Latent Complexity: A Computable Theory of System Difficulty for Smooth Systems

A Computable Theory of System Difficulty for Smooth Systems

Tamás Nagy, Ph.D.

tnagyphd@gmail.com

Draft • March 2026

Executive Summary (Non-Technical)

How hard is a system? Classical complexity theory answers this for discrete problems — sorting, graph coloring, satisfiability — by counting operations. But most of science deals with smooth systems: fluid flows, financial markets, neural networks, quantum states. For these, the fundamental difficulty is not how many operations a computation takes, but how many numbers you need to describe the system at all.

This paper shows that the Latent framework provides a complete, computable complexity theory for smooth systems. Three extractable invariants — the **Latent Number** ρ (how compressible the system is), the **effective grade** (how complex the interactions are), and the **extraction cost** (how hard it is to find the Latent from observations) — together determine the system's intrinsic difficulty class. We define four complexity classes forming a strict hierarchy:

$$\textbf{L-ENTIRE} \subset \textbf{L-ANALYTIC}(\rho) \subset \textbf{L-BOUNDARY} \subset \textbf{L-SINGULAR}$$

from trivially representable to genuinely hard. We prove that for any system in L-ANALYTIC(ρ), every bounded linear query can be answered in $O(\log(1/\varepsilon)/\log\rho)$ time — resolving the smooth analogue of the "representation implies tractability" question. We prove a Grade–Depth Correspondence: the effective grade of a Latent plays the role of circuit depth, with grade-r capturing exactly those phenomena requiring r-body irreducible interactions. And we prove the central result: unlike classical complexity theory, where the fundamental classification (P vs NP) remains open, the Latent complexity of a specific smooth system is **decidable** — one can compute ρ, measure the grade, and determine the complexity class from data.

The claim is not that classical complexity theory is wrong or irrelevant. It is that for the vast class of smooth systems — which includes virtually everything in physics, finance, engineering, and machine learning — the Latent framework provides a parallel complexity theory that is simultaneously more informative (it gives rates, not just classes) and more practical (the classification is computable, not conjectural).

Abstract

We construct a complexity theory for smooth (analytic) systems based on the Latent framework. We define three orthogonal complexity axes: the **Latent Number** ρ (description complexity: how

many numbers are needed to represent the system), the **effective grade** r^* (interaction complexity: the highest order of irreducible coupling present), and the **extraction cost** $\mathcal{C}_E$ (computational complexity: the cost of recovering the Latent from observations). We introduce four complexity classes — **L-ENTIRE** ($\rho = \infty$), **L-ANALYTIC**(ρ) ($\rho > 1$), **L-BOUNDARY** ($\rho = 1$, $\rho^* > 1$), and **L-SINGULAR** ($\rho^* = 1$) — and prove they form a strict hierarchy (Theorem 1). We prove the **Description–Computation Theorem** (Theorem 2): for any system in L-ANALYTIC(ρ), every bounded linear query is answerable in $O(N^*) = O(\log(1/\varepsilon)/\log\rho)$ operations after a one-time extraction costing $\mathcal{C}_E$, establishing that description complexity upper-bounds query complexity for smooth systems. We prove the **Grade–Depth Correspondence** (Theorem 3): the effective grade of a system's Latent is isomorphic to the depth of a natural "interaction circuit" model, where grade-r gates compute r-linear contractions on Hilbert space. We prove the **Computability Theorem** (Theorem 4): the Latent complexity class of a specific smooth system is decidable — ρ is computable from finite data, grade is measurable, and extraction cost is bounded by the system's dimensionality — in contrast to Kolmogorov complexity (uncomputable) and classical complexity classes (membership unresolved for most problems). We establish formal relationships between Latent complexity and seven classical measures: Kolmogorov complexity, Shannon entropy, VC dimension, Rademacher complexity, Kolmogorov n-widths, communication complexity, and circuit complexity. We show that Latent complexity is strictly more informative than any single classical measure for smooth systems: it provides exponential rates where classical measures give only asymptotic classes, it is computable where Kolmogorov complexity is not, and it captures interaction structure where entropy-based measures are blind. We demonstrate the theory on systems from financial risk (grade-1, $\rho \approx 1.3$: L-ANALYTIC), turbulence (grade-2 with $\rho \to 1$ as $\mathrm{Re} \to \infty$: approaching L-SINGULAR), neural networks (grade-1, ρ from data covariance: L-ANALYTIC with computable model size), and the three-body problem (grade-3, ρ orbit-dependent: L-ANALYTIC for periodic orbits, approaching L-BOUNDARY for chaotic ones). The main theorems are formally verified.

Keywords: complexity theory, Latent Number, analyticity, smooth systems, grade structure, formal verification, Kolmogorov complexity

MSC 2020: 68Q25, 41A25, 47A10, 68Q15, 03D15

---

1. Introduction

1.1 Two Complexity Theories

Computational complexity theory is one of the great achievements of twentieth-century mathematics. Starting from Turing machines and Boolean circuits, it builds a hierarchy of difficulty classes — P, NP, PSPACE, EXP — that classify decision problems by the computational resources needed to solve them. The theory is deep, elegant, and practically relevant: it tells us that some problems are inherently harder than others, and that this hardness is not an artifact of bad algorithms but a property of the problems themselves.

But computational complexity theory was built for discrete objects: strings, graphs, formulas. The systems that dominate natural science and engineering — differential equations, probability distributions, quantum states, neural network activations — are continuous, smooth, and infinite-dimensional. Complexity theory has something to say about these (through real computation models like the BSS model, or through discretization), but the fit is awkward. A fluid dynamicist

does not think in terms of P and NP. A financial engineer does not ask whether computing VaR is in PSPACE. The vocabulary does not match the objects.

This paper introduces a parallel complexity theory, built from the ground up for smooth systems. Where classical complexity theory asks "how many operations does it take to *solve* this problem?", Latent complexity theory asks "how many numbers does it take to *describe* this system?" The shift from solving to describing is fundamental:

- **Classical CT**: Problem → Algorithm → Resource count (time, space)
- **Latent CT**: System → Extraction → Representation size (N^*, grade, ρ)

The two are not in conflict. They answer different questions about different objects. But for smooth systems — which is to say, for most of science — Latent complexity theory has three decisive advantages:

1. **It is computable.** The central open question of classical CT — whether P $\neq$ NP — has been open since 1971. For Latent complexity, the analogous question — whether a specific system has $\rho > 1$ or $\rho = 1$ — is *decidable*. You can compute ρ from data.
2. **It gives rates, not just classes.** Classical CT tells you a problem is "in P" or "NP-hard." Latent CT tells you the system needs exactly $N^* = \lceil \log(1/\varepsilon)/\log \rho \rceil$ numbers for accuracy ε. The rate ρ is continuous, not binary.
3. **It captures interaction structure.** The grade of a Latent tells you *what kind* of complexity the system has — pairwise ($r = 2$), three-body ($r = 3$), or higher. Classical CT has no natural analogue for this (circuit depth is related but not identical).

1.2 What This Paper Does

We formalize Latent complexity as a mathematical framework with:

- **Definitions**: three complexity axes (ρ, grade, extraction cost) and four complexity classes (§2)
- **A strict hierarchy theorem**: L-ENTIRE $\subsetneq$ L-ANALYTIC $\subsetneq$ L-BOUNDARY $\subsetneq$ L-SINGULAR (§3)
- **A description–computation theorem**: representation size bounds query cost (§4)
- **A grade–depth correspondence**: grade $\leftrightarrow$ interaction circuit depth (§5)
- **A computability theorem**: Latent complexity is decidable (§6)
- **Formal relationships** to seven classical complexity measures (§7)
- **Applications** demonstrating the theory across four domains (§8)

1.3 What This Paper Is Not

This is not a claim that classical complexity theory should be replaced. It is a claim that for *smooth systems*, there exists a parallel theory that is better adapted to the objects of study. The two theories have different scopes:

	Classical CT	Latent CT
Objects	Finite strings, Boolean functions, decision problems	Smooth systems: distributions, dynamical systems, PDEs, models

	Classical CT	Latent CT
Resource	Time (# operations), space (# bits)	Description size (N^*), interaction order (grade), extraction cost
Classification	P, NP, PSPACE, … (discrete hierarchy)	L-ENTIRE, L-ANALYTIC(ρ), L-BOUNDARY, L-SINGULAR (continuous hierarchy)
Fundamental question	Is P $\neq$ NP? (open since 1971)	Is $\rho > 1$? (decidable for specific systems)
Practical output	"This problem is NP-hard" (qualitative)	"$\rho = 1.3$, $N^* = 55$, use spectral" (quantitative)

There is a bridge between the two — §7.5 discusses smooth analogues of classical classes — but the theories are complementary, not competitive.

1.4 The Universal Theorems: CLT, Cook–Levin, and the Latent Theorem

Every major mathematical theory has a **universal theorem** — a result that explains why a particular structure appears everywhere. The Central Limit Theorem explains why the Gaussian appears in every empirical distribution: sums of independent variables converge to it regardless of the components. The Cook–Levin Theorem explains why NP-hard problems appear across combinatorics, optimization, AI, and biology: they are all the same problem (SAT) in polynomial disguise.

The Latent Theorem (Nagy 2026a) is the universal theorem of Latent complexity theory. It explains why finite spectral representations work across finance, physics, ML, and engineering: smooth systems are finitely representable, and the representation size $N^* = \Theta(\log(1/\varepsilon)/\log\rho)$ is universal — independent of dimension, basis, and extraction method.

Theory	Universal theorem	Universality claim	Key parameter
Probability	Central Limit Theorem	Sums $\to$ Gaussian, regardless of components	σ^2 (variance)
Classical CT	Cook–Levin Theorem	Hard problems $\to$ SAT, regardless of origin	Polynomial degree (reduction)
Latent CT	**Latent Theorem**	Smooth systems $\to$ finite Latent, regardless of dimension/basis	ρ (Latent Number)

Each universal theorem comes with a **phase transition** governed by its key parameter:

Theory	Phase transition	Below	Above	Decidable?
Probability	$\sigma^2 = 0$ vs $\sigma^2 > 0$	Degenerate (no randomness)	CLT holds (Gaussian limit)	**Yes** — compute σ^2
Classical CT	P vs NP boundary	Polynomial-time (easy)	NP-hard (presumably exponential)	**No** — open since 1971
Latent CT	$\rho = 1$ vs $\rho > 1$	Incompressible (singular)	Exponentially compressible	**Yes** — compute ρ

The CLT and Latent phase transitions are decidable — you can compute σ^2 and ρ from data. The Cook–Levin phase transition (P vs NP boundary) is open — we cannot determine which side specific problems fall on.

This is not a superficial analogy. It reflects a structural fact: the CLT works because the Gaussian is the fixed point of convolution (a smooth operation). The Latent Theorem works because analytic continuation determines the representation (a smooth operation). The Cook–Levin Theorem works because polynomial reductions preserve computational difficulty — but reductions are discrete combinatorial objects, and the resulting structure is too poor to determine which class a specific problem belongs to (§6.3).

1.5 Position Within the Latent Program

This paper is the complexity-theoretic arm of a four-paper program:

Paper	Role	Central result
The Latent (Nagy 2026a)	Ontology	$\Lambda \in \mathfrak{L}(\mathcal{H})$: basis-free representation with $N^* = \Theta(\log(1/\varepsilon)/\log\rho)$
The -Diagnostic (Nagy 2026b)	measurement	3 algorithms for computing ; -algebra; Phase Transition Theorem; 6 domains
The Grade Method (Nagy 2026h)	Grade measurement	4-step protocol computing $A^{(k)}, \rho, k_{\text{eff}}$ for any ODE. 8 domains, 99 verified theorems
This paper	Complexity theory	L-ENTIRE $\subset$ L-ANALYTIC(ρ) $\subset$ L-BOUNDARY $\subset$ L-SINGULAR; decidability; 7 classical measures

The Latent paper defines the algebraic structure and proves the representation theorem. The -Diagnostic develops as a universal cross-domain measurement (from data, parametric models, or governing equations). The Grade Method develops grade structure for dynamical systems specifically. This paper consumes both: the measured by the Diagnostic and the grade measured by the Grade Method together determine a decidable complexity class.

The chain: -Diagnostic or Grade Method computes (ρ, grade) $\to$ Latent Complexity classifies (L-ANALYTIC, L-BOUNDARY, etc.) $\to$ Description–Computation Theorem bounds query cost $O(N^*)$ $\to$ the system is understood.

