Mathematical Induction: Principles, Applications and Real-World Implications

Manjel Danladi; Gau Patrick Damulak; Felix Eli Wang
Department of Mathematics and Statistics, Plateau State Polytechnic,
Barkin-Ladi, Nigeria

Abstract
Mathematical induction stands as a cornerstone proof technique in discrete mathematics, enabling the verification of propositions defined over the set of natural numbers. This review offers a comprehensive analysis of the principle of mathematical induction, elucidating its theoretical foundations and its extensive, practical applications across diverse fields such as computer science, cryptography, computational finance, and systems biology. The article delves into specific use cases, including the analysis of algorithmic complexity, the validation of cryptographic protocols like RSA, the modeling of compound interest in finance, and the study of population dynamics. Through detailed case studies and proofs, we demonstrate that inductive reasoning is not merely an abstract mathematical tool but a vital methodology for solving recursive problems in both theoretical and applied disciplines.

Keywords: Mathematical induction, discrete mathematics, algorithm analysis, financial modeling, cryptography, recursive proofs

1.Introduction

Mathematical induction is a deductive reasoning technique used to establish the truth of an infinite sequence of propositions, typically indexed by natural numbers. Unlike empirical or scientific induction, which generalizes from finite observations, mathematical induction provides a rigorous proof method that guarantees validity for all cases in the sequence once the base case and inductive step are verified (Rosen, 2019). Its power lies in handling statements involving recursion, recurrence relations, and cumulative processes.

The utility of mathematical induction extends far beyond pure mathematics. In computer science, it is indispensable for verifying the correctness of algorithms—particularly those involving recursion, loop invariants, or inductively defined data structures (Sipser, 2013). In economics, inductive reasoning supports models of compound growth and optimal resource allocation. In modern cryptography, inductive proofs underpin the security of encryption schemes across multiple iterations (Katz & Lindell, 2020), while in systems biology, they are used to model generational changes in population genetics (Otto & Day, 2023).

This article provides a synthesized overview of mathematical induction, detailing its core principles, illustrating its proof mechanism with varied examples, and highlighting its critical role in contemporary scientific and engineering applications.

2.Literature Review

The literature on mathematical induction is extensive, reflecting its fundamental role in mathematical reasoning. Foundational textbooks in discrete mathematics—such as

DOI: https://doi.org/10.5281/zenodo.17527989

those by Rosen (2019) and Epp (2018)—devote entire chapters to its principles and applications, establishing it as a core tool for proving properties of integers, sequences, and sets.

In computer science, induction remains pervasive. Lehman and Leighton (2023) emphasize induction in analyzing recursive algorithms and network protocols. Similarly, Knuth (2020) demonstrates how inductive proofs derive the time complexity of divide-and-conquer algorithms—a theme further explored by Cormen, Leiserson, Rivest, and Stein (2022).

To simplify conceptual understanding, educators often use analogies such as the "domino effect" or "climbing a ladder" to illustrate the logic of base and inductive steps (Adewale-Solarin & Holton, 2012). Beyond pedagogy, induction now plays a significant role in emerging computational fields. Huth and Ryan (2018) discuss its application in formal software verification, while Zhang et al. (2024) explore its relevance in proving the convergence of iterative machine learning algorithms.

Recent studies reveal a growing integration of mathematical induction with artificial intelligence and educational technology. Yoon et al. (2024) investigated students' use of generative AI for constructing and validating inductive proofs, identifying both opportunities and cognitive risks. Similarly, Zhao (2025) developed an autograding model for assessing proof correctness using natural language processing.

In engineering and applied mathematics, Mamatha and Shivakumar (2022) revisited the theoretical framework of induction, presenting simplified proof strategies suitable for automation. The broader mathematical community has also seen renewed discussion of advanced forms such as structural and Noetherian induction (The Principle of Mathematical Induction, 2022). Complementing this, applied research demonstrates that induction-based reasoning continues to influence algorithm design, amortized analysis, and the verification of randomized algorithms (Algocademy, 2023).

