Dense Sphere Packings

A blueprint for formal proofs

THOMAS C. HALES

University of Pittsburgh
# Contents

*Preface*  

**PART ONE OVERVIEW**  

1 Close Packing  

1.1 History 3  

1.2 Face-Centered Cubic 6  

1.3 Hexagonal-Close Packing 10  

1.4 Gauss 13  

1.5 Thue 14  

1.6 Dense Packings in a Nutshell 17  

**PART TWO FOUNDATIONS**  

2 Trigonometry  

2.1 Background Knowledge 25  

2.2 Trig Identities 27  

2.3 Vector Geometry 33  

2.4 Angle 39  

2.5 Coordinates 51  

2.6 Cycle 55  

2.7 Chapter Summary 60  

3 Volume  

3.1 Background in Measure 61  

3.2 Primitive Volume 63  

3.3 Finiteness and Volume 70
## Contents

4 **Hypermap**  
4.1 Background on Permutations 73  
4.2 Definitions 74  
4.3 Walkup 78  
4.4 Planarity 83  
4.5 Path 86  
4.6 Subquotient 91  
4.7 Generation 95

5 **Fan** 112  
5.1 Definitions 112  
5.2 Topology 117  
5.3 Planarity 121  
5.4 Polyhedron 132

**PART THREE**  
**THE KEPLER CONJECTURE** 143

6 **Packing** 145  
6.1 The Primitive State of Our Subject Revealed 145  
6.2 Rogers Simplex 150  
6.3 Cells 167  
6.4 Clusters 180  
6.5 Counting Spheres 185

7 **Local Fan** 194  
7.1 Localization 194  
7.2 Modification 203  
7.3 Polarity 209  
7.4 Main Estimate 212

8 **Tame Hypermap** 235  
8.1 Definition 236  
8.2 Contravening Hypermap 238  
8.3 Contravention is Tame 243  
8.4 Admissibility 248  
8.5 Linear Programs 250  
8.6 Strong Dodecahedral Theorem 252

**Appendix A**  
**Credits** 257  
References 261  
General index 264  
Notation index 269
Preface

“I think there’s a revolution in mathematics around the corner. I think that . . . people will look back on the fin-de-siècle of the twentieth century and say ‘Then is when it happened’ (just like we look back at the Greeks for inventing the concept of proof and at the nineteenth century for making analysis rigorous). I really believe that. And it amazes me that no one seems to notice.

“Never before have the Platonic mathematical world and the physical world been this similar, this close. Is it strange that I expect leakage between these two worlds? That I think the proof strings will find their way to the computer memories? . . .

“What I expect is that some kind of computer system will be created, a proof checker, that all mathematicians will start using to check their work, their proofs, their mathematics. I have no idea what shape such a system will take. But I expect some system to come into being that is past some threshold so that it is practical enough for real work, and then quite suddenly some kind of ‘phase transition’ will occur and everyone will be using that system.”

–Freek Wiedijk [49]
Alecios: Christos has a problem with the ‘foundational quest’!

Christos: Wrong! I have two problems with your version of it! One, it didn’t fail and, two, it wasn’t a tragedy! Granted, there are some tragic parts! But the ending is happy, as in the ‘Oresteia’!


Christos: ‘The meaning is in the ending!’ you said so yourself! So, follow the quest for ten more years and you get a brand-new triumphant finale with the creation of the computer, which is the quest’s real hero! Your problem is, simply, that you see it as a story of people!

Apostolos: Well, stories do tend to be about people!

Christos: So, choose the right people! And show what they really did! All we we learn of the great von Neumann is he said ‘It’s over’ when he heard Gödel!

Alecios: But it was over in a sense, wasn’t it? Pop went Hilbert’s ‘no ignorabimus’!

Christos: But then came the quest’s jeune premier, its parsifal . . . Alan Turing! He said ‘Ok, we can’t prove everything! So, let’s see what we can prove!’ and to define proof, he invented, in 1936, a theoretical machine which contains all the ideas of the computer! . . . which, after the war, he and von Neumann, the quest’s proudest sons, brought to full life!

—Doxiadis and Papadimitriou, Logicomix [10]
Preface

“Despite the unusual nature of the proof, the editors of the Annals of Mathematics agreed to publish it, provided it was accepted by a panel of twelve referees. In 2003, after four years of work, the head of the referee’s panel Gábor Fejes Tóth (son of László Fejes Tóth) reported that the panel were ‘99% certain’ of the correctness of the proof.”

