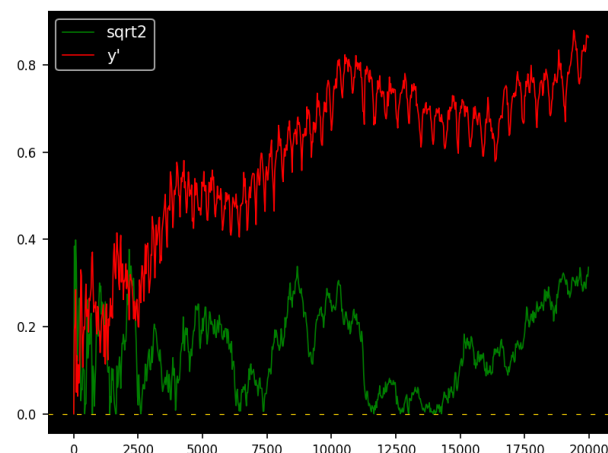
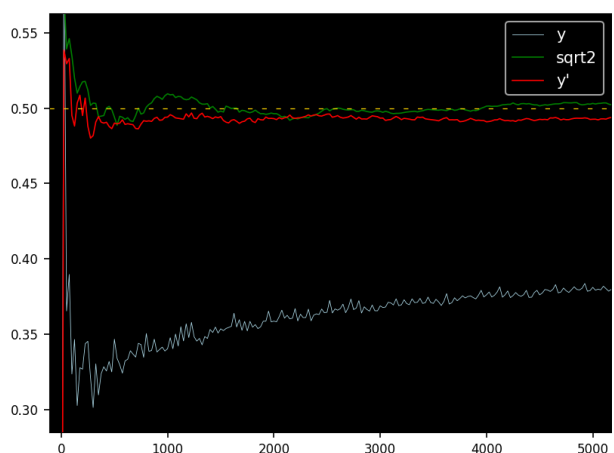
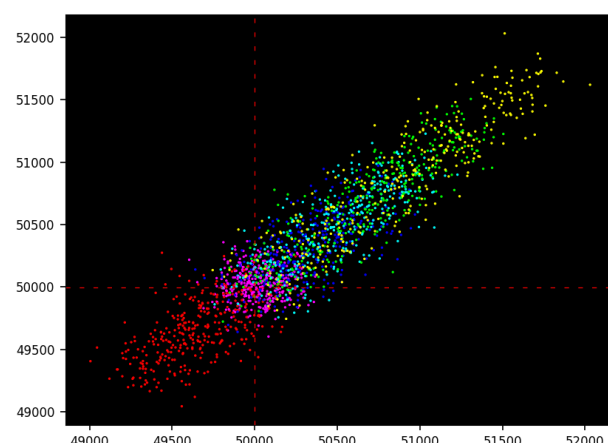
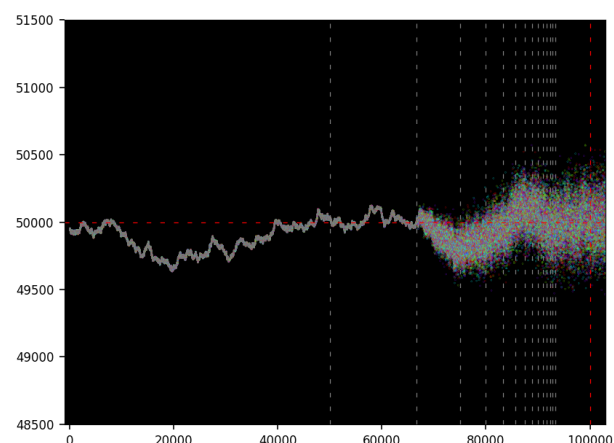
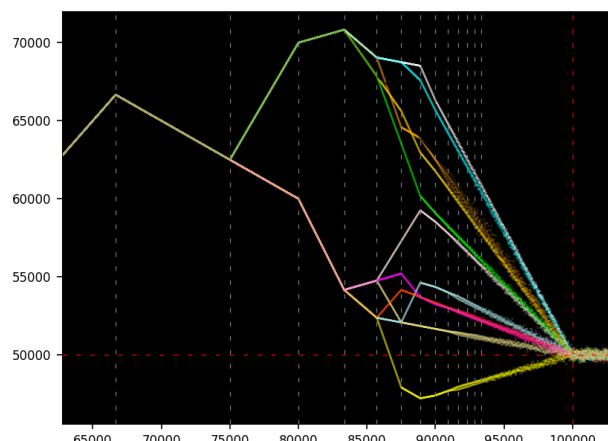
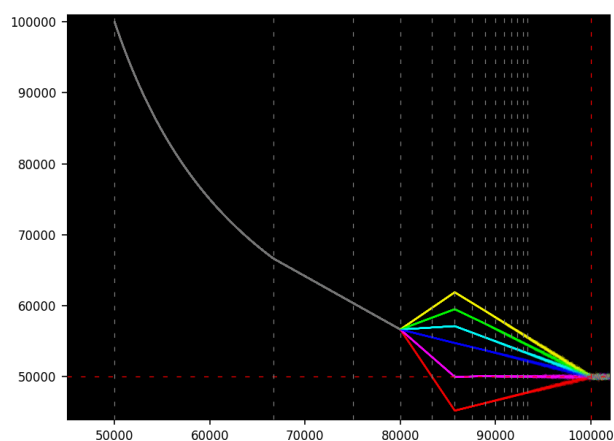