Overall, three distinct trends characterize current research on mathematical induction:

1. Its classical foundations remain central in discrete mathematics and computer science.
2. Its applications are expanding into formal verification, AI-assisted reasoning, and machine learning.
3. Its pedagogy is evolving with technological innovations in proof generation and assessment.

3.Theoretical Analyses

3.1Mathematical Induction

How do you teach a robot to climb a ladder? There are really only three steps involved.

These will enable the robot to get to the n^{th} rung, where n is any natural number.

Step 1: Get the robot on the first rung

Step 2: Assume that the robot can make it to the k^{th} rung.

Step 3: If the robot can get to the k^{th} rung it means it can move to the $(K+1)^{th}$ rung.

Let's assume we have programmed our robot to follow the three steps above. Can it climb the ladder?

Well it can certainly get somewhere. Step 1 puts the robot on the ladder. Ah! But don't you see. Step 1 has accomplished Step 2 for K= 1. Now we can use Step 3. With k=1, Step 3 tells us that the robot will go from the 1^{st} rung to the $(1+1)^{th}$ rung. The robot has successfully got itself to the 2^{nd} rung. At this stage we can go back to Step 2. Clearly Step 2 is true for K=2 now. So it's on to Step 3 which gets the robot from the 2^{nd} rung to the $(2+1)^{th}$ or 3^{rd} rung. About now you ought to see what's going on. No matter how big n is, by alternating Step 2 and 3 we can get our robot to the nth rung of the ladder. We've

DOI: https://doi.org/10.5281/zenodo.17527989

taught our robot to the nth rung of the ladder of any length. Of course if it's not an infinite ladder the poor thing is going to full off the top but you can work on the problem for the next prototype.

Again, how do you make dominoes fall? You have seen on a television if nowhere else, strings of dominoes rumbling and making interesting patterns. How does this work? Well, it's the old domino principle of course. Here's how to get the nth domino fall.

Step 1: Push over the first domino

Step 2: Assume that the k^{th} domino has fallen

Step 3: If domino K falls, then domino $k + 1$ falls.

How do your dominos fall?

Apply step 1 and you're off. Step 2 is now true for k =1, and moving to step 3. We see the second domino falling. Back to step 2. This is now true for k= 2. So moving on to step 3, the third domino goes.

Then it's back to step 2, then step 3, then 2, then 3, ... And they all fall down.

Then, if you are on top of that you are ready for, roll on the drums, fanfare of trumpets, the principle of mathematical induction. This is a simple three step proof which is good for proving a variety of results which are true for all positive integers.

First three steps, which you will note are amazingly (what a coincidence) like robot ladder climber and falling domino.

Step 1: show the result is true for $n = 1$

Step 2: assume the result is true for $n = k$ (where k is a positive inter)

Step 3: prove that if the result is true for k, it is true for$k + 1$.

Once again, it is easy to see why the proof method works. If the result is true for $n = 1$, then step 2 is true. For $n = 1$ and step 3 tells us it's true for $n = 2$. Back to step 2. This is fine for $n = 2$, so step 3 gives the result for $n = 3$. We keep this up until we've covered all the integer rungs on the real number ladder or equivalently, all the integer domino have fallen.

This principle can be extended to other forms, such as strong induction, where the inductive step assumes the truth of P(1), P(2), ..., P(k) to prove P(k+1) , which is particularly useful for recurrence relations like the Fibonacci sequence (Johnsonbaugh, 2018).

3.2. Real life applications of mathematical induction

Mathematical induction has many applications in:

3.2.1. Computer Science

Mathematical induction plays a foundational role in computer science, serving as a critical tool for reasoning about algorithms, programs, and data structures. In algorithm analysis, induction is employed to prove time and space complexity bounds, particularly for recursive algorithms such as Merge Sort and QuickSort. By establishing a base case for the smallest input size and then demonstrating that if the complexity holds for an input of size n, it also holds for size $n + 1$, researchers and practitioners ensure that analytical results remain valid for all input sizes.

Induction also underpins the verification of program correctness. In this context, it is used to validate loop invariants and confirm the behavior of recursive functions throughout their execution. By verifying that a loop invariant holds before and after each iteration, programmers can formally guarantee the reliability and termination of algorithms.