– Wikipedia entry on the Kepler conjecture

“Sometimes fixing a 1 percent defect takes 500 percent effort.”

– Joel Spolsky, Joel on Software [42]

“Every one fully persuaded is a fool.”

– Barthasar Gracián, the Art of Worldly Wisdom [17]
Preface

The Kepler Conjecture

In 1611, Johannes Kepler wrote a booklet in which he asserted that the familiar cannonball arrangement of congruent balls in space achieves the highest possible density. This assertion has become known as the Kepler conjecture. This book presents a proof.

As early as 1831, Gauss established a special case of the conjecture, by proving that the cannonball arrangement is optimal among all lattices [14]. Later in the nineteenth century, Thue solved the corresponding problem in two dimensions, showing that the hexagonal arrangement of disks in the plane achieves optimal density [45] and [46]. Hilbert, in his famous list of mathematical problems, made the Kepler conjecture part of his eighteenth problem. In 1953, Fejes Tóth formulated a general strategy to confirm the Kepler conjecture, but lacked the computational resources to carry it out [12]. The conjecture was finally resolved in 1998, even though the full proof was not published until 2006 [22]. Section 1.1 gives additional historical background.

The Kepler conjecture has become a test of the capability of computers to deliver a reliable mathematical proof. The original proof involved many long computer calculations that led a team of referees to exhaustion. This book has redesigned the proof in a way that makes the correctness of the computer proof as transparent as possible.

Formal Proofs

After all is said and done, a proof is only as reliable as the processes that are used to verify its correctness. The ultimate standard of proof is a formal proof, which is nothing other than an unbroken chain of logical inferences from an explicit set of axioms. While this may be the mathematical ideal of proof, actual mathematical practice generally deviates significantly from the ideal.

In recent years, as part of this project, I have been increasingly preoccupied by the processes that mathematicians rely on to ensure the correctness of complex proofs. A century ago, Russell’s paradox and other antinomies threatened set theory with fires of destruction. Researchers from Frege to Gödel solved the problem of rigor in mathematics and found a theoretical solution but did not extinguish the fire at the foundations of mathematics because they omitted the practical implementation. Some, such as Bourbaki, have even gone so far as to claim that “formalized mathematics cannot in practice be written down in full” and call such a project “absolutely unrealizable” [7, pp. 10–11].

While it is true that formal proofs may be too long to print, computers –
which do not have the same limitations as paper – have become the natural host of formal mathematics. In recent decades, logicians and computer scientists have reworked the foundations of mathematics, putting them in an efficient form designed for real use on real computers.

For the first time in history, it is possible to generate and verify every single logical inference of major mathematical theorems. This has now been done for many theorems, including the four-color theorem, the prime number theorem, the Jordan curve theorem, the Brouwer fixed point theorem, and the fundamental theorem of calculus. Freek Wiedijk reports that 87% of a list of one hundred famous theorems have now been checked formally [50]. The list of remaining theorems contains two particular challenges: the independence of the Continuum hypothesis and Fermat’s Last theorem.

Some mathematicians remain skeptical of the process because computers have been used to generate and verify the logical inferences. Computers are notoriously imperfect, with flaws ranging from software bugs to defective chips. Even if a computer verifies the inferences, who verifies the verifier, or then verifies the verifier of the verifier? Indeed, it would be unscientific of us to place an unmerited trust in computers.

The choice comes down to two competing verification processes. The first is the traditional process of referees, which depends largely on the luck of the draw – some referees are meticulous, while others are careless. The second process is formal computer verification, which is less dependent on the whims of a particular referee. In my view, the choice between the conventional process by human referee and computer verification is as evident as the choice between a sundial and an atomic clock in science.

The standard of proof I have adopted is the highest scientific standard available by current technology. The introduction of steel in architecture is not a mere reinforcement of wood and stone; it changes the world of structural possibilities. There is no longer any reason to limit proofs to ten thousand pages when our technology supports a million pages.

The style of formal proofs is different from that of conventional ones. It is easier to formalize several short snappy proofs than a few intricate ones. Humans enjoy surprising new perspectives, but computers benefit from repetition and standardization. Despite these differences, I have sought proofs that might bring pleasure to the human reader while providing precise instructions for the implementation in silicon.
Preface

Conventions

To make formalization proceed more smoothly, long proofs have been broken
into a sequence of smaller claims. Each claim starts a new paragraph and is set
in italics. The second sentence of the paragraph begins with the word indeed
when the proof of the claim is direct and with the word otherwise when the
proof is indirect by contradiction.

