MATHESIS

THE MATHEMATICAL FOUNDATIONS OF COMPUTING

"In mathematics, you don't understand things. You just get used to them."

— JOHN VON NEUMANN

Mahdi

LIVING FIRST EDITION · 2025

MATH

The Mathematical Foundations of Computing

"From ancient counting stones to quantum algorithms every data structure tells the story of human ingenuity."

Living First Edition

Updated October 31, 2025

LICENSE & DISTRIBUTION

MATHESIS:

A Living Architecture of Computing

DISTRIBUTION & SOURCE ACCESS

Repository: The complete source code (LaTeX, diagrams, examples) is available at:
https://github.com/m-mdy-m/algorithms-data-structures/tree/main/books/books

Preferred Citation Format:
Mahdi. (2025). *MATHESIS*. Retrieved from
https://github.com/m-mdy-m/algorithms-data-structures/tree/main/books/books

Version Control: This is a living document. Check the repository for the most current version and revision history.

WARRANTIES & DISCLAIMERS

TECHNICAL SPECIFICATIONS

Typeset with: LaTeX using Charter and Palatino font families
Graphics: TikZ and custom illustrations
Standards: Follows academic publishing conventions
Encoding: UTF-8 with full Unicode support
Format: Available in PDF, and LaTeX source formats

License last updated: October 31, 2025

For questions about licensing, contact: bitsgenix@gmail.com

Contents

Preface

I KEPT HITTING a wall. Not in my codethat worked fine. Not in algorithmsI could trace through them step by step. The wall was deeper. It was in understanding *why* things worked.

You know that feeling when you follow a proof mechanically, nodding along, getting the right answer... but something's missing? That was me. I could solve problems. I couldn't *see* them.

Writing *Arliz* and *The Art of Algorithmic Analysis* forced me to confront this. How can you explain something you don't truly understand? I'd write a section on recurrence relations, realize I was just regurgitating formulas, delete it, and start over. Again. And again.

So I stopped. Put everything on hold. Went back to the beginning.

Not back to "Intro to Discrete Math." Further. Back to Aristotle trying to formalize human reasoning. To al-Khwarizmi figuring out how to solve equations systematically. To Euclid asking "what can we build from almost nothing?" To Ibn Sina synthesizing Greek logic with Islamic mathematics. To Descartes having his crazy idea that geometry and algebra were the same thing.

I read their actual works. Not summaries. Not textbooks about them. Their words.

And something clicked.

These people weren't just discovering math. They were *thinking*wrestling with hard problems, making wrong turns, having insights, building frameworks. Mathematics wasn't this pristine thing handed down from on high. It was messy. Human. Incomplete. Always evolving.

That's what I want to share here.

What This Book Is

Mathesis is my attempt to understand mathematics the way it actually developedas a series of insights, each solving a real problem or answering a genuine question.

Not "here are 50 formulas to memorize" but "here's why someone needed this idea, here's what they were trying to do, here's how it connects to everything else."

This book sits between three others I'm writing:

- *Mathesis* the mathematical foundations
- *The Art of Algorithmic Analysis* how to analyze algorithms rigorously
- *Arliz* data structures in depth

You can read any of them independently. But together they form a path from "what is a number?" to "how do we build efficient software?"

How It's Different

Most math-for-CS books treat mathematics as vegetables you have to eat before dessert. Get through the boring prerequisite chapters, then you can do the fun stuff.

That's backwards.

Mathematics *is* the fun stuff. It's just taught badly.

I'm not going to give you formulas without context. Every major idea in here starts with a question: What problem were people trying to solve? Why did existing tools fail? What insight made progress possible?

Sometimes that means historical contextseeing how Babylonians tackled problems differently than Greeks, or how Islamic scholars built bridges between cultures. Sometimes it means showing failed approaches that seem reasonable but don't work. Sometimes it means proving something rigorously because the proof itself is enlightening.

The goal isn't to turn you into a mathematician. It's to give you *mathematical intuition*the ability to look at a problem and think "oh, this is really about X" or "I bet Y technique would work here."

Who Helped (Across Centuries)

I owe debts to people I'll never meet. Aristotle for showing thought could be systematic. Al-Khwarizmi for making algebra algorithmic. Ibn Sina for treating math as part of a bigger intellectual picture. Descartes for unifying geometry and algebra. Leibniz for dreaming of universal logical languages. Turing for proving limits of computation.

Also to readers of my other books who asked questions that made me realize I didn't understand something as well as I thought. You made this better.

