The Attralucian Essays: Exploring the Finite

First Edition

 Typeset in LaTeX

The Attralucian Essays

On Computability: A Geofinitist Computability Thesis

Re-examining the Church-Turing Thesis

Kevin R. Haylett

The Church-Turing Thesis: A Geofinitist Reimagining

A Computational Conundrum

Picture a scribe in an ancient library, quill in hand, meticulously calculating the squares of numbers on a parchment scroll. Each step is precise, mechanical, following a clear set of rules. Now imagine a modern supercomputer humming away, crunching the same numbers. Intuitively, we feel both are doing the same thing: computing. But what is computation, really? Can we pin down what it means for something to be "computable" in a way that bridges the scribe's quill and the supercomputer's circuits?

This question lies at the heart of the Church-Turing Thesis (CTT), a cornerstone of computer science that dares to define the limits of what any machine—or mind—can calculate. Yet, as we'll see, this bold idea rests on idealized assumptions that don't quite survive the messy reality of the physical world. Enter Geofinitism, a framework that reimagines computation as a measurable, finite process, revealing new clarity and possibility.

The Church-Turing Thesis: A Bold Claim

In the 1930s, as mathematicians grappled with the foundations of logic, two brilliant minds—Alonzo Church and

Alan Turing—offered answers to a profound question: what can be computed? Church's λ-calculus and Turing's hypothetical "machine" with its infinite tape provided formal ways to describe mechanical procedures. Alongside colleagues like Stephen Kleene, Kurt Gödel, and others, they showed that these systems—despite looking wildly different—could compute the same set of functions.

This convergence birthed the Church-Turing Thesis: any function a human could compute by following a clear, step-by-step recipe (an "effectively calculable" function) could be handled by a Turing machine, a λ-calculus expression, or a recursive function. It's not a theorem you can prove; it's a bold hypothesis, a bridge between our intuitive sense of computation and the formal systems we build.

But the thesis stretches beyond mathematics. Some wonder if it applies to the physical world—can everything computable in our universe be captured by these models? Others ask about efficiency: what can be computed practically within reasonable time and resources? These questions expose cracks in the thesis's idealized foundations, especially when we confront the limits of real-world computation: noisy circuits, finite memory, and the relentless tick of time. Geofinitism offers a way to rethink these cracks, not as flaws, but as opportunities to ground computation in measurable reality.

Applying Geofinitism: Where the Fictions Falter

Geofinitism, with its five pillars, challenges the idealized assumptions of the Church-Turing Thesis, replacing them with a framework rooted in finite, measurable structures. Let's explore how each pillar reshapes our understanding of computation.

Pillar 1: Computation as a Journey

The classical view treats computation as a linear process: a Turing machine chugs along an abstract, infinite tape, and a function is simply "computable" or not. Geofinitism disagrees. Computation is a trajectory through a space of resources—time, memory, energy, precision. This space, called a manifold M, defines what's possible. Ignoring this geometry risks overgeneralizing, pretending computation works the same way in a quantum chip as in a human brain. Instead, Geofinitism embeds computations as paths in M, tying "computability" to the resources you can actually muster.

Pillar 2: The Imperfection of Symbols

In the classical thesis, symbols on a Turing machine's tape are perfect, steps are exact, and precision is unlimited. But real computers don't work like that. A bit flipped in a processor might be misread due to electrical noise; a quantum state might wobble. Geofinitism acknowledges this messiness. Symbols and states are approximations, subject to uncertainties in timing (σ_t) or storage (σ_s). By modeling computation with bounded errors, we make it measurable, rooted in the real world's imperfections.

Pillar 3: The Cascade of Scales

Computation isn't a single, tidy process. It's a cascade, from the flicker of electrons in a chip to the high-level algorithms we write. The classical thesis assumes one description fits all scales, but Geofinitism sees computation as layered. Each layer—hardware, microarchitecture, algorithm—adds constraints and uncertainties. We model this with a recursive formula:

$$C_n = f(C_{n-1}, \Delta r),$$

where C_n represents computation at one scale, built on the layer below (C_{n-1}) with resource increments Δr.

