Computer Assisted Proofs

Introduction

Mathematics is often seen as a way to describe the universe. At its core, mathematics uses proofs to show that ideas and theorems are true. So these ideas can be further iterated upon. Over time, the way we create these proofs has changed a lot. This paper looks into computer-assisted proofs, exploring how they started, why they're important, and the debates about whether they're as valid as traditional proofs.

Background on Mathematical Proofs

Mathematical proofs are important because they show that a statement or theorem is **always** true. Historically, mathematicians create proofs by carefully reasoning through each step, starting from basic already proven principles and building to their final conclusion. This process helps show that every part of the proof is logically sound.

However, proofs do more than just confirm that something is true; they help others understand why it's true. By working through proofs, mathematicians can explore new ideas and deepen their knowledge of math. Many consider the beauty of traditional proofs to lie in their clarity and the fact that they reveal the underlying structure of the math they consist of.

Emergence of Computer-Assisted Proofs

As computers grew more popular though the 1900s mathematicians began using them to help with proofs. Computer-assisted proofs use algorithms to handle complex calculation and check many different cases that would be too time-consuming for any human to do by hand. The combination of the human mind and machine efficiency allows mathematicians to tackle problems that were previously out of reach due their size and complexity.

The idea of using computers to help with proofs started around the mid-20th century as computers became more powerful and efficient. At first computers where only used for simple calculations, but as technology advanced, so did the methods for utilizing computers in proofs. Now, there are specialized programs and proof assistants that can verify proofs and also help construct them with very minimal human input.

Computer-assisted proofs have led to some significant leaps in mathematics, solving many long-standing problems. These achievements show how powerful computational tools can be in advancing mathematical knowledge, challenging the traditional ways of creating proofs.

Understanding Computer-Assisted Proofs

A computer-assisted proof is a mathematical proof that relies on a computer program to verify certain sections of the argument. Where as traditional proofs are done entirely by hand using logical reasoning. Computer-assisted proofs use software to handle many complex calculations or check numerous cases that would be too time-consuming for humans to manage by themselves.

There are two main types of computer-assisted proofs:

1. **Fully Automated Proofs**: In fully automated proofs, the computer will attempt to prove the theorem with very little help. The mathematician simply sets up the problem, and the computer uses algorithms to find the proof without any human intervention. This can be very useful when problems that can be broken down into repetitive steps.

2. **Semi-Automated Proofs**: These proofs use a collaborative approach between humans and computers. While the computer handles specific complex calculations or verifies certain parts of the proof, mathematicians still guide the overall reasoning. This method is often used when parts of the proof need creative insights that computers are not capable of providing.

Creating computer-assisted proofs involves various methodologies and tools designed to work together seamlessly. Some of the most commonly used tools include:

- **Automated Theorem Provers**: These tools attempt to prove theorems automatically without much human input. They use algorithms to explore possible proof paths and find valid arguments. Examples include:
    - **Prover9**: An automated theorem prover for first-order and equational logic.
    - **E-Prover**: A theorem prover for first-order logic with equality.
- **Programming Languages**: Certain programming languages are designed to work closely with proof assistants and theorem provers. They help in writing the formal definitions and proofs that these tools can understand and verify. Examples include:
    - **ML**: Often used in proof assistants like HOL.
    - **Gallina**: The language used by Coq for writing proofs and definitions.

Comparison with Traditional Proofs

Computer-assisted proofs offer some advantages over traditional proofs. One of the main benefits is their ability to handle very complex and large-scale problems that would be impractical to solve by hand. Computers can quickly perform calculations and check almost limitless cases, which enhances the precision and reliability of the proof.

However, there are also some differences in approach between computer-assisted and traditional proofs. Traditional proofs often rely on human intuition and creativity to find elegant and insightful arguments. Computer-assisted proofs may be more mechanical, focusing on exhaustive verification rather than creative reasoning.

Computer assistance is beneficial in situations where the proof requires checking a large or complex number of cases. For example, in the proof of the Four Color Theorem, a computer was used to check many different configurations of maps to ensure that four colors were sufficient. Without computer assistance, such a proof would have been nearly impossible to complete by hand.

Famous Examples of Computer-Assisted Proofs

The Four Color Theorem   Overview

The Four Color Theorem states that any planar map can be colored using at most four colors such that no two adjacent regions share the same color. In this case, adjacent refers to two regions sharing a boundary segment, not just a point.

This theorem was conjectured in 1852 by Francis Guthrie while trying to color a map of England. The theorem applies to abstract planar graphs and is a foundational result in graph theory.

Role of Computer Assistance in the Proof

Due to the difficulty of the Four Color Theorem to prove, many false starts and partial results came as a result. This theorem was proved in 1976 and was a one of a kind proof as it relied on computer assistance.

The key contributors in this proof were Kenneth Appel and Wolfgang Haken. whom were mathematicians at University of Illinois. They reduced the problem to only needing to check 1,936 "reducible configurations," cases.

The role of computers in this proof came when Appel and Haken needed to verify the reducibility of all 1,936 cases. This marked one of the first significant uses of computers to solve a mathematical theorem, sparking debates about the validity and nature of proofs. Due to the heavy reliance on computers, this raised concerns about trustworthiness since no human could feasibly verify all the computational steps by hand.

The Kepler Conjecture   Overview

The Kepler Conjecture assets that the most efficient way to pack identical spheres in three-dimensional space is in a face-centered cubic (FCC) or hexagonal close-packing (HCP) arrangement. These arrangements fill approximately 74.05% of the volume of space.