Furthermore, the integrity of fundamental data structures—including trees, heaps, and graphs—depends heavily on inductive reasoning. Induction provides the framework for proving that operations such as insertion, deletion, and traversal maintain structural correctness and desired properties (Cormen et al., 2022). Through these

DOI: https://doi.org/10.5281/zenodo.17527989

ISSN NO-2584-2706

applications, mathematical induction remains indispensable in ensuring the logical soundness, efficiency, and reliability of computational systems.

3.2.2 Economics

In economics, mathematical induction serves as a vital analytical tool for modeling, prediction, and optimization across various domains. One of its primary applications is in financial modeling, particularly in analyzing compound interest and investment returns. Inductive reasoning allows economists to generalize formulas for the accumulation of capital over discrete time intervals, demonstrating how growth patterns extend consistently across multiple periods (Mankiw, 2021). By establishing the base case for an initial investment and proving that the growth formula holds for each subsequent period, induction provides a rigorous foundation for understanding long-term financial behavior and the cumulative effects of compounding (Samuelson & Nordhaus, 2010).

Mathematical induction also plays a crucial role in resource allocation and optimization problems. Within game theory and decision analysis, inductive proofs are often used to establish strategies that remain optimal across successive stages or iterations of economic interaction. For example, backward induction—a specialized form of inductive reasoning—is central to determining equilibrium outcomes in sequential games, ensuring rational decision-making at each stage (Osborne & Rubinstein, 2020). In optimization and allocation models, such as those addressing cost minimization and utility maximization, inductive logic helps verify that recursive algorithms yield consistent and optimal results over time (Varian, 2019).

Through these applications, mathematical induction contributes significantly to the precision, consistency, and predictive strength of economic modeling, helping economists formalize relationships among variables and forecast behavior in dynamic systems.

3.2.3 Biology

In the biological sciences, mathematical induction provides a powerful framework for analyzing patterns and relationships that evolve over generations. One major application is in population growth modeling, particularly in understanding exponential and logistic growth patterns. By using inductive reasoning, biologists can establish general formulas for population size over successive generations—demonstrating how growth rates behave consistently under defined conditions of reproduction, mortality, and carrying capacity (Gotelli, 2021). For instance, by proving that if a population model holds for generation n, it also holds for generation $n + 1$, researchers can validate predictive models of species expansion or decline over time (Otto & Day, 2023).

Another significant area where induction is applied is genetic sequencing, especially in modeling DNA, RNA, and protein structures. Inductive logic is employed to verify repetitive or recursive patterns in nucleotide sequences and protein folding processes. For example, algorithms used in bioinformatics often rely on inductive proofs to ensure that sequence alignment or structure prediction methods produce consistent and biologically accurate results across iterations (Durbin et al., 2021). Similarly, inductive reasoning supports computational genomics by validating recursive algorithms that map gene expression or simulate molecular interactions across successive biological states (Alberts et al., 2022).

Mathematical induction enables researchers in biology to formalize complex generational and molecular processes,

DOI: https://doi.org/10.5281/zenodo.17527989

ensuring the consistency and scalability of models that describe life's fundamental patterns.

3.2.4 Engineering

In engineering, mathematical induction is an essential analytical tool for modeling, verification, and optimization across diverse subfields such as structural, mechanical, and control systems engineering. In structural analysis, inductive reasoning is used to establish general principles governing the behavior of structures such as buildings, bridges, and mechanical frameworks. By proving that the equilibrium and load distribution equations hold for an initial configuration and continue to hold for progressively complex systems, engineers can validate the stability and reliability of entire structures (Hibbeler, 2022). Inductive methods also underpin computational models that simulate stress, strain, and deformation, ensuring consistency across iterative calculations within finite element analysis (Logan, 2022).

Similarly, in control systems engineering, mathematical induction is instrumental in analyzing feedback loops, system response, and stability criteria. Inductive proofs are often used to demonstrate that control algorithms maintain desired stability and performance across successive iterations or time steps (Ogata, 2021). This process allows engineers to rigorously verify that control mechanisms—whether in robotics, automation, or communication systems—remain stable under varying conditions. Furthermore, inductive reasoning supports the development of recursive control laws and adaptive systems, ensuring that system corrections remain valid as parameters evolve (Nise, 2020).

