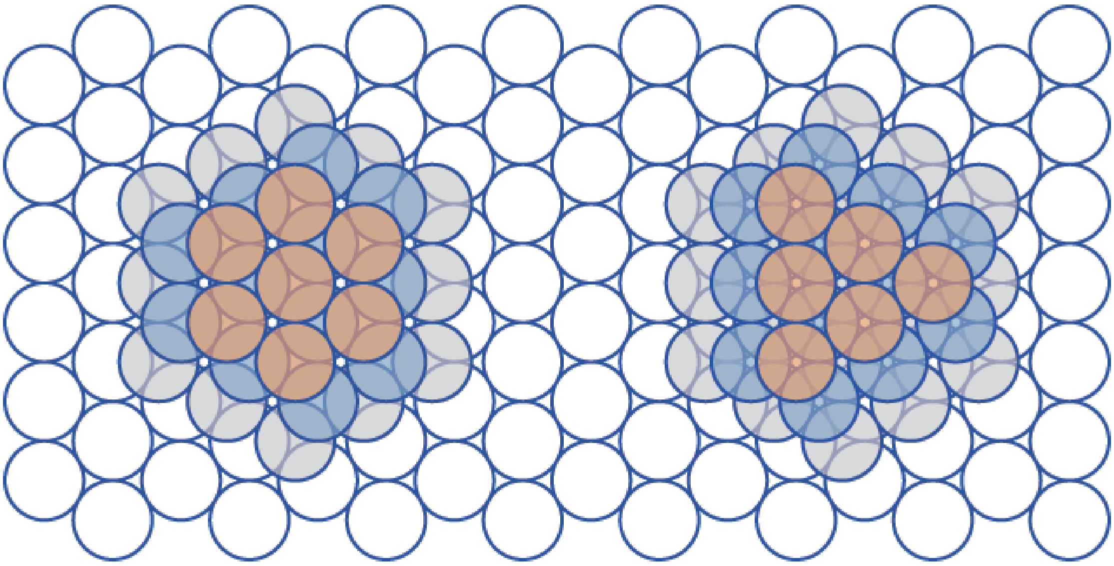


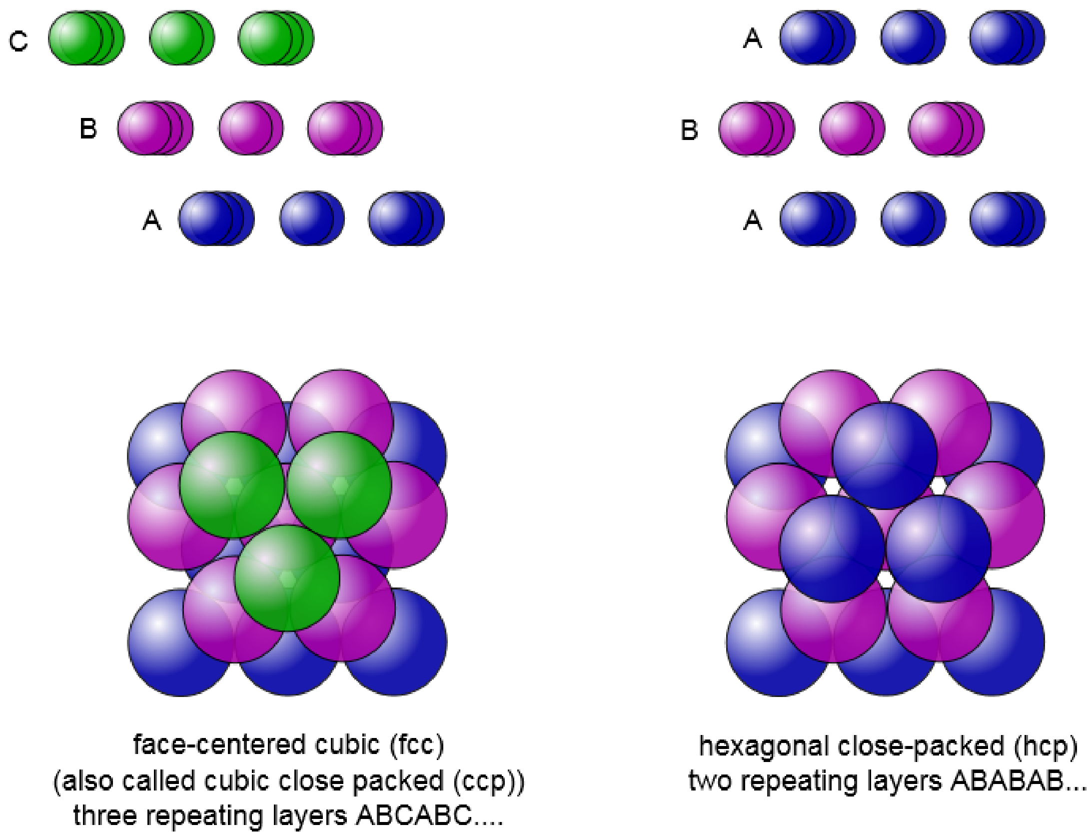
The Kepler Conjecture

Adrian Rauchhaus

Formaler Beweis in Mathematik und Informatik

Contents

Figure 1: Two of the densest packings for equivalent spheres



1 Statement of the Theorem

The Kepler Conjecture states that there is no arrangement of equally sized spheres in the three-dimensional Euclidean space with a higher average density than the *cubic close packing* (also called *face-centered cubic packing*) and the *hexagonal close packing* (see figure 1). Both of these have an average density of $\pi/\sqrt{18}$ (around 74.05%). The theorem does not claim uniqueness of a densest packing.

The *density* of an infinite packing V is defined as the limit of the density obtained within finite spherical containers as the size of the containers grows to infinity.

2 History of the problem and its proof

The Kepler Conjecture was presumed by the famous mathematician Johannes Kepler (1571-1630) in the early 17th century in connection to the question how to store cannon balls most efficiently. It is the oldest problem in discrete geometry and was part of Hilbert's 18th problem. In 1950, Fejes Tóth offered a proof strategy for the Kepler Conjecture and considered the use of computers to study the problem. In 1998 Samuel Fergusen and Thomas Hales proved the statement to be true, but only published the full proof in 2006, since the referees had trouble to reliably check every step of the complex computer proof. Though the reviewers came to the conclusion that the proof was essentially accurate, they were not absolutely sure of that. At the Joint Math Meetings in Baltimore in 2003, Hales announced the *Flyspeck project*[1], with the idea of formalizing the text parts of the proof, as well as the calculations implemented in code. The project finished its work in August 2014.

3 The proof assistants

For the verification of the proof three proof assistants were used: HOL Light, Isabelle and HOL Zero.
HOL Light is considered to be extremely reliable due to its design and compactly written kernel, and was thus used for the majority of the proof. It is especially useful for the *Flyspeck project* since it already contains large libraries of proven results including differential calculus and point-set topology on $\mathbb{R}^n$. The program is based on the language OCaml and has a special syntax for mathematical expressions that makes it less vulnerable to imprecisions of machine arithmetic than OCaml.
