The ability to solve problems defined independently of the processes that compute them is a matter of what computations can be said to compute but not of what basic operations can be implemented on physical systems. Requiring readable inputs and outputs seems to show the most promise, since presumably finite observers cannot access the complete dynamical consequences of all of their actions. For instance, I may be able to throw a ball some classically noncomputable distance (in meters), but I certainly cannot read off the real number corresponding to this distance. But readability itself is at least partially dependent on buildability, in that finite observers may be able to use physical processes to change an output they cannot read into one they can.

The other sub-constraints will be of no help. Whether a purported computing system can be considered “Automatic” will depend on what operations are considered to be automatic processes, which in turn depends on what can be built to act as a computing component and thus cannot be considered to be acting with intuition or ingenuity. Whether a process is Uniform over different inputs and whether a process is capable of getting correct Results are properties of a purported computation as a whole, not its individual components.

Thus, Piccinini’s executability criterion leaves us with a bootstrapping problem: we want to figure out which physical processes we can implement on physical systems (as part of a model of epistemically useful computation), but to determine this we need to already know what these operations are (for use in building computer components to implement a model of computation). In the following section, we will discuss how to avoid this bootstrapping problem and one type of solution adapted from Geroch’s discussion of quantum computation – namely, to take operations corresponding to classical computational operations as a starting point – and why this type of solution is likely the best starting point for investigating physical computation, despite some weaknesses.

4.2 Avoiding the Bootstrapping Problem By Stipulating Constraints

Developing a notion of what is buildable in a physical theory requires us to already have a concept of what operations and systems are allowed in a notion of buildability. Because of the bootstrapping problem, we cannot simply read a notion of buildability off of a physical theory’s treatment of finite systems. Instead, we must stipulate on the theory some external criteria to determine which physical processes finite systems can implement to be considered relevant to physical computation. These criteria do not arise directly from the physical theory or its properties but instead are external constraints on which physically processes are allowed in building procedures and hence which operations and states are considered buildable. This helps us determine what to count as an epistemically useful model of physical computation in a particular physical theory.

To avoid the bootstrapping problem, we could appeal to classical computational processes as a starting point. That is, we would assume that analogs to classical computational operations can be executed in a physical theory; namely, *we can execute finite-step processes where steps involve discrete changes in a physical system's state, and these discrete changes are classically (Turing) computable.* For instance, a rotation of a state through an angle described by a classically computable number is allowed whereas a rotation of a state through a classically noncomputable angle is not – unless we are able to reliably build an operator system that rotates a state through a classically noncomputable angle by using a finite number of classically-derived operations. This solution allows us to bypass the bootstrapping problem by selecting the physical processes used to build systems to act as computational states and operations.

Geroch describes a version of this strategy in his discussion of plausible limits to quantum computability that the quantum computational literature implicitly assumes. He argues that unitary operators that rotate a quantum state through a classically noncomputable angle, though they are allowed by the laws of quantum mechanics, should not be permitted in a model of quantum computation, because a classical, stepwise method of building systems that perform these operations would require hypercomputation. (See chapter 8 of (Geroch 2009).) Though finite quantum systems can execute classically noncomputable unitary operations, Geroch argues for their exclusion from quantum computation based on an appeal to classical computational principles. Similarly, Piccinini assumes this classical computational strategy in his use of physical constructibility, as seen in his discussion of why "unconstrained appeals to real-valued quantities" should not be considered part of epistemically useful physical computation: "[T]here is no reason to believe that a finite observer can use the Turing-uncomputable operations... to compute in the epistemological sense that motivates CT [the Modest Physical Church-Turing thesis] in the first place..." (Piccinini 2011)

One reason for adopting the classical starting point for buildability – at least regarding the exclusion of real-valued quantities – noted by Piccnini, is that "There is no reason to believe that... unbounded precision is available to a finite observer" (Piccinini 2011). Relatedly, the original arguments for what Piccinini calls the "Mathematical Church-Turing thesis"[5] – that "any function that is computable by following an effective procedure is Turing computable" – may bolster this position. Particularly, Turing argues in favor of the notion of computability given by Turing machines on the grounds that they limit the set of computational states and operations to those corresponding to what a human (with pen and paper) would be able to use (Turing 1936-7) (reprinted in (Turing 1965)). This suggests that Turing machines and Turing (classical) computable processes probably capture the types of processes that (human) finite observers can reliably execute, and thus are likely the best starting point for developing physical computation in some physical theory, even though the classical computational processes were not developed to most optimally leverage the properties of that particular physical

[5]Not to be confused with either the Modest or Bold Physical Church-Turing Thesis

theory.