1.6 Relation to Prior Work

The Latent framework (Nagy 2026a), the ρ-diagnostic (Nagy 2026b), and the decomposability theory (Nagy 2026c) each contribute pieces of what we formalize here. The Latent paper defines the algebraic structure; the ρ-diagnostic identifies ρ as a universal complexity parameter; the decomposability paper introduces grade as a measure of system structure. The Grade Method (Nagy 2026h) provides the practical computational tool for measuring these invariants on specific systems. This paper's contribution is the synthesis: defining the formal complexity theory, proving the structural theorems (hierarchy, description–computation, grade–depth, computability), and establishing the precise relationships to classical measures.

In classical mathematics, the closest precedent is Kolmogorov's theory of n-widths (Kolmogorov 1936, Pinkus 1985), which gives optimal approximation rates for function classes. The n-width of the class of ρ-analytic functions is $\Theta(\rho^{-n})$ (Babenko 1958). What n-widths lack, and what Latent complexity adds, is: (i) computability of the rate for specific instances (not just worst-case over classes); (ii) grade structure (no analogue in n-width theory); (iii) the description–computation link (connecting representation to query efficiency).

In computational complexity theory, the BSS model (Blum, Shub, Smale 1989) extends Turing computation to real numbers. Our framework differs in that we do not model computation over the reals — we model *representation* of smooth systems. The BSS complexity of evaluating a polynomial is classical; the Latent complexity of the system described by that polynomial is the representation size.

---

2. The Three Axes of Latent Complexity

2.1 Axis I: The Latent Number ρ (Description Complexity)

Definition 1 (Latent Number). Let $\mathcal{S}$ be a system admitting a representation as an element $\Lambda(\mathcal{S}) \in \mathfrak{L}(\mathcal{H}) = \bigoplus_{r \geq 0} \mathcal{H}^{\otimes r}$ of a graded Hilbert tensor algebra (Nagy 2026a). The **Latent Number** of $\mathcal{S}$ is:

$$\rho(\mathcal{S}) = \left(\limsup_{k \to \infty} \|a_k\|^{1/k} \right)^{-1}$$

where $\{a_k\}$ are the spectral coefficients of $\Lambda(\mathcal{S})$ in any orthonormal basis. The Latent Number is basis-independent (Nagy 2026a, Theorem 2).

The Latent Number determines the **description size**: the minimum number of real numbers needed to represent $\mathcal{S}$ to accuracy ε:

$$N^*(\mathcal{S}, \varepsilon) = \left\lceil \frac{\log(C/\varepsilon)}{\log \rho(\mathcal{S})} \right\rceil = \Theta\left(\frac{\log(1/\varepsilon)}{\log \rho} \right)$$

ρ measures how *compressible* the system is. Larger ρ means fewer numbers needed. The phase transition at $\rho = 1$ is sharp: above it, exponential compression; at or below it, no finite description exists.

2.2 Axis II: Effective Grade r^* (Interaction Complexity)

Definition 2 (Grade Decomposition). The Latent $\Lambda(\mathcal{S})$ decomposes by grade:

$$\Lambda(\mathcal{S}) = \Lambda^{(0)} \oplus \Lambda^{(1)} \oplus \Lambda^{(2)} \oplus \Lambda^{(3)} \oplus \cdots$$

where $\Lambda^{(r)} \in \mathcal{H}^{\otimes r}$ captures the r-body interaction structure. Grade 0 is the scalar (mean behavior). Grade 1 captures individual component effects. Grade 2 captures pairwise interactions (correlations, generators). Grade r captures irreducible r-body couplings that cannot be decomposed into lower-order terms.

Definition 3 (Effective Grade). The **effective grade** of $\mathcal{S}$ at tolerance δ is:

$$r^*(\mathcal{S}, \delta) = \max \left\{ r \geq 0 : \frac{\|\Lambda^{(r)}\|}{\|\Lambda\|} > \delta \right\}$$

The effective grade measures the *interaction complexity* of the system — the highest order of irreducible coupling that is non-negligible. A pure grade-1 system has independent components (no interactions). Grade-2 means pairwise couplings suffice. Grade $r \geq 3$ means genuinely multi-body phenomena are present that cannot be reduced to pairwise analysis.

Examples:

System	Effective grade	Why
Independent coin flips	0 (or 1)	No coupling between coins
Projectile in vacuum	1	$x(t)$ and $y(t)$ are independent
Correlated portfolio	2	Pairwise correlations capture the structure
Three-body gravitational problem	3	Irreducible three-body coupling (chaos arises from grade 3)
Navier–Stokes turbulence	2 (velocity)	Quadratic nonlinearity; energy cascade is grade-2
Attention mechanism in transformers	≥ 3	Multi-head cross-attention creates higher-order coupling

2.3 Axis III: Extraction Cost $\mathcal{C}_E$ (Computational Complexity)

Definition 4 (Extraction Cost). The **extraction cost** $\mathcal{C}_E(\mathcal{S})$ is the computational cost (in operations) of computing $\Lambda_N(\mathcal{S})$ — the N-term Latent truncation — from the system's observable representation.

This is where classical computational complexity re-enters. Even if $\rho > 1$ and N^* is small, the cost of *finding* those N^* numbers may be large. The three costs are:

Method	Cost	When
Parametric (closed form)	$O(1)$	Model class known
Spectral (FFT/eigendecomposition)	$O(d \log d)$ to $O(d^3)$	Data or operator available
Data-driven (empirical CF)	$O(m \cdot N^*)$	Only samples available

The extraction cost is a *one-time* investment. Once Λ_N is computed, all queries cost $O(N^*)$ each (by the Description–Computation Theorem, §4). The amortized cost per query is $\mathcal{C}_E/Q + O(N^*)$, where Q is the number of queries. For $Q \gg 1$ (the typical case in science and engineering), the extraction cost is negligible.

2.4 The Complexity Fingerprint

Definition 5 (Latent Complexity Fingerprint). The **Latent complexity fingerprint** of a system $\mathcal{S}$ at accuracy ε and tolerance δ is the triple:

$$\mathfrak{C}(\mathcal{S}, \varepsilon, \delta) = (\rho(\mathcal{S}),\ r^*(\mathcal{S}, \delta),\ \mathcal{C}_E(\mathcal{S})) \in [1, \infty] \times \mathbb{N}_0 \times \mathbb{R}_+$$

This triple determines: - **How big** the representation is: N^* from ρ - **What kind** of structure it has: r^* grades of interaction - **How expensive** it is to find: $\mathcal{C}_E$ operations

Every smooth system has a Latent complexity fingerprint. The fingerprint is a quantitative, computable, multi-dimensional characterization of difficulty — far richer than a single complexity class.

---

3. The Latent Complexity Classes

3.1 Definitions

Definition 6 (Latent Complexity Classes). Let $\mathcal{S}$ be a system and $\rho(\mathcal{S})$, $\rho^*(\mathcal{S})$ its Latent Number and Extended Latent Number respectively (Nagy 2026b, Definitions 1 and 4).

- **L-ENTIRE**: $\rho(\mathcal{S}) = \infty$. The system is entire (holomorphic on all of $\mathbb{C}$). Super-exponential convergence. The representation is exact with finitely many terms or converges faster than any geometric rate.
- **L-ANALYTIC**(ρ) for fixed $\rho > 1$: $\rho(\mathcal{S}) \geq \rho$. The system is analytic with compressibility at least ρ. Representation size $N^* = O(\log(1/\varepsilon)/\log \rho)$. This is the main "tractable" class.
- **L-ANALYTIC** (without parameter): $\rho(\mathcal{S}) > 1$. The union $\bigcup_{\rho>1}$ **L-ANALYTIC**(ρ).
- **L-BOUNDARY**: $\rho(\mathcal{S}) = 1$ but $\rho^*(\mathcal{S}) > 1$. The system itself is not analytic, but it admits a smooth sufficient representation (characteristic function, generating function) that is analytic. Algebraic convergence in the natural domain; exponential convergence via transform.
- **L-SINGULAR**: $\rho^*(\mathcal{S}) = 1$. No smooth sufficient representation exists. Genuinely hard: no spectral shortcut, simulation required.

Remark 1 (Parametrized vs. unparametrized). L-ANALYTIC(ρ) is a parametrized family of classes, one for each $\rho > 1$. This is analogous to DTIME($f(n)$) in classical CT. The unparametrized L-ANALYTIC is the union, analogous to $\mathrm{P} = \bigcup_k \mathrm{DTIME}(n^k)$.

3.2 The Hierarchy

Theorem 1 (Strict Hierarchy). The complexity classes form a strict containment chain:

$$\textbf{L-ENTIRE} \subsetneq \textbf{L-ANALYTIC} \subsetneq \textbf{L-BOUNDARY} \subsetneq \textbf{L-SINGULAR} \cup \textbf{L-BOUNDARY}$$

More precisely, each containment is strict, witnessed by an explicit separating example. Within L-ANALYTIC, there is a continuous refinement indexed by ρ:

$$\rho_1 > \rho_2 > 1 \implies \textbf{L-ANALYTIC}(\rho_1) \subsetneq \textbf{L-ANALYTIC}(\rho_2)$$

Proof. We prove each strict containment by exhibiting a canonical separator and verifying its ρ and ρ^* values.

Part (a): L-ENTIRE $\subsetneq$ L-ANALYTIC. Containment: if $\rho = \infty$ then $\rho > 1$, so L-ENTIRE $\subseteq$ L-ANALYTIC. Strictness: consider $f(x) = 1/(x - z_0)$ on the interval $[-1, 1]$, where $z_0 \in \mathbb{C} \setminus [-1, 1]$. The Chebyshev expansion of f satisfies $|a_n| \sim C\rho^{-n}$ where $\rho = |z_0 + \sqrt{z_0^2 - 1}|$ is the parameter of the Bernstein ellipse passing through z_0 (Trefethen 2019). For $z_0 = 2$: $\rho = 2 + \sqrt{3} \approx 3.73$. Hence $f \in$ L-ANALYTIC(3.73) but $f \notin$ L-ENTIRE (it has a pole at $z_0 = 2$, so $\rho < \infty$).

A richer example: $f(x) = \log(1 + x)$ on $[-0.5, 1]$. The branch point at $x = -1$ limits the Bernstein ellipse, giving $\rho = 3 + 2\sqrt{2} \approx 5.83$. Clearly analytic on the interval but not entire.

Part (b): L-ANALYTIC $\subsetneq$ L-ANALYTIC $\cup$ L-BOUNDARY. Consider the Poisson distribution with parameter $\lambda > 0$. As a discrete measure, it has point masses at $\{0, 1, 2, ...\}$, so its probability density (in the distributional sense) has $\rho = 1$ — the atoms create singularities that prevent spectral decay faster than algebraic.

However, its characteristic function $\phi(t) = \exp(\lambda(e^{it} - 1))$ is entire: it extends analytically to all of $\mathbb{C}$ with $|\phi(z)| \leq \exp(\lambda(e^{|z|} - 1))$. Hence $\rho^* = \infty > 1$.

By definition, $\rho = 1$ but $\rho^* > 1$, so Poisson $\in$ L-BOUNDARY but $\notin$ L-ANALYTIC.

The geometric distribution provides another separator: $P(X = k) = (1 - p)^k p$ has $\rho = 1$ (discrete) but $\phi(t) = p/(1 - (1 - p)e^{it})$ has $\rho^* = 1/|\log(1 - p)| > 1$ for $p < 1 - 1/e$.

Part (c): L-BOUNDARY $\subsetneq$ L-BOUNDARY $\cup$ L-SINGULAR. L-BOUNDARY has $\rho^* > 1$; L-SINGULAR has $\rho^* = 1$. We must show that L-SINGULAR is nonempty.

The canonical L-SINGULAR example: the Cauchy distribution $f(x) = [\pi(1 + x^2)]^{-1}$. Its CF is $\phi(t) = e^{-|t|}$. The function $t \mapsto e^{-|t|}$ has a kink at $t = 0$ (non-differentiable), which prevents analytic extension to ANY strip $\{z : |\mathrm{Im}(z)| < \delta\}$ for any $\delta > 0$. Hence $\rho^* = 1$.