This conjecture was proposed by Johannes Kepler in 1611. It remained unproven for nearly four centuries and is a important result in the study of sphere packing.

Role of Computer Assistance in the Proof

In 1998, the proof was done by Thomas Hales and his graduate student Samuel Ferguson. The proof combined traditional mathematical reasoning alongside computer-assisted verification.

Hales and Ferguson were able to reduce the problem to an enormously large, but finite, number of specific cases by analyzing local sphere configurations. Additionally, they computed and verified complex inequalities related to the density of sphere arrangements. Manual verification is the traditional way of verification but it was infeasible in this case, so many software tools and algorithms were used for computations and verification.

The proof of the Kepler Conjecture consisted of over 250 pages of mathematical arguments and 3 gigabytes of computer-generated data. This makes this proof one of the most complex proofs ever done.

As a result of this proof, skepticism followed. The proof was submitted to the journal *Annals of Mathematics* in 1998, where it underwent rigorous scrutiny. It was not until 2005 that the journal accepted the proof, however, they noted that verifying the computation components were not feasible by humans.

To help verify every aspect of this proof, Hales launched the Flyspeck Project to formally prove this conjecture using automated theorem-proving systems. By 2014, Flyspeck completed the formal verification process. This achievement confirmed the correctness of Hales' proof.

Results of the Kepler Conjecture Proof

- The proof became a benchmark for computer-assisted proofs for longstanding mathematical problems.
- Many advancements were made in areas like automated theorem proving, discrete geometry and optimization, and mathematical philosophy.

The Classification of Finite Simple Groups (CFSG) Theorem   Overview

The Classification of Finite Simple Groups is one of the most monumental achievements in modern mathematics. It states: Every finite simple group belongs to one of the following families: 1. Cyclic groups of prime order. 2. Alternating groups A_n for $n \geq 5$, the group of even permutations on n elements. 3. Groups of Lie type 4. 26 sporadic groups (no correlation to any of the first three groups)

A finite simple group is a group that is finite, nontrivial, and has no normal subgroups other than the trivial group and itself. These are the building blocks of all finite groups, much like prime numbers are the building blocks of integers.

Role of Computer Assistance in Proof

The proof of the Classification of Finite Simple Groups spanned over 50 years and involved contributions from hundreds of mathematicians. The problem with proving the CFSG are the exhaustive case analyses which proved to be the most complex parts.

Many aspects of the proof required detailed examination of specific configurations, properties of subgroups, and representations. Computers were used to handle these cases efficiently. Algorithms implemented in computation algebra systems, like GAP and MAGMA, to verify properties of specific groups and their representations.

With the first three groups out of the way, studying sporadic groups relied heavily on computational techniques to explore subgroup structures and properties. Alongside sporadic groups, certain subgroup lattice configurations required computer assistance.

Significance in Algebra

The CFSG has many results across mathematics:

1. Structure and Symmetry
2. Applications in fields such as Algebraic Geometry, Number Theory, and Mathematical Physics.
3. Theory of Representations
4. Unification of Mathematics

D. The Boolean Pythagorean Triples Problem - Description of the problem. - Use of SAT solvers in the proof by Marijn Heule, Oliver Kullmann, and Victor Marek. - Discussion of the proof's size and verification.

E. Other Notable Examples - Brief mentions of additional theorems or conjectures utilizing computer assistance (e.g., Robertson–Seymour theorem, Ligocki's work on knot theory).

IV. Are Computer-Assisted Proofs Really "Proofs"?

- **A. Traditional Notions of Proof in Mathematics**
  - Definitions and expectations of mathematical proofs.
  - The role of human intuition and creativity.
- **B. Philosophical Perspectives**
  - Debates on the epistemological status of computer-assisted proofs.
  - Views from mathematicians and philosophers.
- **C. Reliability and Verification**
  - Concerns about software bugs and hardware errors.
  - Efforts to ensure correctness (e.g., independent verifications, formal proofs).
- **D. Acceptance within the Mathematical Community**
  - Historical skepticism and gradual acceptance.
  - Current consensus and differing opinions.

V. Advantages and Limitations of Computer-Assisted Proofs

- **A. Advantages**
  - Ability to handle complex and large-scale problems.
  - Enhancing precision and reducing human error.
  - Facilitating exploration of new mathematical territories.
- **B. Limitations**
  - Dependence on technology and potential obsolescence.
  - Challenges in understanding and interpreting proofs.
  - Accessibility issues for mathematicians without programming expertise.
- **C. Balancing Human and Machine Collaboration**
  - Integrating computational tools with traditional mathematical methods.
  - Future prospects for synergy between mathematicians and technology.

VI. The Future of Computer-Assisted Proofs

- **A. Technological Advancements**

- – Improvements in computing power and algorithms.
    - – Emerging tools and platforms for proof verification.
- **B. Integration with Artificial Intelligence**
    - – Potential for AI to contribute to theorem discovery and proof construction.
    - – Ethical and practical considerations.
- **C. Evolving Standards and Practices**
    - – Developing best practices for creating and validating computer-assisted proofs.
    - – Educational implications for training future mathematicians.

VII. Conclusion

- **A. Summary of Key Points**
    - – Recap of the role and examples of computer-assisted proofs.
    - – Overview of the debate on their validity as proofs.
- **B. Reflection on Their Impact**
    - – Influence on the advancement of mathematics.
    - – Shifts in the landscape of mathematical research and proof strategies.
- **C. Final Thoughts**
    - – The evolving nature of mathematical proof.
    - – The ongoing dialogue between tradition and innovation in mathematics.