A Warning

This is hard. Not "memorize 100 formulas" hard. "Change how you think" hard.

There will be moments where your brain hurts. Where a proof seems impossible to follow. Where you read the same paragraph five times and still don't get it.

That's normal. That's the process. Every mathematician goes through it.

The reward? Eventuallymight take days, might take monthssomething clicks. Patterns emerge. Connections form. You start seeing structure everywhere. It's worth it.

One More Thing

This book is alive. I keep learning. Readers point out mistakes. I find better ways to explain things. New connections become clear.

So the version you're reading now isn't finished. It's a snapshot of current understanding. Check back in six months and parts will be better. That's the nature of genuine learningit never stops.

Let's begin.

Mahdi
2025

Acknowledgments

I would like to express my gratitude to everyone who supported me during the creation of this book. Special thanks to the open-source community for their invaluable resources and to all those who reviewed early drafts and provided feedback.

Introduction

MATHEMATICS IS a language. But unlike English or Persian, it's not something you pick up by immersion. You have to build it, concept by concept, proof by proof, insight by insight.

This book is that construction project.

What We're Actually Doing Here

Think of mathematics as having three layers:

The Surface Layer Formulas, notation, procedures. This is what most textbooks teach. "Here's the quadratic formula. Memorize it. Moving on."

The Structural Layer Why those formulas work. How concepts connect. What patterns repeat across different areas. This is what mathematicians actually think about.

The Foundation Layer What is proof? What is number? What is computation? What can we know and how can we know it? This is where philosophy meets mathematics.

Most books give you the surface, maybe hint at structure, ignore foundation entirely.

We're going to build all three. From the ground up.

How This Works

Each part of this book tackles a major mathematical domain. Not a surveya complete development. We start with motivation (why does this matter?), build formal machinery (what exactly are we talking about?), prove major results (how do we know it's true?), then connect to computation (where does this appear in real systems?).

Some parts are foundationallogic, set theory, algebra. You need these to understand anything else.

Some parts are computationalasymptotic analysis, optimization, numerical methods. These are your tools for designing and analyzing algorithms.

Some parts are appliedmachine learning, cryptography, computer vision. These show how abstract mathematics becomes practical technology.

The parts are ordered to build progressively. But they're also modularif you need quantum computing mathematics right now, jump to that part. Prerequisites are clearly marked. Come back for foundations when you're ready.

Prerequisites? What Prerequisites?

Here's what you actually need: willingness to think hard about abstract ideas.

That's it.

Yes, comfort with algebra helps. Yes, calculus background is useful. Yes, programming experience provides context.

But none of that is required. If you can follow logical arguments and tolerate temporary confusion, you can handle this material.

The limiting factor isn't prior knowledge. It's patience. Mathematical understanding doesn't arrive in sudden flashes of insight. It arrives slowly, through repeated engagement with difficult ideas. You'll read things that don't make sense. You'll work problems that seem impossible. You'll feel stuck.

That's normal. That's the process. Stay with it.

Why Rigor Matters

You might wonder: why prove everything? Why not just show me how to use this stuff?

Because understanding *why* something works changes how you use it. It shows you when it applies and when it doesn't. It reveals connections to other techniques. It lets you adapt methods to new situations.

Hand-waving might feel faster in the moment. But it leaves you helpless when you hit problems slightly different from examples you've seen. Rigor gives you tools to think through genuinely novel situations.

Plusand this might sound strangeproofs are beautiful. There's aesthetic pleasure in seeing how a few simple assumptions lead inevitably to surprising conclusions. Once you develop taste for it, you'll seek out proofs the way you seek out good novels or films.

A Map Is Not The Territory

This introduction can't capture what actually working through this material feels like. It'll be harder than you expect in some places. Easier in others. More interesting than you anticipate in ways you can't predict.

The only way to know what this book contains is to read it. To work through examples. To attempt exercises. To struggle with concepts until they click.

So let's stop talking about mathematics and start doing mathematics.

Turn the page.

"The only way to learn mathematics is to do mathematics." — PAUL HALMOS

Part I

Logic and the Foundations of Reasoning

Before mathematics, there was logic. Before we can reason about numbers, structures, or algorithms, we must understand reasoning itself. This part develops the formal systems of propositional and predicate logic, proof theory, and the philosophical foundations that make mathematics possible.