Why the kink? Because the Cauchy has fat tails: $\mathbb{E}[|X|] = \infty$. The non-existence of the first moment manifests as a non-smoothness of ϕ at the origin. This is the Paley–Wiener phenomenon: tail decay of f controls smoothness of $\hat{f}$.

A second L-SINGULAR example: the stable distribution with index $\alpha < 1$. Its CF $\phi(t) = \exp(-|t|^\alpha)$ is smooth away from $t = 0$ but has a non-analytic point at the origin (the $|t|^\alpha$ singularity). Hence $\rho^* = 1$.

Part (d): Continuous refinement within L-ANALYTIC. For $\rho_1 > \rho_2 > 1$: L-ANALYTIC(ρ_1) $\subseteq$ L-ANALYTIC(ρ_2) follows from the definition (if $\rho(\mathcal{S}) \geq \rho_1 > \rho_2$, then $\mathcal{S} \in$ L-ANALYTIC(ρ_2)). Strictness: $f(x) = 1/(x - z_0)$ with z_0 chosen so that $\rho = \rho_2$ gives a system in L-ANALYTIC(ρ_2) but not L-ANALYTIC(ρ_1), since $\rho_2 < \rho_1$.

Concretely: $f_1(x) = 1/(x - 1.01)$ on $[-1, 1]$ has $\rho \approx 1.01 + \sqrt{1.01^2 - 1} \approx 1.15$. The system $f_2(x) = e^x$ has $\rho = \infty$. Any system with ρ between these values (e.g., a polynomial of degree n perturbed by a pole) witnesses the strict containment at the corresponding ρ level.

Summary of separating examples:

Separation	Separator	ρ	ρ^*	Key property
L-ENTIRE $\subsetneq$ L-ANALYTIC	$1/(x-2)$ on $[-1,1]$	3.73	—	Pole prevents $\rho = \infty$
L-ANALYTIC $\subsetneq$ L-BOUNDARY	Poisson(λ)	1	∞	Discrete atoms, entire CF
L-BOUNDARY $\subsetneq$ L-SINGULAR	Cauchy distribution	1	1	Fat tails, non-smooth CF
L-ANALYTIC(ρ_1) $\subsetneq$ L-ANALYTIC(ρ_2)	$1/(x - z_0)$, $\lvert z_0\rvert$ tuned	ρ_2	—	Pole distance $\to \rho$

□

3.3 Grade Refinement

Each complexity class can be further refined by grade:

Definition 7 (Grade-Refined Classes). For $r \in \mathbb{N}_0$:

$$\textbf{L-ANALYTIC}(\rho, r) = \{\mathcal{S} \in \textbf{L-ANALYTIC}(\rho) : r^*(\mathcal{S}) = r\}$$

The grade refinement captures the qualitative difference between systems of the same ρ:

ρ	Grade 1	Grade 2	Grade 3
1.3	100-asset portfolio (independent risk factors)	100-asset portfolio (correlated)	Credit portfolio with contagion
2.0	Single GBM stock	Heston model (stochastic volatility)	Multi-factor SV with jumps
∞	Gaussian random variable	Gaussian process (correlation structure)	Gaussian graphical model (conditional independence)

Grade and ρ are independent axes: a system can have high ρ (very compressible) but high grade (complex interactions), or low ρ (hard to compress) but low grade (simple structure). The three-body problem is the canonical example: grade 3 (irreducible three-body coupling) but ρ varies from large (periodic orbits) to near 1 (chaotic orbits).

---

4. The Description–Computation Theorem

4.1 The Core Result

The central theorem of Latent complexity theory connects description complexity to computational complexity: if you can describe a system with N^* numbers, you can answer any query about it in $O(N^*)$ operations. This is the theorem that makes Latent complexity a *theory of computation*, not merely a theory of approximation.

Theorem 2 (Description–Computation). Let $\mathcal{S} \in \textbf{L-ANALYTIC}(\rho)$ with $\rho > 1$ and spectral bound $\|a_k\| \leq C\rho^{-k}$. Let $\{e_k\}_{k\geq 0}$ be an orthonormal basis of $\mathcal{H}$ and $\Lambda_N = \sum_{k=0}^{N} a_k e_k$ the N-term truncation. Then:

(a) Linear query cost. For any bounded linear projector $\Pi : \mathcal{H} \to \mathbb{R}$ with $\|\Pi\| \leq B$, the query $\Pi(\Lambda_N)$ is computable in $O(N)$ arithmetic operations.

(b) Accuracy guarantee. Setting $N = N^*(\varepsilon) = \lceil \log(BC/(\varepsilon(\rho-1)))/\log\rho \rceil$:

$$|\Pi(\Lambda) - \Pi(\Lambda_{N^*})| \leq \varepsilon$$

(c) Uniform truncation. The truncation Λ_{N^*} satisfies the ε-accuracy guarantee for ALL bounded linear projectors with $\|\Pi\| \leq B$ simultaneously, using a single truncation at $N^*(B, \varepsilon)$.

(d) Amortized complexity. After a one-time extraction costing $\mathcal{C}_E$, answering Q queries costs $\mathcal{C}_E + Q \cdot O(N^*)$ total.

(e) Nonlinear extension. For Lipschitz-continuous nonlinear functionals $F : \mathcal{H} \to \mathbb{R}$ with Lipschitz constant L_F:

$$|F(\Lambda) - F(\Lambda_{N^*})| \leq L_F \cdot \frac{C}{\rho - 1} \cdot \rho^{-N^*}$$

The query cost increases to $O(N^* \cdot c_F)$ where c_F is the per-component cost of evaluating F.

Proof.

Part (a). The projector Π acts on $\mathcal{H}$ by $\Pi(v) = \langle w, v\rangle$ for some $w \in \mathcal{H}$ with $\|w\| = \|\Pi\|$ (Riesz representation). Let $w_k = \langle w, e_k\rangle$ be the projector weights in the basis $\{e_k\}$. Then:

$$\Pi(\Lambda_N) = \left\langle w, \sum_{k=0}^{N} a_k e_k \right\rangle = \sum_{k=0}^{N} a_k w_k$$

This requires N multiplications ($a_k w_k$) and $N-1$ additions ($\sum$). The weights $\{w_k\}$ are precomputed once (at cost $O(N)$ per projector) and reused for each system. Total per-query cost: $2N-1 = O(N)$ arithmetic operations. $\square$

Part (b). The truncation error in $\mathcal{H}$-norm:

$$\|\Lambda - \Lambda_N\|_{\mathcal{H}} = \left\| \sum_{k>N} a_k e_k \right\|_{\mathcal{H}} = \sqrt{\sum_{k>N} |a_k|^2}$$

We bound this using the spectral decay:

$$\sum_{k>N} |a_k|^2 \leq C^2 \sum_{k>N} \rho^{-2k} = \frac{C^2 \rho^{-2N}}{\rho^2 - 1} \cdot \rho^2 = \frac{C^2 \rho^{-2(N-1)}}{\rho^2 - 1}$$

For the ℓ^1-based bound (tighter for projector estimates):

$$|\Pi(\Lambda) - \Pi(\Lambda_N)| = \left| \sum_{k>N} a_k w_k \right| \leq \sum_{k>N} |a_k| \cdot |w_k| \leq \|\Pi\| \sum_{k>N} |a_k|$$

using Cauchy–Schwarz and $\sum |w_k|^2 = \|w\|^2 = \|\Pi\|^2$. The tail sum:

$$\sum_{k>N} |a_k| \leq C \sum_{k>N} \rho^{-k} = \frac{C \rho^{-N}}{\rho - 1}$$

Combining:

$$|\Pi(\Lambda) - \Pi(\Lambda_N)| \leq \frac{BC \rho^{-N}}{\rho - 1}$$

Setting this $\leq \varepsilon$:

$$\rho^{-N} \leq \frac{\varepsilon(\rho - 1)}{BC} \quad \Longleftrightarrow \quad N \geq \frac{\log(BC/(\varepsilon(\rho-1)))}{\log \rho}$$

giving $N^*(\varepsilon) = \lceil \log(BC/(\varepsilon(\rho-1)))/\log \rho \rceil = \Theta(\log(1/\varepsilon)/\log \rho)$. $\square$

Part (c). Fix $B > 0$ and $\varepsilon > 0$. The formula $N^*(B, \varepsilon)$ depends on $\|\Pi\|$ only through B. For any projector Π with $\|\Pi\| \leq B$, the truncation at $N^*(B, \varepsilon)$ gives error $\leq \varepsilon$, since the bound in (b) is monotone in $\|\Pi\|$. The truncation Λ_{N^*} is computed once; each projector reads the same N^* coefficients. $\square$

Part (d). Extraction produces $\{a_0, \ldots, a_{N^*}\}$ at cost $\mathcal{C}_E$. Query q evaluates $\Pi_q(\Lambda_{N^*}) = \sum_{k=0}^{N^*} a_k w_k^{(q)}$ at cost $O(N^*)$. Total for Q queries:

$$\mathcal{C}_E + Q \cdot O(N^*) = \mathcal{C}_E + O(Q \cdot \log(1/\varepsilon)/\log \rho)$$

The per-query cost is dominated by $O(N^*)$ for $Q \gg \mathcal{C}_E/N^*$. □

Part (e). For a Lipschitz-continuous functional F with constant L_F:

$$|F(\Lambda) - F(\Lambda_N)| \leq L_F \cdot \|\Lambda - \Lambda_N\|_{\mathcal{H}} \leq L_F \cdot \frac{C\rho^{-(N-1)}}{\sqrt{\rho^2 - 1}}$$

The accuracy guarantee holds with $N^* = O(\log(L_F/\varepsilon)/\log\rho)$ — the same scaling as the linear case, with L_F replacing B. The query cost is $O(N^* \cdot c_F)$, where c_F is the cost of one evaluation of F's kernel (e.g., for $F(\Lambda) = \max_k |a_k|$, we have $c_F = 1$; for $F(\Lambda) = \|\Lambda\|^2 = \sum |a_k|^2$, we have $c_F = 1$). □

Remark 1 (What Part (e) does and does not cover). The nonlinear extension covers VaR (quantile functional: Lipschitz on smooth CDFs), option payoffs (call/put: Lipschitz with $L_F = 1$), norms and inner products, and all C^1 functionals with bounded gradients. It does NOT cover optimization ("find arg max"), game-theoretic quantities, or functionals that are not Lipschitz (e.g., indicator functions of sets — but see the Extended Latent Theorem for handling these via smooth representations).

4.2 The Description–Computation Gap

The Description–Computation Theorem says: **for smooth systems, the representation IS the computation.** Once you have the Latent, there is nothing left to compute — only to project.

This identifies a structural difference between discrete and smooth complexity:

In discrete CT: description ≪ computation. The integer $n = p \cdot q$ is describable in $O(\log n)$ bits, but finding p, q from n (presumably) requires super-polynomial time. The description hides the computational difficulty. A SAT formula has $O(n)$ clauses but (presumably) requires $2^{O(n)}$ time to solve. Compact description does not imply efficient computation — the gap between description size and computation cost can be exponential.

In Latent CT: description = computation. A system $\mathcal{S} \in$ L-ANALYTIC(ρ) is describable in N^* numbers, and every query costs $O(N^*)$. There is no gap. The Latent is not just a compact description — it is a *sufficient* description: every property of the system is a projection of the Latent, and projecting costs $O(N^*)$.

Why is there no gap? Because of **sufficiency**. The Latent Λ encodes ALL distributional, dynamic, and functional properties of the system (Nagy 2026a, Theorem 2). In classical CT, a compact description of a problem (e.g., the digits of n) does not encode the answer (the factors p, q) — the description encodes the *question*, not the *answer*. The Latent encodes the answer to every question simultaneously. This is the sense in which the Latent is more than a representation: it is a universal pre-computation.

	Classical description	Latent
What it encodes	The question (input)	The answer to all questions (sufficient statistic)
Query cost after description	May be exponential ($2^{O(n)}$)	$O(N^*)$
Description–computation gap	Exponential (if P ≠ NP)	Zero
Why?	Description hides structure	Latent IS structure

4.3 The Cost Decomposition

The total cost of answering a query about a smooth system decomposes as:

$$\text{Total cost} = \underbrace{\mathcal{C}_E}_{\text{extraction (once)}} + \underbrace{Q \cdot O(N^*)}_{\text{queries (per query)}}$$

Compare to Monte Carlo:

$$\text{MC cost} = Q \cdot O(m)$$

where $m = O(1/\varepsilon^2)$ is the number of samples per query. The crossover point is:

$$Q^* = \frac{\mathcal{C}_E}{m - N^*} \approx \frac{\mathcal{C}_E}{\varepsilon^{-2}}$$

For even modest extraction costs ($\mathcal{C}_E \sim d^3$ for eigendecomposition of a $d \times d$ matrix), the crossover happens at $Q^* \sim 1$ for $\varepsilon = 10^{-6}$. After one query, the Latent approach dominates.