Digit Distribution of Classic Constants

Applications to Fintech, AI, Quantum Systems, Data Synthesis
LLMs, Cybersecurity and High-Performance Computing



Contents

1	Link Between the Digit Sum Function and Auto-Convolutions	6
1.1	A new type of string operators	6
1.1.1	String class	6
1.1.2	Truncation, number representation, and convergence	6
1.1.3	String convolution and square root	7
1.1.4	Well-balanced strings	7
1.2	Infinite sequences of iterated auto-convoluted strings	7
1.2.1	Visualizing iterated auto-convoluted string sequences	8
1.2.2	Fundamental result about the number of zeros and ones	9
1.3	Solving one of the greatest mathematical mysteries	12
1.3.1	Testing different seeds	13
1.3.2	Another interesting sequence	15
1.3.3	Surprising, unpublished results about the digit distribution	15
1.3.4	Application to cryptography	15
1.3.5	Python code	16
2	Petabyte Challenge to Help Crack the Famous Math Conjecture	19
2.1	Introduction	19
2.2	Spectacular behavior of the digit sum function	20
2.2.1	Potential scenarios	20
2.2.2	Dynamics of the unbroken bifurcation process	21
2.3	Case studies	22
2.4	An extreme case	23
2.5	Applications and AI Challenge with petabytes dataset	24
2.5.1	AI challenge	24
2.5.2	The dataset	25
2.6	Python code	26
3	From Digit Sum to Universal Dataset and Benchmarking AI Algorithms	29
3.1	Introduction	29
3.2	Deep dive into the digit sum function	30
3.2.1	Digit sum function: examples	30
3.2.2	Spectacular behavior of digit sum with primorials	31
3.2.3	Future research	32
3.2.4	References	33
3.2.5	Comparison with standard methodology	33
3.3	Infinite dataset and applications	34
3.4	Python code	34
3.4.1	Forward iterations	35
3.4.2	Backward iterations	38
4	Quantum Dynamics, Logistic Map, and Digit Distribution of Special Constants	41
4.1	Introduction	41
4.2	Logistic map and the digit sum function	42
4.2.1	Model comparison, with illustrations	42
4.2.2	Normality of special math constants	45
4.2.3	Applications and references	45
4.3	Re-balancing an uneven digit distribution	46
4.3.1	Digit-balancing transforms	46