What Makes This Different:

- ***Philosophical Depth:*** *From Aristotelian syllogisms to modern proof assistants*
- ***Complete Formalization:*** *Natural deduction, sequent calculus, resolution*
- ***Computational Connection:*** *Logic as the foundation of programming languages*
- ***Metamathematical Results:*** *Completeness, soundness, decidability*

> *"Logic is the beginning of wisdom, not the end."*
>
> — Spock (and Aristotle, essentially)

Chapter 1

Propositional Logic and the Calculus of Reasoning

Chapter 2

Predicate Logic and Quantificational Reasoning

Chapter 3

Modal Logic: Necessity, Possibility, and Temporal Reasoning

Chapter 4

Intuitionistic Logic and Constructive Mathematics

Chapter 5

Mathematical Proof: Structure and Technique

Chapter 6

Proof Theory and Natural Deduction

Chapter 7

The Curry-Howard Correspondence: Proofs as Programs

Chapter 8

Automated Theorem Proving and Proof Assistants

Chapter 9

Metalogic: Completeness, Soundness, and Decidability

Chapter 10

Gödel's Incompleteness Theorems: The Limits of Formal Systems

Part II

Set Theory: The Language of Mathematical Objects

SETS ARE the atoms of mathematical discourse. Every mathematical objectnumbers, functions, spaces, categoriesis ultimately constructed from sets. This part develops axiomatic set theory from ZFC, explores the paradoxes that necessitate axiomatization, and examines the philosophical implications of mathematical existence.

What Makes This Different:

- ***Axiomatic Rigor:*** *Full development of Zermelo-Fraenkel set theory*
- ***Philosophical Context:*** *What does mathematical existence mean?*
- ***Computational Relevance:*** *Sets as data structures, type theory*
- ***Foundations of Infinity:*** *Cantor's paradise and its implications*

> *"No one shall expel us from the Paradise that Cantor has created."*
>
> — DAVID HILBERT

Chapter 11

Naive Set Theory and Its Paradoxes

Chapter 12

Axiomatic Set Theory: ZFC

Chapter 13

Relations, Functions, and Mappings

Chapter 14

Cardinality and the Arithmetic of Infinity

Chapter 15

Ordinal Numbers and Transfinite Induction

Chapter 16

The Axiom of Choice and Its Equivalents

Chapter 17

Large Cardinals and the Set-Theoretic Universe

Chapter 18

Constructible Universe and Forcing

Chapter 19

Set Theory and Type Theory

Chapter 20

Alternative Foundations: Category Theory and HoTT

Part III

Algebraic Structures: Symmetry and Abstraction

ALGEBRA IS the study of structure. By abstracting the essential properties of mathematical operationsassociativity, commutativity, identity, inverseswe discover profound unifying patterns. This part develops abstract algebra from groups through categories, revealing how seemingly disparate areas of mathematics share deep structural unity.

What Makes This Different:

- ***Structural Thinking:*** *Focus on morphisms and universal properties*
- ***Computational Algebra:*** *Algorithms for algebraic computation*
- ***Cryptographic Applications:*** *Group theory in RSA, elliptic curves*
- ***Categorical Perspective:*** *Algebra as the study of categories with structure*

> *"Algebra is the offer made by the devil to the mathematician... All you need to do is give me your soul: give up geometry."*
>
> — MICHAEL ATIYAH

Chapter 21

Groups: Symmetry and Transformation

Chapter 22

Rings and Fields: Arithmetic Structures

Chapter 23

Vector Spaces and Linear Algebra

Chapter 24

Modules and Representation Theory

Chapter 25

Boolean Algebra and Logic Circuits

Chapter 26

Lattices and Order Theory

Chapter 27

Universal Algebra and Algebraic Theories

Chapter 28

Category Theory: The Mathematics of Mathematics

Chapter 29

Homological Algebra and Derived Functors

Chapter 30

Algebraic Topology and Fundamental Groups

Part IV

Number Theory: The Integers and Their Mysteries

NUMBER THEORY*once dismissed as the purest of pure mathematicsnow underpins modern cryptography, pseudorandom generation, and algorithmic complexity. This part develops both classical and computational number theory, from Euclid's algorithm to elliptic curves.*

What Makes This Different:

- ***Computational Focus:*** *Complexity analysis of every algorithm*
- ***Cryptographic Applications:*** *RSA, Diffie-Hellman, ECC in depth*
- ***Analytic Methods:*** *Connection to complex analysis and the Riemann hypothesis*
- ***Algorithmic Number Theory:*** *Primality, factorization, discrete logarithm*

> *"Mathematics is the queen of sciences, and number theory is the queen of mathematics."*
>
> — CARL FRIEDRICH GAUSS