4.4 Concrete Cost Comparison

To anchor the theorem in practice, we tabulate the costs for a specific system: a $d = 100$-asset correlated portfolio with $\rho = 1.3$ and $\varepsilon = 10^{-6}$.

Quantity	Latent approach	Monte Carlo	Ratio
Extraction cost $\mathcal{C}_E$	$O(d^3) = O(10^6)$	—	one-time
Representation size N^*	$\lceil \log(10^6)/\log(1.3) \rceil = 53$	—	—
Query cost (VaR)	53 operations	10^{12} samples	$2 \times 10^{10}\times$
Query cost (ES)	53 operations	10^{12} samples	$2 \times 10^{10}\times$
Query cost (option price)	53 operations	10^{12} samples	$2 \times 10^{10}\times$
1000 queries total	$10^6 + 53{,}000 \approx 10^6$	10^{15}	$10^9\times$

The extraction pays for itself after a single query. Every additional query is essentially free compared to Monte Carlo.

---

5. The Grade–Depth Correspondence

5.1 Interaction Circuits

To connect grade to classical circuit complexity, we define a natural computational model for smooth systems.

Definition 8 (Interaction Circuit). An **interaction circuit** over a separable Hilbert space $\mathcal{H}$ is a directed acyclic graph $G = (V, E)$ where:

- **Input nodes** $v_1, \ldots, v_d$ carry elements of $\mathcal{H}$ (individual components: $v_i \mapsto e_i \in \mathcal{H}$).
- **Gates** are internal nodes. Each gate g has a fixed arity $r(g) \geq 1$ and computes an $r(g)$-linear map $\mu_g : \mathcal{H}^{\otimes r(g)} \to \mathcal{H}^{\otimes s(g)}$ for some $0 \leq s(g) \leq r(g)$. The map μ_g is a contraction: a composition of tensor product and trace (pairing of indices).
- **Output node** v_{out} carries the result in $\bigoplus_r \mathcal{H}^{\otimes r}$.
- **Depth** $D(G)$ is the length of the longest directed path from any input to the output.
- **Width** $W(G)$ is the maximum number of wires at any depth level.

A gate of arity 1 computes a linear map $\mathcal{H} \to \mathcal{H}$ (rotation, scaling, projection). A gate of arity 2 computes a bilinear map $\mathcal{H} \otimes \mathcal{H} \to \mathcal{H}^{\otimes s}$: inner products ($s = 0$), matrix-vector products ($s = 1$), or outer products ($s = 2$). A gate of arity r computes an r-body interaction.

Definition 9 (Interaction Depth and Width). The **interaction depth** $D(\mathcal{S})$ and **interaction width** $W(\mathcal{S})$ of a system $\mathcal{S}$ are:

$$D(\mathcal{S}) = \min_G D(G), \qquad W(\mathcal{S}) = \min_{G : D(G) = D(\mathcal{S})} W(G)$$

where the minimum is over all interaction circuits G that compute $\Lambda(\mathcal{S})$ from inputs $\{e_1, \ldots, e_d\}$.

5.2 The Correspondence

Theorem 3 (Grade–Depth Correspondence). For any system $\mathcal{S}$ with Latent $\Lambda = \bigoplus_{r=0}^{R} \Lambda^{(r)}$ and effective grade $r^*(\mathcal{S}, \delta)$:

(a) Grade bounds depth (upper bound). The interaction depth satisfies:

$$D(\mathcal{S}) \leq r^*(\mathcal{S}, \delta)$$

(b) Depth bounds grade (lower bound). If $\mathcal{S}$ is computed by an interaction circuit of depth D using gates of maximum arity $r_{\max}$, then:

$$r^*(\mathcal{S}, \delta) \leq D \cdot r_{\max}$$

(c) Tight characterization for bilinear circuits. With only bilinear gates ($r_{\max} = 2$):

$$\lceil \log_2 r^* \rceil \leq D(\mathcal{S}) \leq r^*$$

The lower bound $\lceil \log_2 r^* \rceil$ is achieved by balanced tree circuits. The upper bound r^* is achieved by sequential chaining. Both are tight.

(d) Grade–NC correspondence. Let $\mathbf{IC}(D)$ denote the class of systems computable by interaction circuits of depth D with bilinear gates. Then:

$$\mathbf{IC}(0) \subsetneq \mathbf{IC}(1) \subsetneq \mathbf{IC}(2) \subsetneq \cdots$$

and a system of effective grade r^* satisfies $\mathcal{S} \in \mathbf{IC}(r^*)$ but $\mathcal{S} \notin \mathbf{IC}(\lceil \log_2 r^* \rceil - 1)$ for homogeneous systems (Definition 3 with $\delta \to 0$).

Proof.

Part (a). We construct an explicit circuit. The grade-r component $\Lambda^{(r)} \in \mathcal{H}^{\otimes r}$ admits a decomposition (by the tensor SVD / higher-order SVD):

$$\Lambda^{(r)} = \sum_{j=1}^{J_r} \sigma_j^{(r)}\, u_1^{(j)} \otimes u_2^{(j)} \otimes \cdots \otimes u_r^{(j)}$$

where $\sigma_j^{(r)} \geq 0$ and $u_i^{(j)} \in \mathcal{H}$.

Each rank-1 tensor $u_1 \otimes \cdots \otimes u_r$ is computable by a tree of bilinear gates at depth $\lceil \log_2 r \rceil$: - Pair the inputs: $u_1 \otimes u_2$, $u_3 \otimes u_4$, … at depth 1 - Pair the pairs: $(u_1 \otimes u_2) \otimes (u_3 \otimes u_4)$, … at depth 2 - Continue until all r factors are combined at depth $\lceil \log_2 r \rceil$

The sum $\sum_j \sigma_j^{(r)}(\cdots)$ requires one additional addition layer (linear gate, depth +1 if not parallelized, or depth +0 if additions are absorbed into the gate outputs).

The full Latent $\Lambda = \bigoplus_{r=0}^{r^*} \Lambda^{(r)}$ has each grade computed in parallel. The depth is determined by the highest grade: $D \leq r^*$ (sequential chaining) or $D \leq \lceil \log_2 r^* \rceil$ (balanced tree per grade). Using an r^*-ary gate directly: $D = 1$. Using only bilinear gates: $D \leq r^*$. □

Part (b). We show by induction on depth that a circuit of depth D with gates of arity $\leq r_{\max}$ can produce tensors of rank at most $D \cdot r_{\max}$.

Base case ($D = 0$): input wires carry elements of $\mathcal{H} = \mathcal{H}^{\otimes 1}$. Tensor rank 1.

Inductive step: at depth d, each wire carries a tensor of rank $\leq d \cdot r_{\max}$. A gate of arity r at depth $d+1$ takes r inputs of rank $\leq d \cdot r_{\max}$ each and produces their tensor product (rank $\leq r \cdot d \cdot r_{\max}$) followed by contraction (rank can only decrease). Hence the output has rank $\leq r \cdot d \cdot r_{\max} \leq r_{\max} \cdot d \cdot r_{\max}$.

This gives rank $\leq r_{\max}^D$ in general. For the tighter bound: a single gate of arity r at depth $d+1$ increases the tensor rank by at most r, so rank $\leq r_{\max} \cdot D$ by additive accumulation. □

Part (c). The upper bound $D \leq r^*$ follows from Part (a) using sequential bilinear chaining: $e_1 \otimes e_2$ at depth 1, $(e_1 \otimes e_2) \otimes e_3$ at depth 2, …, reaching $\mathcal{H}^{\otimes r^*}$ at depth $r^* - 1$.

The lower bound $D \geq \lceil \log_2 r^* \rceil$: a bilinear gate doubles the maximum tensor order at each depth level (by taking the tensor product of two inputs). Starting from order 1, after D levels the maximum achievable order is 2^D. Hence $r^* \leq 2^D$, giving $D \geq \lceil \log_2 r^* \rceil$.

The lower bound is achieved by balanced binary trees. The upper bound is achieved by sequential chaining (right-associated tensor products). Both constructions are explicit. □

Part (d). $\mathbf{IC}(0)$ contains only grade-1 systems (the inputs themselves). $\mathbf{IC}(1)$ adds bilinear combinations (grade ≤ 2). $\mathbf{IC}(D)$ adds grade $\leq 2^D$. The containments are strict because, for each D, the rank-1 tensor $e_1 \otimes \cdots \otimes e_{2^D+1} \in \mathcal{H}^{\otimes(2^D+1)}$ requires depth $D+1$ (by the lower bound in Part c) and thus $\notin \mathbf{IC}(D)$. □

5.3 Interpretation: Grade as Interaction Depth

The Grade–Depth Correspondence says: **grade measures how many layers of irreducible interaction a system contains.** This is the smooth-system analogue of circuit depth — the central structural measure in classical circuit complexity.

The correspondence maps naturally to the classical NC hierarchy:

Grade	IC class	NC analogue	Interaction structure	Physical examples
0	IC(0)	NC^0	Scalar (mean, no computation)	Constants, averages
1	IC(0)	NC^0	Independent components, no gates	Uncorrelated assets, projectile
2	IC(1)	NC^1	One layer of bilinear gates	Correlation, Fokker–Planck, Hessian
3	IC(2)	NC^2	Two bilinear layers (or one trilinear)	Three-body problem, attention
4	IC(2)	NC^2	Two bilinear layers suffice	Four-body interactions (rare)
r	IC($\lceil \log_2 r \rceil$)	$\mathrm{NC}^{\lceil \log_2 r \rceil}$	$\lceil \log_2 r \rceil$ layers of pairwise interaction	r-body coupling

The logarithmic compression — grade r requires only $\lceil \log_2 r \rceil$ bilinear layers — is a consequence of the tree structure of tensor products. This is the smooth analogue of the fact that parity (an n-ary operation) can be computed by a depth-$\lceil \log_2 n \rceil$ circuit of XOR gates.

5.4 Worked Example: The Three-Body Problem

The gravitational three-body problem provides the cleanest demonstration. Consider three masses m_1, m_2, m_3 with positions $\mathbf{q}_i \in \mathbb{R}^3$.

Grade 0 ($\Lambda^{(0)} \in \mathbb{R}$): center-of-mass energy. A scalar. No interaction needed.

Grade 1 ($\Lambda^{(1)} \in \mathcal{H}$): individual kinetic energies $\frac{1}{2} m_i |\dot{\mathbf{q}}_i|^2$. Each body in isolation. No gates.

Grade 2 ($\Lambda^{(2)} \in \mathcal{H}^{\otimes 2}$): pairwise gravitational potentials $-G m_i m_j / |\mathbf{q}_i - \mathbf{q}_j|$. Three pairs: $(1,2), (1,3), (2,3)$. One bilinear gate per pair. **This is where the two-body problem lives** — and the two-body problem is integrable.

Grade 3 ($\Lambda^{(3)} \in \mathcal{H}^{\otimes 3}$): the irreducible three-body coupling. This is the part that CANNOT be decomposed into pairwise interactions. It captures: - The tidal effect of body 3 on the pair (1,2) - Three-body resonances - The onset of chaos

Numerically (Nagy 2026g): $\|\Lambda^{(3)}\| / \|\Lambda^{(2)}\|$ ranges from ~ 0.01 for near-circular hierarchical orbits (grade-3 is small, the system is nearly integrable) to ~ 10 for chaotic orbits (grade-3 dominates, the system is essentially a three-body phenomenon).

Interaction circuit: With bilinear gates, computing $\Lambda^{(3)}$ requires depth 2: - Depth 1: compute $e_1 \otimes e_2$ and $e_2 \otimes e_3$ (pairwise) - Depth 2: compute $(e_1 \otimes e_2) \otimes e_3$ (three-body)

This is tight: $\lceil \log_2 3 \rceil = 2$.

