A Review of Physical Computing*

Ryan Scott†
Virginia Tech, 925 Price Fork Rd, Blacksburg, VA 24060

(Dated: October 9, 2025)

The theory of computability has had a vast impact on a wide range of physical theories. In this paper, we discuss how non-relativistic classical and quantum systems are related to the ideas of computation. We give a pair of examples followed by a review of quantum computation. We discuss how these ideas give rise to the field of quantum simulation, and finally we speculate on some new areas of investigation based on the foundations of the theory.

I. INTRODUCTION

Although the history of computing can be traced back to calculations done by algorithmic methods to the time of ancient Egypt [1] (perhaps even older, depending on the definition of computation), it didn't become a properly rigorous discipline until the turn of the twentieth century, motivated by questions regarding the foundations of mathematics, and the (complete) formalization of arithmetic. Broadly, this work can be traced back to Gottlob Frege [2], who intended by means of logical inferences to formalize arithmetic. While this predated David Hilbert's *Foundations of Geometry* [3], it wasn't until that work that the discipline of mathematics began to consider the formalization or arithmetic in earnest–the formalization of geometry offered great hope to the field to be able to effect a correlated action in the discipline of arithmetic (and thereby 'solve' mathematics by showing it to be complete). Among the major players of this time were Hilbert, Frege, Georg Cantor, Bertrand Russell, Ludwig Brouwer, and others[3], [4], [5], [6], [7], [8]. The project continued for about 30 years with seemingly little progress (although some major ideas would later influence computer science). In 1931, Kurt Gödel published his famous *On Formally Undecidable Propositions of Principia Mathematica And Related Systems.* [9] The paper effectively made hopeless the completion of the project which Hilbert and others had in mind. Gödel's argument leveraged a definite computational process that amounted to the construction of a mapping between the set of all claims in arithmetic and the natural numbers. He then used a form of Cantor's famous diagonalization argument in order to prove that arithmetic cannot possibly be completed–which means that every system which would list all possible claims about it would end up making a claim which is both false and cannot be proven to be true or false. Concurrent with this, Alonzo Church and Alan Turing (independently) developed methods of formalization of calculating in the context of mathematical claims [10], [11]. This gave birth to the ideas of the celebrated lambda calculus (due to Church), the famous Turing machine formalism, and the field of modern computer science.

50 years later, Richard Feynman, Paul Benioff, David Deutsch, and a few others, sought to extend the formal systems developed in the context of mathematics to physics by considering what it would mean to redefine the idea of computation in the context of physical systems [12], [13],[14]. Feynman sought to simulate complicated physical systems by computational methods. Benioff considered what it would mean to develop systems describing the evolution of a wave function according to the Schrödinger equation by means of Turing machines. This work culminated in Deutsch's 1985 paper *Quantum theory, the Church-Turing principle and the universal quantum computer* in which Deutsch redefines the notion of a computation as well as postulating a correlated principle by which the ideas of computation can be applied to physical systems. In the spirit of that paper, we will review Deutsch's idea, consider some specific examples (both classical and quantum), and finally consider some potential avenues for future research based on this redefinition and an attempt to merge the original idea of computation (due to Church, Turing, etc) with it.

II. THE FEYNMAN-DEUTSCH CONJECTURE

We begin by defining a *computation* as a map which time-evolves an arbitrary physical system Σ from some initial state $\sigma(t)$ to some final state $\sigma(t+\delta t)$. This time-step δt is postulated to partition some segment of time so that we can regard the evolution as occurring in a sequence of N steps. We note that the set to which Σ belongs is ill-defined (if the object to which it belongs is really a set at all); at best, we can only assert that Σ has the property of 'being physical,' but this is of course a circular definition. (The definition of a physical system is thereby contingent upon the idea of physicality–but the property of physicality presupposes systems which themselves are 'physical' in the methodology of our current usage. We could attempt to resolve this circle by defining physicality in terms of more primitive notions, but then we only postpone the problem. This issue is inevitably traceable to the problem of perception and measurement, but that would take us too far from the present discus-

* Virginia Tech Physics Department
† Email: rscott1915@vt.edu

sion.) Consequently, we have to take this property as being a primitive notion. This will be important later when we attempt to consider a connection back to the mathematical expression of computation. In particular, we shall have to operate under the point of view that we cannot correlate an arbitrary physical system to an arbitrary mathematical computation because the former consists of a collection of objects which is ill-defined, and therefore the related map which sends this collection to a (proper) set of mathematical objects is also ill-defined. It is because of this that we need to introduce a *postulate* which is essentially due to Deutsch and Feynman:

> *The Feynman-Deutsch Conjecture: There exists a 1-1 correspondence between the class of constructable mathematical objects and the class of finite physical systems whose states are specified as discrete points in time*

This postulate enables us to set up a formal system which can be used to deduce properties about physical systems on the basis of computable states. Furthermore, if this correspondence is really 1-1, then we also can use experimental results to empirically investigate the foundations of mathematics. See below for more on this. We now turn to the formal details this postulate enables us to set up.