Also, appealing to classical computational operations is a particularly salient strategy when investigating physical theories like quantum mechanics or relativity that are supposed to subsume physical theories that give rise to classical computation, such as classical physics. We can often assume that a physical theory at least allows classical finite state automata, which then provide us with information on some, if not all, B and O systems we may be able to build. This strategy also allows us to utilize an enormous amount of information about the processes we can use to build B and O systems in a theory, as we have a robust understanding of what classical computational processes look like and what they can do.

Though the strategy of appealing to classical computational processes is strong, its limitations should be noted. Most significantly, it is not a strategy that emerges from considerations of a particular physical theory at hand. Restricting ourselves to computational systems that can be built using classical processes and not theory-specific processes may unnecessarily limit the models of computation that we can build relative to a theory. Operations which the theory allows but which cannot be implemented by systems built using classical procedures cannot be used, and there may exist some systematic theory-specific methods for finite observers to build systems which carry out these operations. There may be an operating system O that can be reliably build by finite observers but could not be built by finite observers limited to construction processes using only classically computable operations. So while the classically-grounded approach may be a powerful starting point in theories that are amenable to classical computational operations, it may also unduly constrain the models of computation we can build relative to a theory because it may not allow us to fully leverage the unique properties of that theory. *Ruling out building process that cannot emerge from classically computable steps for finite observers in any physical theory is a significant assumption that we should make consciously.*

These problems are particularly important because *we live in a universe that is not classical.* To better utilize the particular properties of a physical theory, such as quantum mechanics, we may find we want to select some plausible theory-specific operations as our starting point instead of or in addition to the classical operations. The bit systems B and operator systems O that we deem credible will thus be whatever can be built using some preselected set of operations allowed by the theory's dynamics, similar to the classical solution but not necessarily reliant on concepts or operations from classical computing models.

The non-classicality of our world already throws the Modest Physical Church-Turing thesis in doubt; for instance, quantum mechanics allows a genuine random number generator by using multiple measurements of superposition states. Perhaps we will find that we are able to harness quantum operations in a way that surpasses the limits that Geroch's classical building approach, or perhaps some other physical theory will allow us to surpass this classical framework for building processes. In any case, we should be mindful about the assumptions in our discussions of physical computation that we may want to drop when exploring epistemically useful computation in other physical theories.

5 Remarks

Tying our accounts of physical computation to what would in fact be epistemically useful for finite observers requires restrictions on what models of physical computation are considered. Piccinini gives a Usability Constraint which is meant to achieve this goal. But part of this constraint lacks adequate conceptual grounding, in lieu of which Piccinini has implicitly appealed to classical computing notions to determine which physical processes can be included in epistemically useful models of computation. Classical computation so deeply underlies discussions of computation that this assumption may be intuitive to the point of escaping detection, but we should be aware when we are making it in discussions of physical computation, especially as *the world we inhabit is not, in fact, classical.*

As Aitken and Barrett note, what is physically computable depends on the selection of a model of computation relative to a particular physical theory, so we may hope that our understanding of what is buildable, and hence potentially usable for finite observers, in a physical theory will emerge from the physical theory and its descriptions of finite systems. However, attempting to derive a notion of buildability directly from an account of finite systems in a physical theory succumbs to a bootstrapping problem, in which our notion of what is "buildable" is meant to allow us to determine which physical processes finite observers can reliably implement (as part of a computation), but determining what counts as buildable requires that we already know which physical processes finite observers can reliably implement (as part of building the components of computing systems). We are then left with a choice of how to select what physically possible states and operations we will include in building processes. In this situation, the classical-based approach used by Piccinini and explicitly stated by Geroch is an especially strong starting point for investigating what models of computation could be epistemically useful in a physical theory, though we should be that it is one of many possible selections we can make when discussing physical computation. While a classically-founded notion of buildability is a strong default starting point for investigating what models of computation could be epistemically useful in a physical theory, it limits what physical processes are included without attending to the particular properties of a given physical theory, and adopting it without being aware that it is an assumption we're making may lead us to pick sides in debates regarding physical computation without adequate conceptual grounding.

References

[1] Aitken, Wayne, and Jeffrey A. Barrett. "A Note on the Physical Possibility of Transfinite Computation." *British Journal of Philosophy of Science* 61 (2010): 867-874.