Chapter 31

Divisibility and the Fundamental Theorem of Arithmetic

Chapter 32

Modular Arithmetic and Congruences

Chapter 33

The Euclidean Algorithm and Its Complexity

Chapter 34

Chinese Remainder Theorem

Chapter 35

Fermat's Little Theorem and Euler's Theorem

Chapter 36

Primality Testing: Fermat, Miller-Rabin, AKS

Chapter 37

Integer Factorization: Trial Division to Number Field Sieve

Chapter 38

Quadratic Residues and Legendre Symbols

Chapter 39

Continued Fractions and Diophantine Approximation

Chapter 40

Elliptic Curves and Cryptography

Chapter 41

Analytic Number Theory: The Prime Number Theorem

Chapter 42

The Riemann Hypothesis and Zeta Function

Part V

Discrete Mathematics: Combinatorics and Graph Theory

DISCRETE MATHEMATICS *is the native language of computer science. Unlike continuous mathematics, we deal with countable, finite, or denumerable structures: graphs, permutations, recursive sequences. This part develops the combinatorial and graph-theoretic foundations essential for algorithm design and analysis.*

What Makes This Different:

- ***Algorithmic Emphasis:*** *Every result connects to computation*
- ***Generating Functions:*** *Systematic enumeration techniques*
- ***Graph Algorithms:*** *From Euler to modern network science*
- ***Ramsey Theory:*** *The mathematics of inevitable structure*

> *"Combinatorics is an honest subject. No adèles, no sigma-algebras. You count balls in a box, and you either have the right number or you haven't."*
>
> — GIAN-CARLO ROTA

Chapter 43

Fundamental Counting Principles

Chapter 44

Permutations, Combinations, and Binomial Coefficients

Chapter 45

Generating Functions and Recurrence Relations

Chapter 46

The Inclusion-Exclusion Principle

Chapter 47

Pigeonhole Principle and Ramsey Theory

Chapter 48

Graph Theory: Foundations and Representations

Chapter 49

Trees and Spanning Trees

Chapter 50

Connectivity, Paths, and Cycles

Chapter 51

Graph Coloring and Chromatic Numbers

Chapter 52

Planar Graphs and Euler’s Formula

Chapter 53

Network Flows and Matching Theory

Chapter 54

Spectral Graph Theory

Chapter 55

Random Graphs and Probabilistic Methods

Chapter 56

Extremal Combinatorics

Part VI

Mathematical Analysis: Limits, Continuity, and Infinity

Analysis is the mathematics of the infinite and the infinitesimal. From calculus to measure theory, we study limits, continuity, convergencethe machinery needed to reason about algorithms' asymptotic behavior. This part builds rigorous foundations for continuous mathematics and its discrete approximations.

What Makes This Different:

- ***Asymptotic Focus:*** *Everything connects to algorithm analysis*
- ***Rigorous ε-δ:*** *No hand-waving about limits*
- ***Lebesgue Integration:*** *Modern measure-theoretic approach*
- ***Functional Analysis:*** *Infinite-dimensional perspectives*

> *"In mathematics, you don't understand things. You just get used to them."*
>
> — John von Neumann

Chapter 57

Real Numbers: Construction and Completeness

Chapter 58

Sequences and Series

Chapter 59

Limits and Continuity

Chapter 60

Differentiation and Taylor Series

Chapter 61

Integration: Riemann and Lebesgue

Chapter 62

Stirling’s Approximation and Asymptotic Expansions

Chapter 63

Fourier Analysis and Signal Processing

Chapter 64

Complex Analysis and Residue Theory

Chapter 65

Measure Theory and Probability Spaces

Chapter 66

Functional Analysis and Hilbert Spaces

Chapter 67

Operator Theory and Spectral Methods

Chapter 68

Distribution Theory and Weak Derivatives

Part VII

Probability Theory: Randomness and Expectation

PROBABILITY THEORY provides the mathematical framework for reasoning under uncertainty. From randomized algorithms to machine learning, probabilistic thinking permeates modern computation. This part develops probability from measure-theoretic foundations to concentration inequalities and stochastic processes.