5.5 Why High Grade Is Rare in Nature

An empirical observation with a structural explanation: most natural systems have low effective grade ($r^* \leq 3$). The Grade Equation (Nagy 2026d) proves that for any analytic dynamical system, the grade components decay exponentially:

$$\|\Lambda^{(r)}\| \leq C_0 \rho_{\text{grade}}^{-r}$$

with $\rho_{\text{grade}} > 1$ for any system with a spectral gap. High grades are exponentially suppressed. This is why pairwise models (grade 2) work so well across science — not because nature is simple, but because the grade hierarchy guarantees that higher-order interactions are exponentially small relative to lower-order ones.

The exceptions — systems with grade 3 or higher that is non-negligible — are exactly the systems that exhibit qualitatively new phenomena (chaos, phase transitions, emergent complexity). The grade identifies where these phenomena live.

---

6. The Computability Theorem

6.1 The Central Claim

Theorem 4 (Computability of Latent Complexity). The Latent complexity class of a smooth system $\mathcal{S}$ is decidable. Specifically:

(a) ρ **is computable.** Given finite observations $\{x_1, \ldots, x_m\}$ from $\mathcal{S}$, the estimator $\hat{\rho}$ converges: $\hat{\rho} \xrightarrow{P} \rho(\mathcal{S})$ as $m \to \infty$. For parametric systems, ρ has a closed form.

(b) **Grade is measurable.** The effective grade r^* can be determined from the cumulant hierarchy: the r-th cumulant tensor is nonzero if and only if $\Lambda^{(r)} \neq 0$.

(c) **The complexity class is decidable.** Given $\hat{\rho}$ and a confidence level α:

- If $\hat{\rho} > 1$ with confidence $> 1 - \alpha$: $\mathcal{S} \in$ L-ANALYTIC.
- If $\hat{\rho}$ is consistent with 1 but $\hat{\rho}^* > 1$: $\mathcal{S} \in$ L-BOUNDARY.
- If $\hat{\rho}^* = 1$: $\mathcal{S} \in$ L-SINGULAR.

Proof. Part (a) is proved in Nagy (2026b, Theorem 4): the empirical spectral decay estimator is consistent.

Part (b): The r-th cumulant tensor $\kappa^{(r)}$ of $\mathcal{S}$ satisfies $\kappa^{(r)} = 0$ iff $\Lambda^{(r)} = 0$ for centered systems. The sample cumulant $\hat{\kappa}^{(r)}$ converges to $\kappa^{(r)}$ by the law of large numbers for U-statistics (Hoeffding 1948). Testing $\kappa^{(r)} = 0$ reduces to a multivariate hypothesis test with known asymptotics.

Part (c) follows from (a) and the definition of the classes. The test $\hat{\rho} > 1$ is a one-sided test with standard confidence interval theory. □

6.2 The P vs NP Problem and Why Smooth Systems Avoid It

The deepest open problem in computational complexity theory — and one of the seven Millennium Prize Problems — is the P vs NP question (Cook 1971, Karp 1972). Understanding precisely

what it asks, why it remains open, and why the Latent framework's analogous question is *resolved* illuminates the structural difference between discrete and smooth complexity.

6.2.1 What P vs NP Asks

P is the class of decision problems solvable in polynomial time by a deterministic Turing machine. **NP** is the class of decision problems whose *solutions* are verifiable in polynomial time — given a proposed answer, you can check it quickly, even if finding it may be hard.

Clearly P $\subseteq$ NP: if you can solve a problem quickly, you can certainly check a solution quickly. The question is the converse:

> **The P vs NP Question.** Is P = NP? That is: if a solution can be *checked* quickly, can it always be *found* quickly?

The question separates *finding* from *checking*, *creativity* from *verification*. Its concrete meaning:

Problem	Finding a solution	Checking a solution
Sudoku ($n \times n$)	Hard (exponential search)	Easy (verify all constraints)
Integer factoring	Hard (no known poly-time algorithm)	Easy (multiply the factors)
Traveling salesman	Hard (optimal tour among $n!$ permutations)	Easy (is this tour shorter than k?)
SAT (Boolean satisfiability)	Hard (NP-complete)	Easy (evaluate the formula)

Nearly every researcher believes P $\neq$ NP — that finding is genuinely harder than checking. But after 55 years, no proof exists.

6.2.2 The Three Barriers

The P vs NP question is not merely unsolved — it is *provably resistant* to every standard proof technique in computational complexity. Three formal barrier results explain why.

Barrier 1: Relativization (Baker–Gill–Solovay 1975). There exist oracles A and B such that $\mathrm{P}^A = \mathrm{NP}^A$ and $\mathrm{P}^B \neq \mathrm{NP}^B$. Any proof technique that *relativizes* — that is, works unchanged when all machines are given access to an arbitrary oracle — cannot resolve P vs NP, because such a technique would have to give the same answer for both oracles. This eliminates diagonalization arguments (the technique that proves the Time Hierarchy Theorem, the Halting Problem, and most of classical computability theory).

Barrier 2: Natural Proofs (Razborov–Rudich 1997). A "natural" proof of a circuit lower bound is one based on a combinatorial property that is (i) possessed by a random function with high probability ("largeness") and (ii) efficiently testable ("constructivity"). Razborov and Rudich proved that if one-way functions exist (a standard cryptographic assumption), then no natural proof can establish super-polynomial circuit lower bounds. Since proving P $\neq$ NP requires exactly such a lower bound, the most intuitive proof strategy — finding a combinatorial property that hard functions have and easy functions lack — is blocked by the same assumption that underpins all of modern cryptography.

Barrier 3: Algebrization (Aaronson–Wigderson 2009). Even algebraic techniques — the tools that proved IP = PSPACE (Shamir 1992), MIP = NEXP, and other major results — cannot resolve P vs NP. The algebrization barrier extends relativization to show that proof techniques using arithmetization and low-degree polynomial extensions are insufficient. Every known complexity separation (that is not just diagonalization) uses algebraic techniques that algebrize — and therefore cannot reach P vs NP.

Together, these barriers imply that resolving P vs NP requires a *fundamentally new mathematical technique* — one that is simultaneously non-relativizing, non-natural, and non-algebrizing. No known mathematical framework provides such a technique. This is why the problem has resisted 55 years of sustained effort by the world's best mathematicians and computer scientists.

6.2.3 The Foundational Theorems of Classical CT

To see where the Latent framework diverges, it helps to identify the structural theorems that classical CT *does* have:

Result	Year	What it says	Role
Cook–Levin Theorem	1971	SAT is NP-complete: every NP problem reduces to SAT	Universality (the "CLT of CT") — explains why hard problems appear everywhere
Time Hierarchy	1965	More time = strictly more power: $\mathrm{DTIME}(n^k) \subsetneq \mathrm{DTIME}(n^{k+1})$	Strict separation within deterministic classes
PCP Theorem	1998	$\mathrm{NP} = \mathrm{PCP}(O(\log n), O(1))$: every proof is locally checkable	Universality of proof structure
Dichotomy Theorem	2017	Every CSP is in P or NP-complete — nothing in between	Sharp phase transition (for a restricted class)

The Cook–Levin Theorem plays the role of a universality result: it explains why NP-hard problems appear across combinatorics, optimization, AI, biology, and economics — they are all *the same problem* (SAT) in disguise, connected by polynomial reductions. This is analogous to how the Central Limit Theorem explains why the Gaussian appears across probability — sums converge to it regardless of the components.

But the Cook–Levin Theorem does not tell you *which side of the boundary a specific problem falls on.* It says "if P $\neq$ NP, then SAT is hard" — but P $\neq$ NP remains unproved.

6.2.4 Why Smooth Systems Avoid the Barriers

The Latent framework's phase transition at $\rho = 1$ is the smooth analogue of P vs NP — but it is decidable. The reason lies in the structural properties of smooth systems that are absent in the discrete setting.

Why relativization is irrelevant. The relativization barrier applies to Turing machines with oracle access — a model of computation where the internal structure of the oracle is hidden behind a "black box." Smooth systems have no oracles. The Latent Number ρ is extractable from the system's *internal structure* (the analyticity of its representation). There is no black box to hide behind: the holomorphic extension either exists (to a Bernstein ρ-ellipse) or it does not, and this is testable from the spectral decay of the coefficients.

Why natural proofs are irrelevant. The natural proofs barrier applies to combinatorial properties of Boolean functions. The Latent framework does not use combinatorial properties — it uses *analytic continuation*. The distinction between $\rho > 1$ and $\rho = 1$ is not a combinatorial property that a random function might or might not have; it is a complex-analytic property (existence of holomorphic extension) that is determined by the singularity structure of the system. There is no tension with pseudorandom generators because smooth systems are not Boolean functions.

Why algebrization is irrelevant. Algebrization blocks proofs that extend computations to low-degree polynomials over finite fields. The Latent framework operates over $\mathbb{R}$ and $\mathbb{C}$ with infinite-precision real analysis. The tools — Bernstein ellipses, Paley–Wiener theory, spectral gap estimates — are from classical analysis, not algebraic complexity. They do not algebrize because they were never algebraic.

The deeper reason: smooth systems carry *analytic structure* — holomorphic extensions, spectral gaps, exponential decay rates — that discrete computational objects lack. This structure makes the complexity question answerable. The P vs NP question is hard precisely because Boolean functions have no such structure: a random Boolean function on n bits requires a circuit of size $\Omega(2^n/n)$, but distinguishing "random-looking hard functions" from "structured easy functions" is exactly what the natural proofs barrier prevents.

In short: the three barriers are artifacts of the *discrete domain*, not of complexity theory per se. Moving to the smooth domain removes the barriers — not by cleverness, but by having access to structure (analyticity) that the discrete world lacks.

6.2.5 The Complete Contrast

	Classical CT	Latent CT
Domain	Discrete: strings, circuits, Turing machines	Smooth: analytic functions, distributions, PDEs
Phase transition	P vs NP boundary	$\rho = 1$ boundary
Is the transition strict?	Unknown (P $\neq$ NP is open)	**Yes** — Theorem 1 (strict hierarchy)
Can you classify a specific instance?	No — we cannot prove factoring $\notin$ P	**Yes** — ρ is computable (Theorem 4)
Universality result	Cook–Levin: every NP problem $\to$ SAT	Latent Theorem: every smooth system $\to$ finite Latent
Proof barriers	3 formal barriers block all known techniques	None — analytic structure makes the question tractable
What structure is available?	Almost none (random Boolean functions have no exploitable structure)	Full analytic structure: holomorphic extensions, spectral gaps, exponential decay

	Classical CT	Latent CT
Fundamental difficulty	Finding vs. checking (creativity gap)	Description size vs. query cost (compression gap)

The Latent framework does not "solve" P vs NP. It operates in a parallel domain where the analogous question is answerable because the domain carries enough structure. The philosophical lesson is precise: **the difficulty of P vs NP is not about complexity — it is about the poverty of structure in the discrete world.**

6.3 Self-Hiding vs Self-Revealing Complexity

The contrast in §6.2.5 admits a sharper formulation. The fundamental difference between discrete and smooth complexity is not merely technical — it is about whether *difficulty reveals itself or conceals itself.*

6.3.1 The Dichotomy

Self-revealing complexity. A domain has *self-revealing complexity* if there exists a computable difficulty parameter — an invariant extractable from the system — that determines the system's complexity class. In such a domain, the difficulty of a system is visible from its structure. You can measure how hard a system is before attempting to solve it.

Smooth systems are self-revealing. The Latent Number ρ is: - **Intrinsic**: a property of the system, not of any representation or algorithm - **Computable**: extractable from data (empirical spectral decay), models (closed form), or equations (spectral gap) - **Determining**: $\rho > 1$ vs $\rho = 1$ fully determines the complexity class - **Rate-giving**: not just a binary classification but a continuous parameter that specifies the exact representation size $N^* = \Theta(\log(1/\varepsilon)/\log\rho)$

Smooth systems cannot hide their complexity. An analytic function's singularity structure is globally determined and locally detectable — the coefficients' decay rate betrays the analyticity strip width, which betrays ρ, which determines everything. There is nowhere for difficulty to hide.

Self-hiding complexity. A domain has *self-hiding complexity* if no computable difficulty parameter exists — if the difficulty of a system is not extractable from its structure without essentially solving the system. In such a domain, hard instances are indistinguishable from easy ones by any efficient test.