Lemmas and theorems that are marked with an asterisk appear out of the
natural logical sequence. Care should be taken to avoid logical gaps when they
are cited.

The pronoun we is used inclusively for the author and reader as we work our
way through the proofs in this book. The pronoun I refers to the author alone.

The asterisk * is used as a wildcard symbol in patterns. It replaces a term in
contexts where the name of the term is not relevant. It can also denote a bound
variable. For example, the function \( f(\ast, y) \) of a single variable is obtained from
\( f \) by evaluating the second argument at a fixed value \( y \).

The union of the family \( X \) of sets is written as \( \bigcup X \) or as \( \bigcup_{x \in X} x \) without
any difference in meaning. The first form is preferred because of its economy.
We also use both expressions \( \bigcap X \) and \( \bigcap_{x \in X} x \) for the intersection of a family
of sets.

The documentation of the computer calculations for the Kepler conjecture
has evolved over time. The 1998 preprint version of the proof of the Kepler
conjecture contains long appendices that list hundreds of calculations that en-
ter into the proof. These appendices were cut from the published version of the
proof because it is more useful to store the computer part of the proof at a com-
puter code repository that is permanent, versioned, and freely available. The
computer code and documentation are housed at Google Code project host-
ing. Separate documentation, which is available at the project site, describes
the computer calculations that appear in this book. When this book uses an
external calculation, it is marked in italic font as a computer calculation\(^1\) [21].

A Blueprint

The book is a blueprint for formal proofs because it gives the design of the
formal proof to be constructed. The parts of this book that cover the text por-
tions of the proof of the Kepler conjecture are being formally verified in the
proof assistant HOL Light. I dream of a fully formally verified solution to the

\(^1\) [notation] This explains notation.
Preface

proof that includes the computer portions of the proof as well. Details about and credits for this large team effort appear in Appendix A.

Decisions about what to include in this book have been shaped by the list of theorems already available in the library of the proof assistant HOL Light. For example, this book accepts basic point-set topology and measure theory because they have been formalized by Harrison [26].

The book is divided into three parts, the first of which describes the major ideas, methods, and organization of the proof.

The part on foundations provides background material about constructions in discrete geometry. The first of these chapters covers trigonometric identities and basic vector geometry. The second treats volume from an elementary point of view. The third chapter covers planar graph theory from a purely combinatorial perspective. The fourth chapter continues with planar graphs, now from a geometric perspective.

The final part of the book gives the solution to the packing problem. The first of these chapters gives a top-level overview of the major steps of the proof, describing how the problem can be reduced from a problem with infinitely many variables to one in finitely many variables. The remaining chapters in this part flesh out the proof.

The final section of the book views dense sphere packings from a larger perspective. It resolves another longstanding conjecture in discrete geometry: Bezdek’s strong dodecahedral conjecture.

Simplifications

Many simplifications of the original proof have been found over the past several years. These simplifications are published here for the first time. Gonthier has reworked the proof of the four-color theorem to avoid the use of the Jordan curve theorem, using instead the much simpler notion of Möbius contour from the theory of hypermaps. I have followed Gonthier’s lead.

The optimality of the face-centered cubic packing is an assertion about infinite space-filling packings. For computational purposes, it is useful to reduce the sphere packing problem to finite packings. A correction term is associated with each different reduction from infinite packings to finite packings. Ferguson and I worked together to produce the original proof of the Kepler conjecture. The two of us considered a large number of different correction terms, seeking one that would simplify the computations as much as possible. In a discussion of the solution of the packing problem, I wrote that “correction terms are extremely flexible and easy to construct, and soon Samuel Fergu-
son and I realized that every time we encountered difficulties in solving the minimization problem, we could adjust $f$ [the correction term] to skirt the difficulty. . . If I were to revise the proof to produce a simpler one, the first thing I would do would be to change the correction term once again. It is the key to a simpler proof” [19]. Marchal has recently found a simple correction term, giving a new way to reduce from infinite packings to finite packings [31]. This book implements his reduction step.

There are many other improvements of the proof that are not visible in the book because they are implemented in computer code, including a reduction of the number of lines of computer code from over 187,000 to about 10,000. Needless to say, the quickest way to be sure that a block of computer code will not execute a bug is to delete the code altogether.

Thomas C. Hales
Pittsburgh, PA