[2] Aitken, Wayne, and Jeffrey A. Barrett. *On the Physical Possibility of Ordinal Computation*. 21 Mar. 2009. www.socsci.uci.edu/ jabarret/bio/publications/ncompSubmit.pdf.

[3] Arrighi, Pablo, and Gilles Dowek. "The Physical Church-Turing Thesis and the Principles of Quantum Theory." *International Journal of Foundations of Computer Science* 23, no. 05 (2012): 1131-1145. 10.1142/s0129054112500153.

[4] Cotogno, Paolo. "Hypercomputation and the Physical Church-Turing Thesis." *The British Journal for the Philosophy of Science* 54, no. 2 (2003): 181-223. 10.1093/bjps/54.2.181.

[5] "Schematizing the Observer and the Epistemic Content of Theories." *arXiv* (2020). https://arxiv.org/abs/1903.02182v3.

[6] Dawson, C. M., and M. A. Nielsen. "The Solovay-Kitaev Algorithm." *Quantum Information and Computation* 6, no. 1 (2006): 81–95. https://doi.org/10.26421/qic6.1-6.

[7] Geroch, Robert. *Perspectives in Computation*. University of Chicago Press, 2009.

[8] Hogarth, Mark. "Non-Turing Computers and Non-Turing Computability." *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association* 1994, no. 1 (1994): 126-138. 10.1086/psaprocbienmeetp.1994.1.193018.

[9] Kitaev, A. Yu. "Quantum Computations: Algorithms and Error Correction." *Russian Mathematical Surveys* 52, no. 6 (1997): 1191–1249. https://doi.org/10.1070/rm1997v052n06abeh002155.

[10] Müller, Vincent C. "Pancomputationalism: Theory or Metaphor?" In Ruth Hagengruber and Uwe Riss (eds.), *Philosophy, Computing and Information Science*, 213-221. Pickering & Chattoo, 2014. Available on PhilArchive: https://philarchive.org/archive/MLLPTO-2.

[11] Piccinini, Gualtiero. "Computation in Physical Systems." In *Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta, Spring 2017 ed. Stanford University, 2017. Accessed 4 June 2023. plato.stanford.edu/entries/computation-physicalsystems/.

[12] Piccinini, Gualtiero, and Neal G. Anderson. "Ontic Pancomputationalism." In *Physical Perspectives on Computation, Computational Perspectives on Physics*, edited by Michael E. Cuffaro and Samuel C. Fletcher, 23-37. Cambridge University Press, 2018.

[13] Piccinini, Gualtiero. "Computational Modelling vs. Computational Explanation: Is Everything a Turing Machine, and Does It Matter to the Philosophy of Mind?" *Australasian Journal of Philosophy* 85, no. 1 (2007): 93-115. 10.1080/00048400601176494.

[14] Piccinini, Gualtiero. *Physical Computation: A Mechanistic Account.* Oxford University Press, 2018.

[15] Piccinini, Gualtiero. "The Physical Church-Turing Thesis: Modest or Bold?" *The British Journal for the Philosophy of Science* 62, no. 4 (2011): 733-769. 10.1093/bjps/axr016.

[16] Shagrir, Oron, and Itamar Pitowsky. "Physical Hypercomputation and the Church-Turing Thesis." *Minds and Machines* 13, no. 1 (2003): 87-101. 10.1023/a:1021365222692.

[17] Stein, Howard. "Some Reflections on the Structure of Our Knowledge in Physics." In *Logic, Methodology and Philosophy of Science IX, Proceedings of the Ninth International Congress of Logic, Methodology and Philosophy of Science*, 1995, pp. 633–655. https://doi.org/10.1016/s0049-237x(06)80067-4.

[18] Turing, Alan M. 1936–7. "On Computable Numbers, with an Application to the Entscheidungsproblem." *Proceeding of the London Mathematical Society* 42(1): 230–265.

[19] Turing, Alan M. "On Computable Numbers, with an Application to the Entscheidungs Problem." In M. Davis (ed.), *The Undecidable*, Ewlett, NY: Raven, 1965, pp. 116-154.

[20] Ziegler, Martin. "Physically-Relativized Church–Turing Hypotheses: Physical Foundations of Computing and Complexity Theory of Computational Physics." *Applied Mathematics and Computation* 215, no. 4 (2009): 1431-1447. 10.1016/j.amc.2009.04.062.