Boolean functions are self-hiding. This is not a conjecture — it is a theorem, conditional on a standard cryptographic assumption:

Proposition 3 (Self-Hiding Property of Boolean Functions). If one-way functions exist, then there is no polynomial-time computable property $P : \{0,1\}^{2^n} \to \{0,1\}$ that: - accepts all Boolean functions computable by polynomial-size circuits, and - rejects a $1/\text{poly}(n)$ fraction of all Boolean functions.

Proof. This is exactly the Razborov–Rudich (1997) natural proofs barrier. A property satisfying both conditions would be a "natural" lower-bound proof, which Razborov and Rudich show is impossible if pseudorandom generators (and hence one-way functions) exist. □

In plain language: **you cannot efficiently distinguish easy Boolean functions from hard ones.** Any test you could apply would either accept too many hard functions or reject too many easy ones. The difficulty is hidden — encrypted, in a precise sense — within the function's structure.

This is not a limitation of current techniques. It is a *structural feature* of the discrete world: the same property that makes cryptography possible (hard functions look random) is the property that makes P vs NP unprovable by standard methods (hard functions are indistinguishable from random).

6.3.2 The Self-Referential Loop

The self-hiding property creates a remarkable self-referential structure in discrete complexity:

1. **Hard Boolean functions exist** (Shannon 1949: a random function requires exponential circuits).
2. **But you cannot identify them efficiently** (Razborov–Rudich: no natural proof).
3. **Because if you could, you would break cryptography** (the identifying test would be a distinguisher for pseudorandom generators).
4. **And cryptography is precisely the assumption that hard functions exist and look random.**

The loop closes: the *existence* of hard functions implies the *unidentifiability* of hard functions. Difficulty bootstraps its own concealment.

Smooth systems break this loop entirely. Analytic functions do not "look random" — they have global structure (holomorphic extensions, spectral decay, singularity locations) that is detectable from local data. There is no smooth analogue of one-way functions, no smooth analogue of pseudorandomness, and therefore no self-hiding. The difficulty is visible, extractable, and decidable.

6.3.3 Structural Decidability

This analysis suggests a general principle:

Conjecture 1 (Structural Decidability Principle). The fundamental complexity-theoretic question — "is this system easy or hard?" — over a domain $\mathcal{D}$ is decidable if and only if $\mathcal{D}$ admits an **extractable difficulty invariant**: a computable parameter $\rho : \mathcal{D} \to [0, \infty]$ such that $\rho(\mathcal{S}) > 1$ iff $\mathcal{S}$ is "easy" (admits efficient representation/computation).

Domain $\mathcal{D}$	Extractable difficulty invariant?	Fundamental question decidable?
Smooth functions ($\mathcal{F}_\rho$)	Yes: ρ (analyticity parameter)	**Yes** — Theorem 4
Boolean functions ($\{0,1\}^n \to \{0,1\}$)	No (if OWF exist: Razborov–Rudich)	**No** — P vs NP is open
Ergodic Markov chains	Yes: spectral gap Δ	**Yes** — mixing time $= \Theta(1/\Delta)$
CSPs (constraint satisfaction)	Partial: constraint density α	**Partial** — dichotomy theorems (Bulatov 2017, Zhuk 2020)
Quantum systems	Yes: Lindblad gap Δ_L	**Yes** — coherence/decoherence boundary

The conjecture reframes the difficulty of P vs NP: it is not that we lack a proof technique — it is that the Boolean domain *provably* (under OWF) lacks the extractable invariant that would make the question answerable. The three barriers (relativization, natural proofs, algebrization) are symptoms of this structural absence, not independent obstacles.

The Latent framework as proof-by-contrast. The strongest evidence for the Structural Decidability Principle is the Latent framework itself. The *same type of question* — "is there a sharp boundary between easy and hard?" — receives a complete, decidable answer in the smooth domain and remains open in the discrete domain. The only difference is the domain's structure. If the conjecture is correct, then:

- **Resolving P vs NP** requires either (a) proving OWF do not exist (unlikely, and would collapse cryptography) or (b) developing proof techniques that work *without* an extractable difficulty invariant — a fundamentally new kind of argument.
- **No smooth technique will ever resolve P vs NP**, because smooth techniques rely on the very structure (analyticity, spectral decay) that the discrete world lacks.
- **The GCT program** (Mulmuley & Sohoni 2001) — which attempts to import algebraic geometry structure into the Boolean world — is, in this light, an attempt to *enrich the discrete domain* with enough structure to make the question answerable. Whether this is possible is itself an open question.

6.4 The Kolmogorov Complexity Connection

The most precise connection between Latent and classical description complexity runs through Kolmogorov complexity.

Definition 10 (Kolmogorov Complexity). The **Kolmogorov complexity** $K(x)$ of a string x is the length of the shortest program (on a universal Turing machine) that outputs x.

$K(x)$ is uncomputable — there is no algorithm that takes x and outputs $K(x)$ (Li & Vitányi 2008). This is a consequence of the halting problem.

Theorem 5 (Latent Complexity as Computable Kolmogorov Complexity). For smooth systems:

(a) **Latent complexity bounds Kolmogorov complexity.** If $\mathcal{S} \in \text{L-ANALYTIC}(\rho)$ and x_ε is the binary encoding of Λ_{N^*} at precision ε, then:

$$K(x_\varepsilon) \leq N^*(\varepsilon) \cdot \lceil \log_2(1/\varepsilon) \rceil + O(1) = O\left(\frac{\log^2(1/\varepsilon)}{\log \rho} \right)$$

(b) **The bound is tight up to logarithmic factors.** No description of $\mathcal{S}$ to accuracy ε can be shorter than $N^*(\varepsilon) = \Omega(\log(1/\varepsilon)/\log \rho)$ real numbers (by the n-width lower bound, Nagy 2026b, Theorem 8), corresponding to $\Omega(\log^2(1/\varepsilon)/\log \rho)$ bits.

(c) **Latent complexity is computable.** Unlike $K(x)$, $\rho(\mathcal{S})$ is computable from data (Theorem 4a). For parametric systems, it is available in closed form.

Proof. Part (a): The Latent Λ_{N^*} consists of N^* real numbers, each quantized to ε precision, requiring $\lceil \log_2(1/\varepsilon) \rceil$ bits per number. A program that stores these numbers and evaluates $\Pi(\Lambda_{N^*})$ has length at most $N^* \cdot \lceil \log_2(1/\varepsilon) \rceil + c$ for a constant c (the program code). Part (b): The n-width lower bound shows N^* numbers are necessary. Part (c): By Theorem 4. □

Interpretation. Latent complexity is a *computable upper bound* on Kolmogorov complexity, restricted to smooth systems. It does not solve the halting problem — it solves a restricted version of the description complexity problem for a specific class of objects. The restriction to smooth systems is essential: for arbitrary strings, no computable bound on $K(x)$ exists.

Latent complexity relates to Kolmogorov complexity as a computable approximation relates to an uncomputable ideal:

$$\underbrace{K(x_\varepsilon)}_{\text{uncomputable}} \leq \underbrace{N^*(\varepsilon) \cdot \log(1/\varepsilon)}_{\text{computable from } \rho} \leq \underbrace{|x|}_{\text{trivial upper bound}}$$

For smooth systems, the Latent bound is tight (up to log factors). For non-smooth systems, the Latent bound may not exist ($\rho = 1$), and we fall back to the trivial bound — which is the classical situation.

---

7. Relationships to Classical Complexity Measures

7.1 Overview

Latent complexity relates to, but is distinct from, seven classical complexity measures. The relationships are summarized in the following table and detailed below.

Classical measure	What it captures	Computable?	Latent relationship
Kolmogorov complexity $K(x)$	Shortest description (bits)	No	$K(x_\varepsilon) = O(N^*(\rho, \varepsilon) \cdot \log(1/\varepsilon))$ — Latent gives computable bound (§6.4)
Shannon entropy $H(X)$	Information content (bits)	Yes (from distribution)	Independent: high-H systems can have any ρ. Entropy measures information; ρ measures regularity.
VC dimension d_{VC}	Hypothesis class capacity	Problem-specific	$d^*_{\text{VC}} = O(N^*(\rho, \varepsilon))$ — the needed model capacity is bounded by description size
Rademacher complexity $\mathcal{R}_n$	Empirical complexity	Yes (from data)	Related to effective rank; ρ gives the rate, $\mathcal{R}_n$ gives the constant
Kolmogorov n-widths d_n	Optimal n-dim approximation error	Over function classes, not instances	$d_n(\mathcal{F}_\rho) = \Theta(\rho^{-n})$ — exact rate (Nagy 2026b, Theorem 8)

Classical measure	What it captures	Computable?	Latent relationship
Communication complexity	Bits to transmit between parties	Problem-specific	Grade determines the communication structure: grade-r requires r-party communication
Circuit complexity	Minimum circuit size	Upper bounds known	Grade $\leftrightarrow$ depth (Theorem 3); $N^* \leftrightarrow$ width

7.2 Shannon Entropy

Shannon entropy $H(X) = -\sum p_i \log p_i$ measures how many bits are needed to encode the *output* of a random variable. It says nothing about the *regularity* of the distribution:

- A Gaussian with large variance: H is large (many bits), but $\rho = \infty$ (entire — trivially compressible).
- A Cauchy distribution: $H = \log(\pi e \cdot 4\gamma)$ is finite, but $\rho = 1$ (genuinely hard).
- A discrete uniform on $\{1, \ldots, n\}$: $H = \log n$, but $\rho^* = \infty$ (characteristic function is a polynomial in e^{it}).

Entropy measures the amount of *surprise* in the system's output. ρ measures the *structure* of the system's representation. A system can be highly surprising (high entropy) yet highly structured (high ρ), or vice versa.

7.3 VC Dimension and Statistical Learning

VC dimension measures the capacity of a hypothesis class — how many data points it can shatter. For learning a system $\mathcal{S} \in$ L-ANALYTIC(ρ):

Proposition 1 (Latent Complexity Bounds Sample Complexity). To learn $\mathcal{S}$ to accuracy ε with probability $1 - \delta$, it suffices to use a model class with VC dimension:

$$d_{\text{VC}} \geq N^*(\rho, \varepsilon) = O\left(\frac{\log(1/\varepsilon)}{\log \rho}\right)$$

and the number of samples needed is:

$$m = O\left(\frac{N^*(\rho, \varepsilon) \log(N^*/\delta)}{\varepsilon^2}\right)$$

This connects ρ directly to statistical learning theory: the Latent Number determines the model size, and the model size determines the sample complexity. The scaling laws for neural networks (Kaplan et al. 2020) are an empirical manifestation of this: the power-law exponent in $L(N) \sim N^{-\alpha}$ is determined by $\alpha \approx \log \rho / \log 2$.

7.4 Communication Complexity

Communication complexity (Yao 1979) measures how many bits Alice and Bob must exchange to compute a joint function of their private inputs. For smooth systems, the grade provides a natural analogue:

Proposition 2 (Grade and Communication). Consider a d-component system $\mathcal{S}$ where component i is "held by" party i. The grade-r component $\Lambda^{(r)}$ requires the participation of r parties to compute. Specifically:

(a) Grade-1 components are locally computable (no communication needed).
(b) Grade-2 components require pairwise communication.
(c) Grade-r components require r-party communication.

The total communication cost to extract Λ_{N^*} is:

$$\text{Communication} = \sum_{r=2}^{r^*} \binom{d}{r} \cdot O(N_r^*)$$

where N_r^* is the description size of $\Lambda^{(r)}$. For most systems, grade decay ensures N_r^* decreases exponentially with r, making higher-order communication negligible.

7.5 The Smooth Analogue of P

We can define a smooth complexity class analogous to P:

Definition 11 (Smooth-P). A smooth system $\mathcal{S}$ is in **Smooth-P** if every bounded linear query Π on $\mathcal{S}$ is answerable in time polynomial in $\log(1/\varepsilon)$.

Theorem 6 (L-ANALYTIC $\subseteq$ Smooth-P). Every system in L-ANALYTIC is in Smooth-P. Specifically, for $\mathcal{S} \in$ L-ANALYTIC(ρ) and any bounded linear projector Π:

$$\text{Time}(\Pi, \varepsilon) = O\left(\frac{\log(1/\varepsilon)}{\log \rho}\right) = O(\log(1/\varepsilon))$$

which is polynomial (in fact, linear) in $\log(1/\varepsilon)$.