Let $\overline{\Sigma}$ denote the class of physical systems, and $\Sigma \in \overline{\Sigma}$ an arbitrary physical system. We suppose that to each Σ there is a Hamiltonian H_Σ which specifies a full spectrum $\{\sigma\}$ of states that completely characterize the time evolution of the system. The Hamiltonian may evolve in time, but for simplicity we will begin with the time-independent case. Define a computation to be a sequence of functions

$$\{f_n\} : \sigma(t + n\delta t) \xrightarrow{f_n} \sigma(t + (n+1)\delta t) \tag{1}$$

such that the sequence satisfies the initial condition $\text{dom}[f_0] = \sigma(t)$, where $\sigma(t)$ is some initial state. In general, the evolution of the dynamical system cannot be terminated. Consequently, the final state must be specified so that one can determine the result of some computation. Therefore, we must also define the result of the calculation to be $\text{cod}[f_N] = \sigma(t + (N+1)\delta t)$. Therefore, we see that the definition of a physical computation χ amounts to a finite sequence of functions:

$$\chi(t, \delta t) : \sigma(t) \xrightarrow{f_0} ... \xrightarrow{f_N} \sigma(t + (N+1)\delta t) \tag{2}$$

Since the states will be time-ordered (therefore making the sequence of computations into a poset), and because there must physically be an initial and terminal state, we have in time a least and greatest element associated to every computation. The arrows must also be composable on physical grounds. Therefore each computation χ will be a lattice. Clearly this definition requires that each of the discrete states is measurable (at least in theory). Given the structure of (1), we can deduce that the f_n must have the physical interpretation of evolving the system in time.

III. EXAMPLES

We now consider a pair of examples. The first shall be a classical case, and the second a quantum case.

A. Classical System: Central Potential

Consider a mass m moving in an attractive potential[15]

$$V(r) = -\frac{\alpha}{r} \tag{3}$$

The standard result using Hamiltonian mechanics, one obtains the equations of motion

$$\begin{aligned} \dot{p}_r &= -\frac{\alpha}{r^2}, \\ \dot{r} &= \frac{p_r}{m}, \\ \dot{p}_\phi &= 0 \end{aligned} \tag{4}$$

which leads to the usual elliptical orbits. Now consider the case of uniform circular motion–in that case, we have a constant radius, and we can imagine partitioning the circle into N arcs of equal length, and the length of the arc stands in relation to the time it takes to trace out the curve. In this case, if the period is T, then the time segments are given by $\delta t = T/N$. If the angular velocity is constant, we obtain the usual $R\omega = \delta r/\delta t$, so that $\delta t = \delta r/R\omega$, so that we can use the periodicity to calculate elements of the group $\mathbb{Z}/N = \{\overline{0}, ..., \overline{N-1}\}$, where the elements $\overline{k}$ are the equivalence classes defined by $k\mathbb{Z} = \{0, \pm k, \pm 2k, ...\}$. This is one of the most basic methods of encoding computable structures into physical systems, and likely one of the oldest. Indeed, this precise line of reasoning is *how we keep a calendar!* See figure 1. Under this view, we have an assignment which sends the physical system characterized by the (non-relativistic) central potential to the set of integers mod N–whereby we immediately see that the latter can be calculated by physical means.