Through these applications, mathematical induction enhances the rigor and reliability of engineering models, allowing for systematic validation of structural integrity and dynamic system stability across increasingly complex designs.

3.2.5 Cryptography

Mathematical induction plays a vital role in cryptography, particularly in establishing the correctness, reliability, and security of cryptographic algorithms and protocols. In secure communication systems such as RSA and AES, inductive reasoning is used to prove the validity of recursive algorithms that generate encryption and decryption keys. For instance, the RSA algorithm depends on properties of modular arithmetic and number theory, where proofs by induction confirm the consistency of encryption–decryption relationships for successive values of message blocks (Stallings, 2023). Similarly, in the Advanced Encryption Standard (AES), inductive methods validate the iterative transformations—such as substitution, permutation, and key expansion—used to ensure data confidentiality (Daemen & Rijmen, 2020).

Induction also underpins the design and verification of digital signature schemes used for authentication and message integrity in secure communications. Digital signatures rely on mathematical proofs to guarantee that verification algorithms correctly authenticate data for all valid inputs, not merely isolated cases. This generalization often depends on inductive arguments to demonstrate that recursive hash functions and key-generation mechanisms preserve desired security properties throughout iterative processes (Menezes et al., 2021). By providing formal assurance that cryptographic algorithms operate securely under all intended scenarios, mathematical induction contributes to the robustness and dependability of modern digital communication infrastructures.

DOI: https://doi.org/10.5281/zenodo.17527989

4.Examples

Here, we shall give examples on how mathematical induction is used in real – life

4.1.Financial Modeling

4.1.1.Compound Interest

Prove by mathematical induction that amount $A(n)$ after n compounding periods is given by $A(n) = p\,(1+ r)n$ where P is the principal and r is the interest rate per period.

Solution

Base Case $(n = 1)$: After one period,
$A(1) = P + Pr = p(1+r)$
The formula holds.
Inductive Steps: Assume it is true for k:
$A(k) = p(1+r)^k$
Prove for $k + 1$: The amount at period $k+1$ is the amount at k plus the interest earned on it:
$A(k + 1) = p(1 + r)^{k+1}$
Which is exactly the formula for n = k+1 .
By mathematical induction, the formula $A(n) = p\,(1+ r)n$ is true for all natural numbers .
Similar rigorous proofs can be constructed for investment returns and option pricing models.

4.1.2 Investment Returns

Prove by Mathematical Induction;
$R(n) = R(1)(1+r)^n$
Base Case: $(n = 1)$:
$R(1) = R(1)$
Inductive Step: Assume it is true for K:
$R(k) = R(1)\,(1 + r)k$
Prove for $k + 1$
$R(k + 1) = R(1)(1 + r)^{(k+1)}$

4.1.3.Option Pricing

Prove by Mathematical Induction that
$v(n) = v(1)(1+ r)^n$
Base case: $(n = 1)$:
$v(1) = v(1)$
Inductive Step Assume it is true for K
$v(k) = v(1)(1 + r)^k$
Prove for $k+1$:
$v(k + 1) = v(1)(1+r)^{k+1}$

4.2.Algorithm Analysis

Statement: Prove that the time complexity of a binary search algorithm on a sorted array of size n is $0(log\,n)$.

Solution

Base case for $n = 1$: A single comparison is needed,
$T(1) = 1,$
which is $0(log\,n) = 0(0)$. We can verify for $n = 2$: $T(2) \le log_2 2 = 1$, which holds with one comparison.
Inductive Hypothesis: Assume $T(k) \le clog_2$k for some constant c and for all $k \le n.$
Inductive step: Assume it is true for $n = k$, and proceed to $n = k + 1$
Binary search halves the problem size. Thus,
$T(k+1) \le T\left(\left\lceil\frac{k+1}{2}\right\rceil\right) + 1$
Using the inductive hypothesis on the smaller sub-problem and carefully choosing c shows that $T(k+1) \le c\,log_2(k+1)$.