What Makes This Different:

- ***Measure-Theoretic Rigor:*** *Probability as a branch of analysis*
- ***Concentration Bounds:*** *Chernoff, Hoeffding, Azuma inequalities*
- ***Randomized Algorithms:*** *Probabilistic method and derandomization*
- ***Stochastic Processes:*** *Markov chains, martingales, Brownian motion*

> *"The theory of probability is at bottom nothing but common sense reduced to calculus."*
>
> — PIERRE-SIMON LAPLACE

Chapter 69

Probability Spaces and Measure Theory

Chapter 70

Random Variables and Distributions

Chapter 71

Expectation and Linearity

Chapter 72

Variance, Covariance, and Correlation

Chapter 73

Markov, Chebyshev, and Chernoff Bounds

Chapter 74

Limit Theorems: Law of Large Numbers and CLT

Chapter 75

Markov Chains and Ergodic Theory

Chapter 76

Martingales and Stopping Times

Chapter 77

Brownian Motion and Stochastic Calculus

Chapter 78

Information Theory and Entropy

Chapter 79

Coding Theory and Error Correction

Chapter 80

Probabilistic Method in Combinatorics

Part VIII

Topology: Continuity and Connectedness

TOPOLOGY STUDIES *properties preserved under continuous deformation. While initially abstract, topological thinking appears throughout computer science: fixed-point theorems in semantics, computational topology in data analysis, homotopy type theory in foundations.*

What Makes This Different:

- ***Computational Topology:*** *Persistent homology and TDA*
- ***Homotopy Type Theory:*** *Topological foundations for CS*
- ***Fixed-Point Theorems:*** *Applications to program semantics*
- ***Metric Spaces:*** *Foundations for analysis and optimization*

> *"Topology is the mathematics of continuity."*
>
> — HENRI POINCARÉ

Chapter 81

Metric Spaces and Topological Spaces

Chapter 82

Continuity and Homeomorphisms

Chapter 83

Compactness and Connectedness

Chapter 84

Separation Axioms and Metrizability

Chapter 85

Fundamental Group and Covering Spaces

Chapter 86

Homology and Cohomology Theory

Chapter 87

Manifolds and Differential Topology

Chapter 88

Knot Theory and Low-Dimensional Topology

Chapter 89

Fixed-Point Theorems and Applications

Chapter 90

Computational Topology and Persistent Homology

Chapter 91

Homotopy Type Theory

Part IX

Formal Languages and Automata Theory

COMPUTATION IS *symbolic manipulation following formal rules. This part develops the theory of formal languages, automata, and computabilitythe mathematical foundations of what computers can and cannot do.*

What Makes This Different:

- ***Philosophical Depth:*** *What is computation?*
- ***Chomsky Hierarchy:*** *The structure of syntactic complexity*
- ***Decidability:*** *The limits of algorithmic solvability*
- ***Complexity Theory:*** *P vs NP and beyond*

> *"Computer science is no more about computers than astronomy is about telescopes."*
>
> — EDSGER DIJKSTRA

Chapter 92

Finite Automata and Regular Languages

Chapter 93

Context-Free Grammars and Pushdown Automata

Chapter 94

Turing Machines and Computability

Chapter 95

The Church-Turing Thesis

Chapter 96

Decidability and the Halting Problem

Chapter 97

Reducibility and Undecidability

Chapter 98

Complexity Theory: Time and Space Classes

Chapter 99

P vs NP: The Millennium Problem

Chapter 100

NP-Completeness and Cook's Theorem

Chapter 101

Approximation Algorithms and Hardness

Chapter 102

Randomized Complexity Classes

Chapter 103

Interactive Proofs and Zero-Knowledge

Part X

Mathematical Logic and Foundations

WHAT ARE *the foundations of mathematics itself? This part examines the deepest questions: Can mathematics be reduced to logic? Are there mathematical truths beyond proof? What is the relationship between syntax and semantics?*

What Makes This Different:

- ***Metamathematics:*** *Mathematics studying itself*
- ***Model Theory:*** *Structures and their theories*
- ***Proof Theory:*** *The mathematics of proofs*
- ***Philosophical Implications:*** *What can we know?*

> *"In mathematics, the art of asking questions is more valuable than solving problems."*
>
> — GEORG CANTOR

Chapter 104

Frege's Logical Foundations of Arithmetic

Chapter 105

Russell’s Paradox and Type Theory

Chapter 106

Principia Mathematica and Logicism

Chapter 107

Hilbert's Program and Formalism

Chapter 108

Gödel's Completeness Theorem

Chapter 109

Gödel's Incompleteness Theorems (Detailed Proof)

Chapter 110

Model Theory and Satisfaction

Chapter 111

Proof Theory and Cut Elimination

Chapter 112

Intuitionistic and Constructive Mathematics

Chapter 113

Reverse Mathematics

Chapter 114

Large Cardinals and Consistency Strength