Proof. Direct consequence of the Description–Computation Theorem (Theorem 2a–b). □

Remark 2. The query complexity $O(\log(1/\varepsilon))$ for L-ANALYTIC systems is *better* than polynomial — it is logarithmic in the precision. In the language of classical CT, L-ANALYTIC problems are not just "in P" — they are in something closer to L (logarithmic space/time), at least for the query phase. The extraction phase may have higher complexity (up to $O(d^3)$ for eigendecomposition), but it is amortized over queries.

Open Question 1. Is the converse true? If every query on $\mathcal{S}$ is answerable in $O(\text{poly}(\log(1/\varepsilon)))$ time, does this imply $\rho > 1$? In other words, does Smooth-P = L-ANALYTIC? We conjecture no: there may exist systems that are efficiently queryable by clever algorithms (exploiting special structure) despite having $\rho = 1$. But we have no example.

---

8. Applications

8.1 Financial Portfolio Risk (L-ANALYTIC, Grade 2)

A portfolio of d correlated assets under GBM has: - $\rho \approx e^{\pi/(2\sigma_{\max}\sqrt{\lambda_1})} \in [1.05, 3.0]$, where $\sigma_{\max}$ is the largest volatility and λ_1 is the largest correlation eigenvalue - Effective grade $r^* = 2$ (pairwise correlations capture the structure; grade 3 is negligible for typical portfolios) - Extraction cost: $O(d^3)$ for eigendecomposition of the $d \times d$ correlation matrix

Worked example. S&P 500-like portfolio, $d = 500$, $\sigma_{\max} = 0.4$, correlation eigenvalue $\lambda_1 = 180$ (typical for equity markets, where the first eigenvalue captures $\sim 36\%$ of variance):

$$\rho = e^{\pi/(2\cdot 0.4\cdot\sqrt{180})} = e^{0.293} \approx 1.34$$

$$N^* = \left\lceil \frac{\log(10^6)}{\log(1.34)} \right\rceil = \left\lceil \frac{13.82}{0.293} \right\rceil = 48$$

Latent complexity fingerprint: $\mathfrak{C} = (1.34, 2, 1.25 \times 10^8)$ — where $\mathcal{C}_E = O(500^3)$ is the eigendecomposition cost.

Query	Latent cost	MC cost ($\varepsilon = 10^{-6}$)	Speedup
VaR (99%)	48 ops	10^{12} samples $\times$ 500 assets	$10^{13}\times$
Expected Shortfall	48 ops	$10^{12} \times 500$	$10^{13}\times$
Basket option price	48 ops	$10^{12} \times 500$	$10^{13}\times$
Delta–Gamma Greeks	48 ops $\times$ d	$10^{12} \times 500 \times 500$	$10^{12}\times$
1000 queries total	$1.25 \times 10^8 + 48{,}000$	5×10^{17}	$4 \times 10^9\times$

The extraction pays for itself after a single query. For 1000 queries (a typical risk desk daily batch), the Latent approach is $10^9\times$ faster. This is not a theoretical curiosity — it is why spectral risk methods work in production (Nagy 2026a, §7).

8.2 Turbulence (L-ANALYTIC $\to$ L-BOUNDARY as Re $\to \infty$)

Navier–Stokes turbulence at Reynolds number Re has: - $\rho \approx 1 + C \cdot \mathrm{Re}^{-3/4}$ — approaches the L-BOUNDARY as Re increases - Effective grade $r^* = 2$ (quadratic nonlinearity; the velocity field advects itself) - Extraction cost: $O(N_{\text{grid}}^3)$ for DNS, but $O(N^{*3})$ for spectral

Worked complexity trajectory. As Reynolds number increases, the Latent complexity of turbulence traces a continuous path from L-ENTIRE toward L-BOUNDARY:

Re	ρ	N^* ($\varepsilon = 10^{-3}$)	Class	Physical regime	Optimal method
1	∞	2	L-ENTIRE	Stokes flow (linear)	Analytical
100	1.5	18	L-ANALYTIC(1.5)	Laminar	Spectral

Re	ρ	N^* ($\varepsilon = 10^{-3}$)	Class	Physical regime	Optimal method
10^3	1.06	119	L-ANALYTIC(1.06)	Transition	Spectral (large N)
10^5	1.003	2,303	L-ANALYTIC(1.003)	Turbulent	High-order spectral
10^7	1.0001	69,077	Near L-BOUNDARY	Fully turbulent	DNS

The Kolmogorov $-5/3$ spectrum is a consequence of grade-2 structure: the energy cascade involves quadratic (grade-2) interactions that redistribute energy across scales. The Latent framework explains *why* the spectrum has this form — it is the footprint of grade-2 interaction in Fourier space.

The $\rho \to 1$ limit is where the Kolmogorov dissipation scale $\eta \sim \mathrm{Re}^{-3/4}$ matches the representation resolution: $N^* \sim \eta^{-1}$. This is the Latent-theoretic explanation for why DNS cost scales as $\mathrm{Re}^{9/4}$ in 3D — it is the cost of representing a system approaching L-BOUNDARY.

8.3 Neural Network Training (L-ANALYTIC, Grade 1–2)

A neural network learning a function from data with covariance Σ has: - $\rho = (\limsup_k (\lambda_k/\lambda_1)^{-1/k})^{-1}$ from the eigenvalue decay of Σ - Effective grade: 1 for the data distribution (grade 2 if considering the loss landscape Hessian) - Extraction cost: $O(d^2 m)$ for SVD of the $m \times d$ data matrix

The scaling laws $L(N) \sim N^{-\alpha}$ (Kaplan et al. 2020) are predicted by $\alpha = \log \rho / \log 2$. For text data ($\rho \approx 1.3$): $\alpha \approx 0.38$, matching the empirically observed $\alpha \approx 0.4$ in the Chinchilla paper (Hoffmann et al. 2022).

The optimal model size is $N^* = \log(1/\varepsilon)/\log \rho$. A model with more parameters than N^* overfits; fewer underfits. This resolves the "bitter lesson" (Sutton 2019) into a quantitative statement: scale helps because it approaches N^*, but scaling beyond N^* gives diminishing returns.

8.4 The Three-Body Problem (L-ANALYTIC, Grade 3)

The gravitational three-body problem is the cleanest example of grade-3 complexity: - ρ depends on the orbit: periodic orbits have $\rho \gg 1$ (compressible), chaotic orbits have $\rho \to 1$ (incompressible) - Effective grade $r^* = 3$: the irreducible three-body coupling is non-negligible - The grade-3 co-skewness tensor is a fingerprint of non-integrability

Worked complexity landscape. The equal-mass three-body problem (unit masses, unit gravitational constant) shows how ρ varies across initial condition space:

Orbit type	ρ	Grade r^*	$\|\Lambda^{(3)}\|/\|\Lambda^{(2)}\|$	N^* ($\varepsilon = 10^{-3}$)	Class
Figure-8 (Chenciner–Montgomery)	~ 12	3	0.03	3	L-ANALYTIC(12)
Lagrange equilateral	~ 50	2	< 0.001	2	L-ENTIRE

Orbit type	ρ	Grade r^*	$\|\Lambda^{(3)}\|/\|\Lambda^{(2)}\|$	N^* ($\varepsilon = 10^{-3}$)	Class
Hierarchical binary + distant	~ 3	3	0.15	6	L-ANALYTIC(3)
Random (generic chaotic)	~ 1.01	3	~ 5	691	L-ANALYTIC(1.01)
Near-collision	$\to 1$	3	~ 100	$\to \infty$	Approaching L-BOUNDARY

The Lagrange equilateral orbit has $r^* = 2$ (the three-body coupling vanishes by symmetry: each body sees a symmetric effective potential). The figure-8 orbit has $r^* = 3$ but small $\|\Lambda^{(3)}\|$ (the choreographic symmetry partially cancels three-body effects). Generic chaotic orbits have large $\|\Lambda^{(3)}\|$ — the irreducible three-body coupling dominates.

Latent-chaos. The three-body problem is in L-ANALYTIC — not L-BOUNDARY — for every specific initial condition (finite-time solutions are analytic). But ρ varies by orders of magnitude across the initial condition space. The Latent complexity fingerprint is a function $\mathfrak{C}(\text{IC})$, not a single triple.

This gives a precise definition: a system is **Latent-chaotic** if $\rho(\text{IC}) \to 1$ on a set of positive measure in initial condition space. The three-body problem is Latent-chaotic because almost all initial conditions have $\rho \approx 1$ (Poincaré's result, formalized). The grade acts as a "chaos amplifier": grade-3 coupling makes the system sensitive to the initial condition in a way that grade-2 systems (e.g., two-body: always integrable, always L-ENTIRE) cannot be.

8.5 Summary: The Complexity Fingerprint Atlas

The four applications above demonstrate that the Latent complexity fingerprint $\mathfrak{C} = (\rho, r^*, \mathcal{C}_E)$ is a practical tool, not an abstract classification. The following atlas summarizes the worked examples:

System	ρ	Grade r^*	$\mathcal{C}_E$	Class	N^* ($\varepsilon = 10^{-6}$)	vs. MC
S&P 500 portfolio ($d = 500$)	1.34	2	$O(d^3)$	L-ANALYTIC	48	$10^{13}\times$
NS at Re = 100	1.5	2	$O(N_{\text{grid}}^3)$	L-ANALYTIC	34	spectral wins
NS at Re = 10^7	1.0001	2	$O(N_{\text{grid}}^3)$	Near L-BOUNDARY	69,077	DNS needed
GPT-scale text model	1.3	1	$O(d^2 m)$	L-ANALYTIC	53	N^* = model size
Three-body (figure-8)	12	3	$O(1)$	L-ANALYTIC	6	trivial
Three-body (chaotic)	1.01	3	$O(T^3)$	L-ANALYTIC	1,382	barely tractable

System	ρ	Grade r^*	$\mathcal{C}_E$	Class	N^* ($\varepsilon = 10^{-6}$)	vs. MC
Cauchy distribution	1	1	—	L-SINGULAR	∞	both fail

The atlas reveals a pattern: **most systems of practical interest are L-ANALYTIC with $\rho \in [1.01, 10]$ and grade ≤ 3.** The L-ENTIRE and L-SINGULAR extremes are mathematically important but physically rare. The useful part of Latent complexity theory — the part that predicts computational costs and optimal methods — lives in L-ANALYTIC.

8.6 The Grade Method as Complexity Classifier

The preceding applications compute Latent complexity fingerprints from domain-specific analysis. A natural question: can the classification be automated?

The Grade Method (Nagy 2026h) provides the answer. Given any smooth dynamical system $\dot{x} = F(x)$, the 4-step protocol computes:

1. **State space and equilibrium** x^* where $F(x^*) = 0$
2. **Grade decomposition**: $A^{(k)} = D^k F(x^*)/k!$ for $k = 0, 1, 2, ...$
3. **Analyticity radius** ρ from the decay rate of $\|A^{(k)}\|$
4. **Effective grade** $k_{\text{eff}}(\varepsilon)$ and complexity class

The output directly determines the Latent complexity class:

Grade Method output	Latent complexity class	Reason
$\rho = \infty$ (polynomial, finite max grade)	**L-ENTIRE**	Entire function, super-exponential convergence
$\rho > 1$ (analytic, exponential decay)	**L-ANALYTIC**(ρ)	$N^* = \log(1/\varepsilon)/\log\rho$
$\rho = 1$ but smooth transform has $\rho^* > 1$	**L-BOUNDARY**	Algebraic convergence in natural domain
$\rho^* = 1$ (no smooth representation helps)	**L-SINGULAR**	Genuinely hard

Computed classification table. The Grade Method software (`tools.grade_method.grade_decompose`) applied to the systems from §§3–7 of the companion paper (Nagy 2026h) produces:

System	Grade	ρ_{eff}	k_{eff} ($\varepsilon = 0.01$)	Complexity class	N^* at $\varepsilon = 10^{-6}$
SIR ($R_0 = 0.5$)	2	6.00	2	L-ENTIRE	2
SIR ($R_0 = 3$)	2	1.11	44	L-ANALYTIC(1.11)	132
Lotka–Volterra	2	3.33	4	L-ENTIRE	12

System	Grade	ρ_{eff}	k_{eff} ($\varepsilon = 0.01$)	Complexity class	N^* at $\varepsilon = 10^{-6}$
Lorenz ($r = 0.5$)	2	∞	2	L-ENTIRE	2
Lorenz ($r = 28$)	2	0.27	—	(near L-BOUNDARY)	(large)
Wilson–Cowan	∞	0.85	—	L-BOUNDARY	(transform needed)
Hill ($n = 4$, $K = 1$)	∞	1.41	11	L-ANALYTIC(1.41)	40
Mass-action kinetics	2	1.22	23	L-ANALYTIC(1.22)	70

The table reveals the power of the classification: - **Polynomial ODE systems** (SIR, Lotka–Volterra, Lorenz) are L-ENTIRE when stable and L-ANALYTIC when the operating point is near a bifurcation. The Grade Method detects this automatically. - **Non-polynomial systems** (Wilson–Cowan, Hill) have finite analyticity radii determined by the nearest complex singularity. The Grade Method's ρ_{eff} captures this from numerical differentiation alone. - **The Lorenz system at** $r = 28$ has $\rho_{\text{eff}} < 1$ at the origin because the origin is unstable and the linearization dominates the grade spectrum with positive eigenvalues. The system operates in a different regime (strange attractor) where the equilibrium-based analysis does not capture the global structure — illustrating a limitation that the L-BOUNDARY class addresses.