The proof assistant Isabelle/HOL is very similar to HOL Light with the addition of some properties, most importantly computational reflection, which is neccessary for the *tame graph classification*, but not supported by HOL Light. Isabelle also supports a module system and type classes, and can export certain terms as ML, execute them and reintegrate the results as theorems.
HOL Zero is only required for a second check of the main statement.

[1]The name originates from the acronym FPK for *Formal Proof of the Kepler Conjecture.*

4 Formalization of the theorem

As the density of a packing V is defined via a limit, the statement is formalized by determining the density over a finite spherical container with an error term that, divided by the total volume, tends to zero for growing containers.
The density is scale invariant (as it is a ratio of volumes) and one can w.l.o.g. assume that the spheres in the packing are unit balls. These can be identified with their centers, such that the packing is represented by a set of points. This immediately implies that the distance between any two objects in V is at least two (the diameter of one sphere) or else the objects are identical. In HOL Light this is expressed as follows:

```
|- packing V <=>
        (!u v. u IN V /\ v IN V /\ dist(u,v) < &2 ==> u = v )
```

The |− indicates the beginning of a theroem, ! and ? symbolize the $\forall$ and $\exists$ quantifiers, & the embedding of natural numbers in the real numbers (i.e. &2 = 2.0), /\ is the logical $\wedge$, and ==> and <=> stand for $\Rightarrow$ and $\Leftrightarrow$. As one can see, the mathematical signs are usually approximated by ASCII characters.

The statement of the Kepler Conjecture is captured in the constant *the_kepler_konjecture* and looks like this

```
|- the_kepler_conjecture <=>
        (!V. packing V
        ==> (?c. !r. &1 <= r
                ==> &(CARD(V INTER ball(vec 0,r))) <=
                        pi * r pow 3 / sqrt(&18) + c * r pow 2)
        )
```

"packing" is the class of all packings, "CARD" the cardinality of a set, "INTER" the intersection of sets and "ball(vec 0,r)" a sphere with radius r around the center of the Euclidean space.
This translates to: For every packing V there exists a constant c such that for all constants r bigger or equal to 1, the cardinality[2] of the intersection of V and the sphere with center 0 and radius r is less than or equal to $\pi * r^3/\sqrt{18} + c * r^2$.

[2] The cardinality of the intersection is always finite, because the packing is represented by discrete points with minimum distance two.