Pillar 4: A Useful Fiction, Not a Cosmic Truth

The Church-Turing Thesis often feels like a grand, universal truth: computation is this, forever and always. Geofinitism calls this a fiction—not a lie, but a story that only holds where we can measure it. The thesis's claim to "effective calculability" stretches beyond what we can test, making it unprovable. Instead, Geofinitism treats it as a useful fiction, valid only within stable, measurable boundaries.

Pillar 5: Embracing the Finite

The classical thesis assumes infinite time, space, and precision—an endless tape, an eternal clock. But reality imposes hard limits: clock speeds stall, memory fills up, energy runs dry. Geofinitism insists on finite quanta—minimum units of time (δt) and space (δs)—and reasons within those bounds.

A Formal Lens: Geofinitist Computability

To make this concrete, Geofinitism offers a new way to define computability. Imagine a function f (say, squaring a number) with input x. We define a computability functional:

$$C_f(x) = \frac{\Delta O}{\delta t \cdot \delta s} + \sigma_c(x, \delta t, \delta s),$$

where ΔO measures progress toward the correct output (e.g., verified bits of $f(x)$), δt and δs are the smallest units of time and space, and σ_c captures uncertainties like hardware noise or decoding errors. A function is "Geofinitist-computable" if $C_f(x)$ exceeds a threshold θ along a stable path in the resource manifold M.

For uncertainty, we might model:

$$\sigma_c(x, \delta t, \delta s) = k\sqrt{\text{Var}(O \text{ on } [t, t+\delta t] \times [s, s+\delta s])} + k_t\sigma_t + k_s\sigma_s,$$

where constants k, k_t, and k_s weigh different sources of error. Across scales, we aggregate:

$$C_f(x) = \frac{1}{K}\sum_{i=1}^{K} C_f^{(i)}(x),$$

ensuring each layer contributes to the whole.

Where the Thesis Breaks—and How Geofinitism Rebuilds

The classical Church-Turing Thesis falters when it assumes infinite resources, perfect symbols, and a one-size-fits-all view of computation. Sending time or space to infinity leaves the measurable world behind (Pillar 1). Ignoring noise violates Pillar 2. Oversimplifying scales misses Pillar 3. Claiming universal truth oversteps Pillar 4. And pretending resources are limitless defies Pillar 5.

Geofinitism doesn't just point out these flaws—it offers a fix. By enforcing finite quanta ($\delta t, \delta s > 0$) and using measurable trajectories, it transforms the thesis into an engineering principle. Different models—Turing machines, λ-calculus, recursive functions—produce equivalent paths in M for computing a function, but only within stable, finite regimes. This makes the thesis practical, testable, and grounded.

The Geofinitist Payoff: A New Way to Compute

Imagine an engineer comparing how a function like x^2 is computed on a classical computer versus a quantum one. They measure $C_f(x)$, tracking resource use and uncertainties across layers—circuits, microarchitecture, algorithms. Where trajectories align, the Church-Turing Thesis holds, showing that different systems compute the same thing. Where they diverge, physical limits (like quantum noise or energy costs) reveal the boundaries of computability.

By reframing computation as measurable paths in a finite world, Geofinitism liberates us from the thesis's lofty abstractions. It lets us quantify computability—how much time, space, or energy a task demands—and compare systems rigorously. We can decide what's computable not by philosophical fiat, but by what we can measure and build. This is the Geofinitist revolution: a vision of computation that's not just theoretical, but alive, practical, and ready to shape the future.

Context.

The narrative reframes the Church–Turing Thesis (CTT) as an empirical regularity about what *finite, physical* procedures can do, not a metaphysical identity. In $\mathbb{M}$, algorithms are measured processes with uncertainty and provenance; "effective computability" means reproducible transformation under resource budgets and tolerance.

Measured Procedures (Physical Algorithms). A device/model D with control parameters θ and input $x \in \Sigma^\star$ induces a measured procedure

$$\mathsf{Proc}_{D,\theta}(x) \;=\; \Big(y,\ \varepsilon_y,\ P_D;\ T,\ \varepsilon_T;\ S,\ \varepsilon_S\Big) \;\in\; \mathbb{M},$$

returning output $y \in \Sigma^\star$ with uncertainty ε_y, time T, space S, and provenance P_D (hardware, calibration, noise model, operator protocol). Randomized/quantum devices include a seed/state ω with measured distribution P_ω.