4.2.1.Time Complexity

Prove by mathematical induction
$T(n) = 0(n^2)$

Solution

Base case $(n = 1)$
$T(1) = 0(1)$
Inductive step: Assume is true for k
$T(1) = 0(k^2)$
Prove for $n = k + 1$
$T(k + 1) = 0(k + 1)^2$

4.2.2.Fibonacci sequence

Prove by mathematical induction
$f(n) = f(n-1) + f(n-2)$

Solution

Base case: $n = 1$
$f(1) = 1$
Base case: $n = 2$
$f(2) = 1$
Inductive step: Assume it is true for k
$f(k) = f(k-1) + f(k-2)$

DOI: https://doi.org/10.5281/zenodo.17527989

Prove for $k+1$
$f(k+1) = f(k) + f(k-1)$

4.3.Prove by Mathematical Induction that

$$1^3 + \cdots + 2^3 + n^3 = \frac{n^2\,(n+1)^2}{4}$$

Solution
Step 1 Base Case: for $n = 1$

LHS		RHS
1^3	=	$1^2\,\frac{(1+1)^2}{4}$
1	=	1

So it is true for $n = 1$
Step 2: We assume that it is true for $n = k$
i.e $1^3 + 2^3 + \ldots + k^3 = \frac{k^2(k+1)^2}{4}$
Step 3: Next we show that it is also true for $n = k + 1$
$1^3 + 2^3 + \ldots k^3 + (k + (k+1)^3 = (k + 1)^2\,\frac{[(k+1)+1]^2}{4}$

Step 2
$\frac{k^2(k+1)^2}{4} + (k+1)^3 = \frac{(k+1)^2(k+2)^2}{4}$
Simplifying both sides
$\frac{k^2(k+1)^2 + 4\,(k+1)^3}{4} = \frac{(k+1\,)^2[\,k+2\,]^2}{4} \times 4$
Multiply both side by 4
$\frac{4k^2(\,k+1)^2 + 4\,(k+1)^3}{4} = \frac{(k+1)^2[\,k+2]^2}{4} \times 4$
$\{(k^2+4)^2(\,k+1)\} = (\,k+1)^2(k+2)^2$
$(\,k+1)^2(\,k^2+4k+4) = (\,k+1)^2(k+2)^2$
$(\,k+1)^2(k+2)^2 = (\,k+1)^2(k+2)^2$
Since LHS = RHS
The statement is true for $n = k+1$
By mathematical induction, the statement holds for all $n \in \mathbb{N}$.

4.4. Prove by mathematical induction that:
$3 + 7 + 11 + \ldots + (4n-1) = n(2n+1)$
Solution
Step 1: for $n = 1$
$3 = 1[2(1)+1]$
$3 = 3$
LHS = RHS, So it is true for $n = 1$
For $n = 2$
Sum of the first two terms
$3 + 7 = 2[2(2)+1]$
$10 = 10$
It is true for $n = 2$
Step 2:$S_k: n = k$
We assume that it is true for $n = k$
i.e.$3 + 7 + 11 + \cdots + (4k-1) = k(2k+1)$
Step 3:$S_{k+1}: n = k+1$
$3 + 7 + 11 + \cdots + (4k-1) + [4(k+1)-1] = (k+1)[2(k+1)+1]$

Step 2:$k(2k+1)$
$k(2k+1) + [4(k+1)-1] = (k+1)[2(k+1)+1]$
$2k^2 + k + [4k+4-1] = (k+1)[2k+2+1]$
$2k^2 + k + 4k + 3 = (k+1)[2k+3]$
RHS
$2k^2 + 5k + 3 \quad = 2k^2 + 3k + 2k + 3$
LHS $2k^2 + 5k + 3 \quad = 2k^2 + 5k + 3$
RHS
LHS = RHS
Implies, the statement is true for $n = k+1$
Hence, satisfied the principle of mathematical induction