B. Quantum System: Double Slit Distribution

Because quantum mechanical systems allow for continuous interference, encoding discrete information can pose certain challenges–since ultimately we will be measuring a probability *distribution.* Consequently, one can compose maps like so:

$$\overline{\Sigma} \xrightarrow{f} [0,1] \xrightarrow{g} \mathbb{Z}/N \tag{5}$$

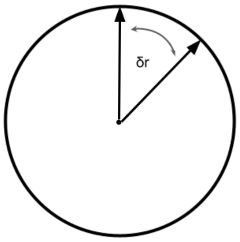


FIG. 1. Model of a Celestial Calendar

where $[0, 1]$ is the (closed) unit interval. Here we intend to perform measurements using the state vector (rather than using eigenvalues only). The latter is typically harder (as opposed to their expectation values), as it requires isolation of a physical system. In the former, we already have a host of robust experiments at hand which we can use. In this case, we have in mind the double slit experiments using electrons; we can then partition the screen as a classical object as we did above, and based upon the where an electron strikes (associated with some probability) we can map the outcome to some $\overline{n} \in \mathbb{Z}/N$. The results are standard and can be derived on the basis of assuming a wave-like nature associated with the electron wave function [16]. (Of course, this is where the function gets its namesake.) Recall that the locations of the constructive interference patterns are given by:

$$d \sin\theta = m\lambda, \qquad m = 0, \pm 1, ... \tag{6}$$

where θ is the angle from the perpendicular connecting the double slit and the screen, d is the slit separation, and λ is the wavelength. The corresponding intensity plays the role of the probability distribution, and is given by:

$$I(\theta) = I_0 \cos^2\left(\frac{\pi d \sin\theta}{\lambda}\right) \cong I_0 \cos^2\left(\frac{\pi d\theta}{\lambda}\right) \tag{7}$$

We can then map the peaks, valleys, or any other periodic feature, to some finite set, and therefore encode the distribution in some computational structure. In this case, one simple example would be to regard each measurement of an electron to correspond to a single time step. Since the velocities of the electrons are all approximately equal, and we suppose that the width of the distribution is much less than the distance to the screen (which is bound up in the approximation $\sin\theta \cong \theta$), the measurements can be partitioned by some time interval δt as we did before. In this case, if we form the map [17]:

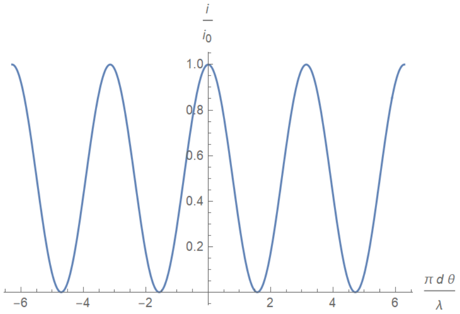


FIG. 2. Dimensionless Interference Pattern

$$\begin{aligned} \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] &\to 0 \\ \left[\frac{\pi}{2}, \frac{3\pi}{2}\right] &\to 1 \\ \left[-\frac{\pi}{2}, -\frac{3\pi}{2}\right] &\to -1 \\ &... \end{aligned} \tag{8}$$

then we can form sequences of the first $2n+1$ integers, the lengths of which would correspond to the total duration over which the experiment is run. Since the peaks are all equally likely in theory, we would expect that every finite sequence is equally likely, so that the probability of measuring some sequence is just $1/(2n+1)!$. In this case, we have a physical process which calculates a sequence of digits with some probability. This is nothing other than a *random number generator!*

C. Relativistic Phenomena

Although we will not introduce an explicit example here, this method of encoding is not restricted to non-relativistic phenomena. For example, one could use the fact that the property of causality in Special Relativity partitions spacetime into a pair of equivalence classes to map relativistic events to the set $\mathbb{Z}/2 = \{\overline{0}, \overline{1}\}$, which can be used to set up a Boolean logic on the basis of measurements [18]. Because events are causal if and only if their separation is timelike, one can imagine a pair of observers whose origins are simultaneous in time but whose separation is spacelike at time $t = 0$ for both frames, but whose light cones will overlap at some time $t' > t$. Then comparison of a the set of all measurable events will enable the setting up of unions and intersections of events, which will correspond to disjunction and conjunction respectively. Using this, the set of all propositions in two-valued logic could be encoded using a relativistic framework, which in turn would correspond to calculations in the field $\mathbb{Z}/2$.

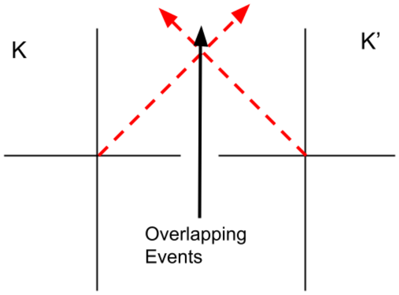


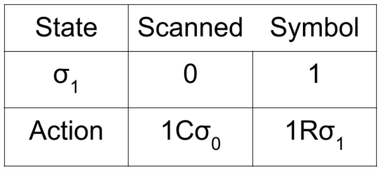
FIG. 3. Two inertial frames with overlapping events.