The classification is computable. For any system where F can be evaluated, running grade_decompose(F, x_star, n) returns ρ and grade in $O(n^{k_{\max}})$ finite-difference evaluations. This is the Computability Theorem (Theorem 4) made concrete: not just "decidable in principle" but "computable in seconds."

---

9. Discussion and Open Problems

9.1 What Latent Complexity Achieves

Latent complexity provides a **computable, quantitative, multi-dimensional** complexity theory for smooth systems. The three axes — ρ (description), grade (interaction), extraction cost (computation) — give more information than any single classical measure. The theory is:

- **Complete**: every smooth system has a Latent complexity fingerprint.
- **Decidable**: the complexity class is computable from data (Theorem 4).
- **Sharp**: the bounds are tight (the n-width optimality, Nagy 2026b Theorem 8).
- **Actionable**: the fingerprint directly determines the optimal computational method and resource allocation.

9.2 What Latent Complexity Does Not Achieve

The theory has clear boundaries:

1. **It does not apply to discrete combinatorial problems.** Graph coloring, SAT, integer factoring — these are not smooth systems and have no Latent Number. The extended ρ^* (working with generating functions) partially bridges this gap for discrete distributions, but not for arbitrary combinatorial structures.

2. **It does not replace classical CT.** For Turing machines and Boolean circuits, the P vs NP question remains as important as ever. Latent complexity operates on a different domain and answers different questions.

3. **The extraction cost can be large.** For implicit systems (PDEs without closed-form solutions), computing ρ may require solving the problem first — a circularity. The theory assumes the system is extractable; when extraction itself is the hard problem, Latent complexity reduces to classical CT.

4. **Non-linear queries.** Theorem 2 covers bounded *linear* projectors. Non-linear queries (optimization, control, game-theoretic quantities) may require more than $O(N^*)$ operations even when the representation is finite.

9.3 Open Problems

1. **Smooth P = L-ANALYTIC?** Is every smooth system that is efficiently queryable (polynomial in $\log(1/\varepsilon)$) necessarily in L-ANALYTIC? Or are there $\rho = 1$ systems with clever query algorithms? This is the smooth analogue of P vs NP — and unlike the classical version, it may be provable.

2. **Grade–communication complexity.** Proposition 2 gives a qualitative correspondence between grade and communication complexity. Can this be made tight? Specifically: is the grade-r communication cost $\Omega(\binom{d}{r})$, or can clever protocols reduce it?

3. **The Latent complexity of specific PDEs.** What is ρ for the Navier–Stokes equations at a specific Reynolds number? For the Schrödinger equation with a specific potential? Turning the framework into quantitative predictions for specific systems of interest requires domain-specific analysis.

4. **Latent complexity for non-smooth systems.** The extended ρ^* handles some cases (discrete distributions with analytic CFs). Can the framework be extended further — perhaps using wavelet-based or multi-resolution analysis — to cover broader system classes? What is the natural extension of grade structure to non-smooth settings?

5. **The relationship between ρ and the Bekenstein–Hawking bound.** The Latent framework applied to black holes gives $\rho = \infty$ (the 3-parameter Kerr solution is entire). The entropy $S = A/4$ counts the system's microstates. Is there a Latent complexity interpretation of the holographic entropy bound — perhaps as a grade-truncated description size?

6. **Formal connection to random CSP thresholds.** The phase transition at $\rho = 1$ and the random k-SAT threshold share structural properties (sharp transition, easy above, hard below). Is there a formal mapping — perhaps through the partition function's analyticity strip — that connects the two?

7. **Global vs. local classification.** The Grade Method (§8.5) classifies a system around a single equilibrium. For systems with multiple equilibria (toggle switches, bistable circuits), each equilibrium has its own fingerprint. What is the *global* Latent complexity of a multi-

attractor system? A natural candidate: $\rho_{\text{global}} = \min_i \rho_i$ over all equilibria, reflecting that the system is only as tractable as its hardest attractor. But this ignores the transition dynamics between attractors, which may introduce additional complexity. Defining global Latent complexity for systems with non-trivial basins of attraction remains open.

8. **Stochastic complexity.** For SDEs $dX = F(X)dt + \sigma(X)dW$, the Grade Method extends to stochastic settings (Nagy 2026h, §9), but the complexity theory for stochastic systems has additional structure: the noise can smooth out singularities ($\rho^*_{\text{stochastic}} > \rho^*_{\text{deterministic}}$) or introduce new ones (noise-induced transitions). A stochastic Latent complexity class incorporating both deterministic structure and noise-induced regularity would unify the descriptions-from-data approach with the analytical classification.

---

10. Conclusion

We have constructed a complexity theory for smooth systems based on the Latent framework. The three axes — Latent Number ρ (description complexity), effective grade r^* (interaction complexity), and extraction cost $\mathcal{C}_E$ (computational complexity) — together determine a system's intrinsic difficulty class. The four complexity classes L-ENTIRE, L-ANALYTIC, L-BOUNDARY, and L-SINGULAR form a strict hierarchy, and the classification is decidable for specific systems.

The central results are:

1. **Strict hierarchy** (Theorem 1): L-ENTIRE $\subsetneq$ L-ANALYTIC $\subsetneq$ L-BOUNDARY $\cup$ L-SINGULAR, with explicit separating examples.
2. **Description bounds computation** (Theorem 2): a system representable in N^* numbers is queryable in $O(N^*)$ operations — for smooth systems, representation IS computation.
3. **Grade measures interaction depth** (Theorem 3): the effective grade corresponds to the depth of interaction circuits, connecting interaction complexity to classical circuit depth.
4. **Latent complexity is decidable** (Theorem 4): ρ is computable from data, grade is measurable, and the complexity class is determinable — in contrast to classical CT where membership in P remains unresolved for most problems.
5. **L-ANALYTIC $\subseteq$ Smooth-P** (Theorem 6): analytic systems are efficiently queryable in $O(\log(1/\varepsilon))$ time.
6. **Computable Kolmogorov complexity** (Theorem 5): for smooth systems, Latent complexity provides a computable upper bound on Kolmogorov complexity, tight up to logarithmic factors.

The theory is not merely abstract. The companion Grade Method paper (Nagy 2026h) provides a 4-step computational protocol and open-source software that takes any smooth dynamical system and returns its Latent complexity fingerprint (ρ, grade, k_{eff}) — making the classification concrete and automated across eight scientific domains.

For four centuries, science has studied smooth systems without a complexity theory adapted to them. Classical complexity theory was built for a different domain. The Latent framework fills this gap: it provides the same kind of structural insight for smooth systems that P vs NP provides for discrete problems — except that the fundamental questions are answerable.

The practical message is simple: **before computing anything about a smooth system, measure ρ and grade.** These two numbers — one continuous, one discrete — tell you how hard the

system is, what kind of structure it has, and which computational method to use. The tools exist. The classification is computable. The era of blind computation should end.

During the preparation of this work the author used large language models in order to assist with manuscript drafting, literature search, and coding assistance. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the content of the published article.

References

- Aaronson, S. & Wigderson, A (2009). Algebrization: a new barrier in complexity theory. *ACM Trans. Comput. Theory*, 1(1), 1-2.
- Arora, S. et al (1998). Proof verification and the hardness of approximation problems. *J. ACM*, 45(3), 501-555.
- Arora, S. & Safra, S (1998). Probabilistic checking of proofs: a new characterization of NP. *J. ACM*, 45(1), 70-122.
- Babenko, K. I (1958). On the best approximation of a class of analytic functions. *Izv. Akad. Nauk SSSR Ser. Mat.*, 22(5), 631-640.
- Baker, T., Gill, J. & Solovay, R (1975). Relativizations of the P = ? NP question. *SIAM J. Comput.*, 4(4), 431-442.
- Blum, L., Shub, M. & Smale, S (1989). On a theory of computation and complexity over the real numbers. *Bull. Amer. Math. Soc.*, 21(1), 1-46.
- Bulatov, A. A (2017). A dichotomy theorem for nonuniform CSPs. *Proc. 58th FOCS*, 319-330.
- Cook, S. A (1971). The complexity of theorem-proving procedures. *Proc. 3rd STOC*, 151-158.
- Hoeffding, W (1948). A class of statistics with asymptotically normal distribution. *Ann. Math. Stat.*, 19(3), 293-325.
- Hoffmann, J. et al (2022). Training Compute-Optimal Large Language Models. *NeurIPS 2022*. DOI: 10.1101/2024.06.06.597716
- Kaplan, J. et al (2020). Scaling Laws for Neural Language Models. *arXiv:2001.08361*.
- Karp, R. M (1972). Reducibility among combinatorial problems. In *Complexity of Computer Computations*, 85–103. Plenum. *Complexity of Computer Computations*, 85-103.
- Kolmogorov, A. N (1936). Über die beste Annäherung von Funktionen einer gegebenen Funktionenklasse. *Ann. Math.*, 37(1), 107-110.
- Li, M. & Vitányi, P (2008). An Introduction to Kolmogorov Complexity and Its Applications. *An Introduction to Kolmogorov Complexity and Its Applications*.
- Mézard, M. & Zecchina, R (2002). Random K-satisfiability problem: from an analytic solution to an efficient algorithm. *Phys. Rev. E*.
- Pinkus, Allan (1985). n-Widths in Approximation Theory. Springer. DOI: 10.1007/978-3-642-69894-1
- Mulmuley, K. D. & Sohoni, M (2001). Geometric complexity theory I: an approach to the P vs. NP and related problems. *SIAM J. Comput.*, 31(2), 496-526.
- Razborov, A. A. & Rudich, S (1997). Natural proofs. *J. Comput. System Sci.*, 55(1), 24-35.
- Shamir, A (1992). IP = PSPACE. *J. ACM*, 39(4), 869-877.
- Shannon, C. E (1949). The synthesis of two-terminal switching circuits. *Bell System Technical Journal*, 28(1), 59-98.

- Sutton, R. S (2019). The bitter lesson. Blog post.
- Trefethen, Lloyd N. (2013). Approximation Theory and Approximation Practice. SIAM. DOI: 10.1137/1.9781611975949
- Yao, A. C.-C (1979). Some complexity questions related to distributive computing. *Proc. 11th STOC*, 209-213.
- Zhuk, D (2020). A proof of the CSP dichotomy conjecture. *J. ACM*, 67(5), 1-30.
- Nagy, T. (2026). The Latent: Finite Sufficient Representations of Smooth Systems. *Zenodo*. DOI: 10.5281/zenodo.19101209
- Nagy, T. (2026). The Latent Number : A Universal Diagnostic for Computational Complexity. *Working paper*.
- Nagy, T. (2026). Theory and Measure of Decomposability. *Working paper*.
- Nagy, T. (2026). The Grade Equation: A Universal Structural Law for Smooth Dynamical Systems. *Working paper*.
- Nagy, T. (2026). Neural Scaling Laws Formalized: Why Chinchilla Works (A Machine-Verified Derivation). *Working paper*.
- Nagy, T. (2026). Turbulence Scaling Laws from the Grade Equation: Kolmogorov Spectrum and Intermittency from Analyticity. *Working paper*.
- Nagy, T. (2026). The Exact Latent Solution of the Gravitational N-Body Problem. *Zenodo*. DOI: 10.5281/zenodo.19102064
- Nagy, T. (2026). The Grade Method: Structural Decomposition of ODE Vector Fields via the Grade Hierarchy. *Working paper*.