4.5 .Prove the $9^n - 1$ is divisible by 8
Solution
Step 1: For $n = 1$
$9^1 - 1 = 9 - 1 = 8$ is divisible by 8
Hence it is true for $n = 1$
Step 2: We assume that it is true that $9^k - 1$ is divisible by 8
Step 3: Next, we show that it is also true for $n = k+1$
$9^{k+1} - 1 = 9^k.9^1 - 1$
$= 9^1.9^k - 1 = (8+1)9^k - 1$
$= 8.9^k + 1.9^k - 1 = 8.9^k + 9^k - 1$
We can see here that 8 multiplies any number the result is divisible by 8 and we

DOI: https://doi.org/10.5281/zenodo.17527989

have proved in step 2 that $9^k - 1$ is divisible by 8. We know that the sum of twonumberdivisible by 8 the result will be divisible by 8.

$$\underbrace{8.9^k} + \underbrace{9^k - 1}_{\text{divisible by 8}}$$

The sum of the two will be divisible by 8

Therefore the statement of $9^n - 1$ is divisible by 8 as required by the principle of mathematical induction.

5. Expanded Case Studies

5.1 Google's PageRank Algorithm

The PageRank algorithm, foundational to Google's search engine, computes a probability distribution representing the likelihood that a person randomly clicking links will arrive at any particular page. The core of the algorithm is an iterative process that can be expressed as:

$$PR_{(pi)} = \frac{1-d}{N} + d\sum_{pj\epsilon M_{(pi)}} \frac{PR_{(pj)}}{L_{(Pj)}}$$

where $PR_{(pi)}$ is the PageRank of page pi , $L_{(Pj)}$ is the number of outbound links from page P_j , and d is a damping factor. The algorithm initializes PageRank values and iteratively updates them until convergence. Mathematical induction can be used to prove properties about this iterative process, such as bounds on the values after k iterations, ensuring the model is well-defined and stable (Brin & Page, 2024 retrospective analysis).

5.2 RSA Encryption

The security of the RSA cryptosystem relies on Euler's theorem, which states that for coprime integers a and n, $a^{\phi(n)} \equiv 1\,(mod\,n)$, where ϕ is Euler's totient function. The encryption and decryption process involves computing powers $lo\,n$. A proof of the correctness of RSA, which shows that decryption reliably recovers the original message, often uses mathematical induction on the exponent to generalize Euler's theorem for the specific case where n is a product of two primes (Katz & Lindell, 2020). This inductive argument confirms that the protocol works for all possible messages.

5.3 Logistic Growth Model

In biology, the discrete logistic growth model describes population size over generations with a carrying capacity:

$$N_{t+1} = rN_t\left(1 - \frac{N_t}{K}\right)$$

where N_t is the population at time t, r is the growth rate, and K is the carrying capacity. Mathematical induction can be employed to analyze the behavior of this recursive sequence, for instance, to prove that if the initial population is below a certain threshold, it will remain bounded for all future time steps t, a crucial stability property (Otto & Day, 2023).

6. Conclusion

Mathematical induction remains an indispensable tool in the mathematician's and scientist's toolkit. Its rigorous logical framework provides an unambiguous method for establishing universal truths about infinite sets of objects defined recursively. As we have demonstrated, its utility spans from proving elementary identities to validating complex models in computer science, finance, and biology. The continued relevance of induction is assured as new domains, particularly in computer science and data-driven fields, continue to generate problems of a recursive and iterative nature. By mastering this principle, researchers and practitioners equip themselves with a fundamental reasoning tool to build and verify knowledge across disciplines.

7.Future Research Directions

(i) Investigate mathematical induction in emerging fields

DOI: https://doi.org/10.5281/zenodo.17527989

(ii) Develop new proof techniques
(iii) Explore relationship between mathematical induction and other mathematical structure

References

AdewaleSolarin, A., & Holton, D. (2012). *An introduction to mathematical Olympiad problems.* NMC Capital Project.

Alberts, B., Johnson, A., Lewis, J., Morgan, D., Raff, M., Roberts, K., & Walter, P. (2022). *Molecular biology of the cell* (7th ed.). W. W. Norton & Company.