Operational Computability. A (partial) function $f : \Sigma^\star \rightharpoonup \Sigma^\star$ is (τ, δ)-*computable by D on domain $\mathcal{X}$* if

$$\forall x \in \mathcal{X} :\ \Pr_{\omega}\Big[\, d_{\mathbb{M}}\big(\mathsf{Proc}_{D,\theta}(x;\omega),\ f(x)\big) \;\le\; \tau \,\Big] \;\ge\; 1-\delta,$$

within declared time/space budgets and with documented provenance. Here $d_{\mathbb{M}}$ compares the measured output token to the ideal $f(x)$ embedded in $\mathbb{M}$ (exact string, or encoder with tolerance).

Emulation Between Devices. A (universal) reference Turing machine U *emulates* D at tolerance τ and confidence $1-\delta$ if there exists an encoding map E and program p_D such that

$$\forall x \in \mathcal{X} : \Pr_{\omega}\Big[d_{\mathbb{M}}\big(U(p_D, E(x); \omega),\ \mathsf{Proc}_{D,\theta}(x; \omega)\big) \ \le\ \tau \Big] \ \ge\ 1 - \delta,$$

with resource overhead bounded by a measured polynomial $n \mapsto \mathrm{poly}(n)$ on the relevant size parameter. Provenance P_E, P_{p_D} documents the encoding and emulator.

Geofinitist Church–Turing (Finite, Testable Form). For the class $\mathfrak{D}$ of admissible physical procedures (finite, reproducible, locally causal; bounded energy/precision per step),

$$\mathbf{CTT}_{\mathbb{M}}\textbf{:}\ \forall D \in \mathfrak{D}\ \exists\ (U, p_D, E)\ :\ U \text{ emulates } D \text{ at } (\tau, \delta)$$

with poly overhead on all calibrated ranges. Equivalently: any (τ, δ)-computable transformation realized by an admissible device is (τ', δ')-computable by U with τ', δ' controllable by standard error-reduction.

Notes on Models.

- *Randomized:* ω captured in P_ω; U emulates distributions via PRNG or sampling, preserving total variation within τ.

- *Analog:* Signals are discretized at finite resolution; admissibility requires Lipschitz/energy bounds preventing super-TM encodings in noise.
- *Quantum:* For devices obeying finite-precision unitary dynamics and measurement, U emulates to (τ, δ) via standard quantum circuit simulation with poly overhead in time and exponential in qubits only if required by target accuracy; claims of super-TM power must specify how readout exceeds admissible precision.
- *Oracles:* Oracle access is treated as provenance (external service); emulation includes the same oracle, not a claim of computation beyond U.

Verification and Universality Tests. Given a test battery $\mathcal{B}$ of inputs and seeds, report an *emulation residual*

$$\begin{aligned} R_{\text{emu}}(D \Rightarrow U) &= \text{median}_{(x,\omega)\in\mathcal{B}} \\ &\qquad d_{\mathbb{M}}\big(U(p_D, E(x); \omega),\ \mathsf{Proc}_{D,\theta}(x;\omega)\big), \end{aligned}$$

with uncertainty bands. Universality at (τ, δ) is corroborated when $R_{\text{emu}} \leq \tau$ and failure rates $\leq \delta$ across perturbations of x, θ, and environment.

Limits and Abstention. If a device's behavior is *unstable* (non-reproducible beyond tolerance) or requires unbounded precision/energy to specify inputs or read outputs, label it INADMISSIBLE for $\mathrm{CTT}_{\mathbb{M}}$; report UNDERDETERMINED rather than asserting super-Turing capability.

Collapse Note. As measurement uncertainties $\varepsilon \to 0$, budgets extend, and encoders/decoders become ideal, the above reduces to the classical CTT: every effectively calculable function is (Turing-)computable, and every reasonable model is simulable by a universal TM with at most polynomial overhead. Geofinitism keeps finite guardrails: computability and universality are *measured claims* with provenance, tolerance, and resource profiles.

Interpretation. CTT becomes an *auditable universality principle*: what counts as "effective" is precisely what we can reproduce, emulate, and verify within finite tolerance on physical devices. The thesis is upheld to the degree that admissible devices fall into the same emulation class as U under $\mathbb{M}$; departures must specify which guard (precision/energy/causality/reproducibility) they break.