IV. QUANTUM MECHANICAL SIMULATIONS

Now that we've seen some basic examples regarding what it means represent computability by physical systems, we can turn to the more enticing aspect of physical computing, and quantum computing specifically–simulating complex physical systems by means of more primitive interacting systems which we will regard as performing computations. In order to discuss this, we need the more formal machinery of quantum computation, which we examine now.

A. A Brief Review of Quantum Computing

Although the Feynman-Deutsch conjecture radically redefines the notion of computation, the formal system which physicists set up in order to extend the notion of computability to quantum systems is remarkably similar to the methods of Turing. Loosely speaking, a Turing machine is defined to be a map from a product of sets to a product of sets [5], [19]. More formally, a Turing machine is to be thought of as a measuring head which reads a two-way, infinitely long tape divided into cells which can store symbols, and consists of a finite set of *states* $\{\sigma_1, ..., \sigma_n\}$, a set of symbols (which we shall take to consist of only two elements, 0 and 1) which is called its *alphabet*, and a set of output actions it can take based on its instructions. The set of actions in this case is: 1) do nothing; 2) move the tape left and update its internal state; 3) move the tape right and update its internal state; 4) write a 1 and update its internal state; 5) write a 0 and update its internal state. The *name* of the machine, denoted $\mathfrak{G}$, is the function the machine computes. That is, one feeds it a sequence of 0s and 1s, and it yields a sequence of 0s and 1s based on the instructions it holds. One can build a table of instructions for a given input in order to get a sense for how the machine calculates.



State	Scanned	Symbol
σ_1	0	1
Action	$1C\sigma_0$	$1R\sigma_1$

FIG. 4. A simple Turing machine, $\mathfrak{G}$, which calculates $y = x + 1$.

One specifies the machine by specifying its table–the analog of which in the case of physical computing is the simulation's algorithm, which will be given in terms of the physical states through which the machine evolves. This means, in the case of quantum physical computing, one need only specify the Hamiltonian, and the solution to the Schrödinger equation.

An example of a simple Turing machine is given in figure 4. In this case, a special symbol, σ_0, is reserved for the termination state, so the actions the machine takes in that state are not specified. If the machine is in σ_0, it will halt. Otherwise, it will take continued action. In this case, the machine will read either a 0 or a 1. If the machine reads a 0, it will replace it with a 1, stay where it is, and halt. Otherwise, if it reads a 1, it will leave the 1 unchanged, move to the right, and stay in the same state. The output of this machine is just $y = x + 1$.

The generalization to the quantum case should be relatively clear from this simple example. If the device is to be properly quantum, its time evolution between states will be given by a unitary operator which will in general allow evolution into superpositions of states. For example, in the above case, one could imagine a head entangled with a sequence of noninteracting spins. If the head 'measures' a spin-up, it moves forward; if it 'measures' a spin-down, it applies some magnetic field which flips the spin and halts. Then some measuring apparatus in the lab could be used to measure the whole chain of spins plus the location of the head after N time steps. In practice, this method of analysis is rather clunky for quantum computations. Instead of the typical Turing representation, we use a quantum circuit model which can be read 'left to right' as if the bits were flowing through the gates. Each gate corresponds to a composition of unitary operators, and the output is taken to be the set of quantum states at the end of the circuit [20].

B. Simulating Quantum Systems

We suppose that the system we will simulate will be an isolated system, evolving coherently in time according to some Hamiltonian H. H is now allowed to evolve in time, but we suppose it has the form

$$H = \sum_{j}^{N} H_j \tag{9}$$

Any Hamiltonian consisting of local interactions can be written in this form, which makes it particularly attractive [21]. The simulation works by evolving the system in time by means of local interactions over small time steps. We can then approximate the time evolution operator according to

$$e^{iHt} \cong \left(e^{iH_1 t/n} ... e^{iH_N t/n}\right)^n \tag{10}$$

where n is the total number of desired time steps. This form is accurate to within $n(e^{iHt/n} - 1 - iHt/n)$, and can be made arbitrarily small, so that the approximation is arbitrarily accurate. Turning to an interpretation of equation (10), we see that for each instance of the product of exponents, the system time evolves through each of the (independent) components of the net Hamiltonian– for example, given three spins in a chain, the first Hamiltonian may evolve an interaction between the first two spins, while the second may evolve an interaction between the second two spins. This is then repeated until the desired time length has been reached.