4.3.2	Digit block balancing	48
4.4	Conclusion	52
4.5	Main Python code	53
5	Test of Normality and Digit Distribution of Algebraic Numbers	56
5.1	Simple normality test with application to PRNGs	56
5.1.1	High-performance computing with Chebyshev polynomials	57
5.1.2	Application with test of randomness and Python code	58
5.1.3	Problem and solution	60
5.2	Another interesting discrete quadratic dynamical system	60
5.2.1	Case with multiple limits	61
5.2.2	Case with single limit	61
5.3	Surprising results about the digit distribution	62
5.3.1	Python code for the computer-assisted proof of the main theorem	64
5.3.2	Python code for the deeper theorem	66
5.4	Strong patterns found in the digits of algebraic numbers	67
5.4.1	Python code to compute the digits	69
5.5	Correlated bit strings: seminal result and applications	70
5.5.1	Autocorrelations in related sequences	72
5.5.2	Python code	72
6	Quantum States and the Riemann Zeta Function	74
6.1	Synthetic primes, quantum states, and the Riemann Hypothesis	74
6.1.1	Definitions	74
6.1.2	Building a Beurling eta function by deletion	75
6.1.3	Building a Beurling eta function by swapping	75
6.1.4	Applications and Python code	76
6.2	Quantum derivatives, GenAI, and the Riemann Hypothesis	82
6.2.1	Cornerstone result to bypass the roadblocks	83
6.2.2	Quantum derivative of functions nowhere differentiable	84
6.2.3	Project and solution	85
6.2.4	Python code	89
7	Quantum, chaotic and fractal types of algorithmic convergence	94
7.1	Digit count generating function and fractal convergence	94
7.2	Other examples of chaotic convergence	96
7.2.1	Smooth convergence but with multiple branches	96
7.2.2	Chaotic convergence with multiple branches	96
7.2.3	Deep dive into the chaotic case	100
7.2.4	Interesting connection between 3^n and the digits of $\sqrt{2}$	102
7.3	Random polynomials and spectral analysis of digit distributions	105
7.3.1	Orbits of the digit generating function	105
7.3.2	Connection to Littlewood, Shapiro, and Newman Polynomials	107
7.3.3	Spectral signature of even versus uneven bit strings	108
7.3.4	Mesmerizing video featuring 1000 cases, with Python code	110
8	Towards Deterministic AI: The NPG Random Generator	115
8.1	NPG: fast non-periodic pseudo-random number generator	115
8.1.1	NPG non-standard fundamental examples	115
8.1.2	Deterministic vs cryptographically secure NPG vs other PRNGs	116
8.1.3	NPG success vs PCG64 and PCG64DXSM statistical failures	117
8.1.4	Optimum parameters to speed up computations	119
8.2	Compute time and Python code	119
8.2.1	NPG source code	120
8.2.2	PRNG tests of randomness	124
9	Gilbreath's Conjecture: Fintech and Cybersecurity Applications	128
9.1	Introduction	128
9.2	Proof for sifted sequences: standard and reverse sieving	129
9.3	Full list of short sequences and sequence categorization	132
9.3.1	Testing an exhaustive infinite list of admissible short sequences	132

9.3.2	Efficient sequence corridors	133
9.3.3	Hidden patterns causing failures	134
9.4	Synthetic sequences with Poisson gaps to mimic prime numbers	136
9.4.1	High performance computing to discover rare failure patterns	137
9.4.2	Prime gaps and forbidden prime constellations	138
9.5	A deeper version of Gilbreath's conjecture with partial proof	139
9.5.1	Distance between a successful sequence and its closest failing sister	140
9.5.2	General framework, proof for magic primes and other cases	141
9.5.3	Canonical form reduction and equivalence classes for sequences	146
9.6	Applications: cybersecurity, fraud detection and Fintech	148
9.6.1	Hidden patterns, error detection, checksum, and data quality audit	148
9.6.2	Fraud detection and transaction score synthesis	148
9.6.3	Cellular automata random generator and new tests of randomness	149
9.6.4	Unusual patterns in time series: modeling, detection and synthesis	150
9.6.5	Security issues in ciphers relying on prime numbers	152
9.7	Conclusions	152
Appendix A The Pi Day xLLM agent		153
Appendix B Convolution, Approximations, and Signal Processing		156
B.1	Approximations to mathematical function	156
B.1.1	Finding the roots of ζ with fast-converging series	156
B.1.2	Approximation based on quantization	157
B.2	Non-causal discrete convolution with Gaussian kernel	158
B.2.1	Problem	158
B.2.2	Solution	159
B.2.3	Python code	160
Appendix C Python Code for Gilbreath's Conjecture		163
C.1	Main program	163
C.2	Gilbreath library	169
Bibliography		179
Index		182

Introduction

Since the first edition entitled “0 and 1 – From Elemental Math to Quantum AI” and released in early 2025, a lot of progress has been made. This fourth edition offers a trip deep into one of the most elusive multi-century old conjectures in number theory: are the binary digits of the fundamental math constants evenly distributed? No one even knows if the proportions of ‘0’ and ‘1’ exist, for any of them: it could oscillate forever between 0% and 100%.