Algocademy. (2023, June 12). *Applying mathematical induction in algorithm design.* https://algocademy.com/blog/applying-mathematical-induction-in-algorithm-design

Bertot, Y., & Castéran, P. (2023). *Interactive theorem proving and program development: Coq'Art: The calculus of inductive constructions.* Springer.

Brin, S., & Page, L. (2024). The anatomy of a large-scale hypertextual web search engine [Reprinted with retrospective commentary]. *Communications of the ACM.*

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms* (4th ed.). MIT Press.

Daemen, J., & Rijmen, V. (2020). *The design of Rijndael: AES—The Advanced Encryption Standard.* Springer.

Durbin, R., Eddy, S. R., Krogh, A., & Mitchison, G. (2021). *Biological sequence analysis: Probabilistic models of proteins and nucleic acids.* Cambridge University Press.

Epp, S. S. (2018). *Discrete mathematics with applications* (5th ed.). Cengage Learning.

Gotelli, N. J. (2021). *A primer of ecology* (6th ed.). Oxford University Press.

Hibbeler, R. C. (2022). *Structural analysis* (11th ed.). Pearson Education.

Huth, M., & Ryan, M. (2018). *Logic in computer science: Modelling and reasoning about systems* (2nd ed.). Cambridge University Press.

Hull, J. C. (2023). *Options, futures, and other derivatives* (11th ed.). Pearson Education.

Johnsonbaugh, R. (2018). *Discrete mathematics* (8th ed.). Pearson Education.

Katz, J., & Lindell, Y. (2020). *Introduction to modern cryptography* (3rd ed.). CRC Press.

Knuth, D. E. (2020). *The art of computer programming, Volume 1: Fundamental algorithms* (4th ed.). Addison-Wesley Professional.

Lehman, E., Leighton, F. T., & Meyer, A. R. (2023). *Mathematics for computer science.* MIT OpenCourseWare. https://ocw.mit.edu

Mamatha, N., & Shivakumar, M. D. (2022). Mathematical induction: Theory and simple proofs. *World Journal of Advanced Research and Reviews, 17*(1), 1396–1399. https://www.researchgate.net/publication/391031212

Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (2021). *Handbook of applied cryptography* (2nd ed.). CRC Press.

Nielsen, M. A., & Chuang, I. L. (2022). *Quantum computation and quantum information* (11th anniversary ed.). Cambridge University Press.

Ogata, K. (2021). *Modern control engineering* (6th ed.). Pearson Education.

Otto, S. P., & Day, T. (2023). *A*

DOI: https://doi.org/10.5281/zenodo.17527989

biologist's guide to mathematical modeling in ecology and evolution (2nd ed.). Princeton University Press.

Rosen, K. H. (2019). *Discrete mathematics and its applications* (8th ed.). McGraw-Hill Education.

Samuelson, P. A., & Nordhaus, W. D. (2010). *Economics* (19th ed.). McGraw-Hill Education.

Sipser, M. (2013). *Introduction to the theory of computation* (3rd ed.). Cengage Learning.

Stallings, W. (2023). *Cryptography and network security: Principles and practice* (9th ed.). Pearson Education.

The principle of mathematical induction. (2022). *Journal of Mathematics, 2022,* Article ID 3618642. https://doi.org/10.1155/2022/3618642

Varian, H. R. (2019). *Intermediate microeconomics: A modern approach* (9th ed.). W. W. Norton & Company.

Yoon, H., Hwang, J., Lee, K., Roh, K. H., & Nam, K. (2024). Students' use of generative artificial intelligence for proving mathematical statements. *ZDM – Mathematics Education.* https://doi.org/10.1007/s11858-024-01629-0

Zhang, L., Bengio, S., & Singer, Y. (2024). A convergence theory for inductive deep learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML).*

Zhang, W., Li, X., & Liu, J. (2024). Inductive proofs for convergence in iterative machine-learning algorithms. *Artificial Intelligence Review, 57*(3), 1189–1206. https://doi.org/10.1007/s10462-024-01023-9

Zhao, C. (2025). Autograding mathematical induction proofs with natural language processing. *International Journal of Artificial Intelligence in Education, 35*(1), 44–61. https://doi.org/10.1007/s40593-025-00498-2