V. PROSPECTS

One of the most important questions in the original research done by Church, Turing, and others was regarding how computation could inform methods of proof and its broader implications for mathematics as a whole. The method developed there was manifestly constructive, and abstracted all content out of the symbols it manipulated, and it was precisely this property that allowed Gödel to prove his celebrated incompleteness theorem. Because this constructive property is removed by Deutsch in his 1985 paper, it would be interesting to study the new class of functions which quantum computers characterize. What do they imply for the foundations of mathematics? Certainly the mathematician would never think physical experiments could impose boundaries on his knowledge. Furthermore, does the property of incompleteness persist in the case of quantum computers? Because one is bound to the laws of physics, it is no longer clear that one can abstract out all content and therefore deduce the incompleteness of the class of functions which the quantum computers can feasibly access. Regarding this however, the few algorithms we have do still behave *algorithmically*, albeit they leverage the property of superposition. For example, the quantum part of Shor's algorithm still proceeds by means of syllogism (i.e. a finite sequence of logical steps based on implication). Based on this, it appears as though the class of accessible mathematical objects (via physical construction) is broader than was supposed by Brouwer and others at the turn of the twentieth century. Consequently, can we construct mathematical systems whose objects are inherently quantum? What would such a mathematical system entail?

VI. CONCLUSION

In this brief review, we attempted to give an introduction to the idea of computation in the context of physical systems. We introduced the Feynman-Deutsch conjecture, showed how it generalizes the original idea of computation, and considered its application to a collection of typical physical systems. After that, we looked at its use in the realm of quantum simulation, discussing specifically how it is represented there and how simpler systems can be used to create more complex systems. Finally, we took at look at some prospective questions which may guide further research.

VII. ACKNOWLEDGEMENTS

We'd like to thank Thomas O'Donnell and the Virginia Tech physics department.

[1] D. M. Burton, *The History of Mathematics: An Introduction, Sixth Edition* (McGraw-Hill, 2007).

[2] G. Frege *et al.*, Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought, From Frege to Gödel: A source book in mathematical logic **1931**, 1 (1879).

[3] D. Hilbert, *The foundations of geometry* (Open court publishing Company, 1902).

[4] J. Von Neumann, An axiomatisation of set theory, (1925).

[5] S. C. Kleene, Introduction to metamathematics, (1952).

[6] G. Cantor, Letter to dedekind, (1899).

[7] A. N. Whitehead and B. Russell, *Principia mathematica to* 56*, Vol. 2 (Cambridge University Press, 1997).

[8] L. E. J. Brouwer, On the significance of the principle of excluded middle in mathematics, especially in function theory, van Heijenoort vH67 , 334 (1879).

[9] K. Gödel, *On formally undecidable propositions of Principia Mathematica and related systems* (Courier Corporation, 1992).

[10] A. Church, A formulation of the simple theory of types, The journal of symbolic logic **5**, 56 (1940).

[11] A. M. Turing *et al.*, On computable numbers, with an application to the entscheidungsproblem, J. of Math **58**, 5 (1936).

[12] R. P. Feynman, Simulating physics with computers, in *Feynman and computation* (CRC Press, 2018) pp. 133–153.

[13] P. Benioff, Quantum mechanical models of turing machines that dissipate no energy, Physical Review Letters **48**, 1581 (1982).

[14] D. Deutsch, Quantum theory, the church–turing principle and the universal quantum computer, Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences **400**, 97 (1985).

[15] L. D. Landau and E. Lifshits, *Mechanics*.

[16] C. C. Gerry and K. M. Bruno, *The quantum divide: why Schrödinger's cat is either dead or alive* (OUP Oxford, 2013).

[17] Strictly speaking, this map is not well-defined, since the supremum and infimum of each set maps to two integers. This can easily be fixed formally by sending the first occurrence of any of the given fractions in the intervals to the integer that interval maps to, and then taking subsequent intervals in which the fraction appears as half-open on the side where the fraction is appearing an additional time.

[18] R. M. Wald, *General relativity* (University of Chicago press, 2010).

[19] S. C. Kleene, *Mathematical logic* (Courier Corporation, 2002).

[20] D. E. Deutsch, Quantum computational networks, Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences **425**, 73 (1989).

[21] S. Lloyd, Universal quantum simulators, Science **273**, 1073 (1996).