The book includes new chapter 5 on testing randomness with a much simplified version of Weyl’s criterion. It also features a breakthrough result regarding the binary digit distribution, stating that the proportion of 1 must lie between $\frac{5}{16}$ and $\frac{11}{16}$ for a large class of numbers including all the standard mathematical constants such as π, e or $\sqrt{2}$. Details with a computer-assisted proof are in chapter 5, published here for the first time. In another example, I use quadratic dynamical systems on a matrix space with Chebyshev polynomials and self-convolution to uncover beautiful results. Truncated self-convolution of strings and dynamical systems are also core topic in chapters 1–4. Random polynomials in chapter 7 lead to a spectacular data animation and new developments on the subject. Chapter 9 features new interesting topics related to prime numbers: forbidden constellations, reverse sieving, and magic primes, with a connection to cellular automata.

The style emphasizes simple English even when covering advanced topics, avoiding jargon and advanced mathematics when not necessary. I included enterprise-grade Python code for scientific and high-performance computing with the Gmpy2 library, numerous high-quality illustrations, a comprehensive clickable index and bibliography, along with efficient algorithms not taught in any classroom or textbook. The target audience includes professionals in computer science, physics, AI, machine learning, engineering, quantitative finance, and related fields, as well as students and beginners with one year of exposure to college-level mathematics and Python.

The material opens up new fundamental research areas in theoretical and computational number theory, numerical approximation, dynamical systems, quantum dynamics, and the physics of numbers. It has a strong emphasis on applications: automated pattern detection and theorem proving with AI, agent-based modeling, building a universal unbiased pattern-rich synthetic dataset, cryptography (fast, strong random number generators based on irrational numbers), dynamical systems with chaos detection and isolation, computer-intensive simulations, and high performance computing to handle numbers such as $2^n + 1$ at power 2^n with $n = 10^6$.

Each chapter is self-contained and can be read separately from the others. Most feature results published here for the first time. Chapter 6 and appendix B are new additions and contain a mix of theory, applications, and off-the-beaten path problems with solution. Quantum states and the Riemann zeta function are central themes in each of them. The section on signal processing and discrete convolution is a strong introduction to the topic, serving as a cheat sheet for practitioners or as a solid presentation for beginners, summarizing in a few pages material usually spread over several chapters. Finally, the new chapter 8 discusses a cryptographically secure PRNG that outperforms the most recent addition to the Numpy library, both in terms of speed and randomness. Chapter 9, also a new addition, covers the most recent developments regarding Gilbreath’s conjecture, with applications to HPC, cybersecurity, Fintech, PRNGs, chaos modeling, and fraud detection.

About the author

Vincent Granville is a pioneering AI builder, co-founder at Data Science Central (acquired by TechTarget), co-founder and CAIO at [Bonding AI](#), author, patent owner, expert witness and investor included Limited Partner at CalculusVC, a Bay Area VC firm. Vincent worked with Visa, Wells Fargo, eBay, NBC, Microsoft, CNET and several startups. He is also a top AI influencer for NVIDIA and other brands. His AI newsletter has 200,000 subscribers.



Vincent is a former post-doc at University of Cambridge. He published in *Journal of Number Theory*, *Journal of the Royal Statistical Society* (Series B), and *IEEE Transactions on Pattern Analysis and Machine Intelligence* (500+ citations). He is the author of multiple books, available [here](#), including “Synthetic Data and Generative AI” (Elsevier, 2024). Vincent lives in Washington state, and enjoys doing research on stochastic processes, dynamical systems, probabilistic and computational number theory.