The proof of this statement can naturally be divided into four main parts, the formalization and three different parts of calculations, specified in HOL Light under the following names:

1. *the_nonlinear_inequalities*: A list of nearly a thousand nonlinear inequalities
2. *import_tame_classification*: Each possible counterexample to the Kepler Conjecture can be encoded as a plane graph satisfying a set of conditions, which classify it as *tame*. An exhaustive computer search has generated the finite list of *tame plane graphs* (up to isomorphism). It is necessary to show that every tame plane graph is isomorphic to one of these graphs.
3. *linear_programming_results*: A large collection of linear programs. These are shown to be infeasible, which disproves the existence of the counterexamples.

Due to the sheer size the whole proof was not obtained in a single session of HOL Light. Instead another theorem was formalized that represents the formalization of the text part and the proof of the linear programming and explicitly assumes the other two parts:

```
|- the_nonlinear_inequalities /\ import_tame_classification
        ==> the_kepler_conjecture
```

5 Proof of the theorem

There are two versions of the proof, the *original proof* and the *blueprint proof*. The formalization of the proof was developed simultaniously to the blueprint proof and follows it closely.

5.1 Outline of the proof

Consider an arbitrary packing V of unit balls in the Euclidean space with the properties necessary for it to be a counterexample. Now one reduces the problem with infinitely many spheres to a one with finitely many by partitioning the Euclidean space into so called Marchal cells.
A sphere on the boundary of the cell is called a *vertex*, a line segment on the boundary of the cell between two vertices is called an *edge*. Some edges are called *critical* if they satisfy a specific length condition, and cells that share a critical edge form a *cell cluster*.
Each cell X (and one of its critical edges) in a cell cluster gets assigned a real number $\Gamma(\epsilon, X)$, depending on its volume and the angles between and the lengths

of the cells edges.

To represent the Kepler Conjecture as a local optimization problem one uses the *cell-cluster inequality*

$$\forall \text{ critical edges } \epsilon : \sum_{X \in C} \Gamma(\epsilon, X) \geq 0$$

where C is the cell cluster induced by the packing, and the *local annulus inequality*:
The *constant ball annulus* is defined as the set $A = \{x \in \mathbb{R}^3 : 2 \leq \|x\| \leq 2.52\}$. As A is compact and V discrete the intersection is finite. For $f(t) := \frac{2.52-t}{2.52-2}$ (this function decays from 1 to 0 on A) the *local annulus inequality* for $V \subset A$ is defined as

$$\sum_{v \in V} f(\|v\|) \leq 12$$

where v are the vertices.

With these inequalities the Kepler Conjecture (which is a problem about volumes and densities) can be transformed into a problem of distances between spheres and leads to this intermediate result:

```
|- the_nonlinear_inequalities /\
        (!V. cell_cluster_inequality V) /\
        (!V. packing V /\ V SUBSET ball_annulus
                ==> local_annulus_inequality V)
        ==> the_kepler_conjecture
```

That means that if the nonlinear inequalities and the cell cluster inequality are proven[3] all that is left to show, is that the local annulus inequality holds.
The local annulus equation is proven by showing that every possible counterexample is infeasible. One assumes there is a counter example V[4]. Since A is compact we can assume V has special properties that we sum up under the term *contravening*. These imply that V is isomorphic to a special combinatorical structure called *tame planar hypermap* (which one can imagine as plane graphs with certain restrictions). The number of the hypermaps is finite up to isomorphism which leads to a finite number of possible counterexamples. For each given tame planar

[3] The proof of the cell-cluster inequality is a computer calculation that reduces it to nonlinear inequalities.

[4] It is sufficient to consider V with at most 15 elements.

hypermaps H one considers all associated contravening packings V. All of these violate the local annulus inequality by definition.
The conditions on V can be expressed by a system of nonlinear inequalities which gets relaxed to a system of linear inequalities that is shown to be infeasible via linear programming techniques. One concludes that the nonlinear system is inconsistent and thereby contradicts the existence of the corresponding counterexample. Exhaustive repetition of this procedure leads to the refutation of every possible counterexample and concludes the proof.

5.2 The nonlinear inequalities

Nearly all nonlinear inequalities in the Flyspeck project have the form

$$\forall x, x \in D \Rightarrow f_1(x) < 0 \wedge \cdots \wedge f_k(x) < 0$$

with $n \in \mathbb{N},\ n \leq 6,\ D = [a_1, b_1] \times \cdots \times [a_n, b_n]$ and $x = (x_1, \ldots, x_n)$. For the remaining inequalities k equals 1 and the inequality is not strict. The inequalities contain basic arithmetic operations, square roots, trigonometric functions and the analytic continuation of $\arctan(\sqrt{x})/\sqrt{x}$ to the region $x > -1$. For every $x \in D$ at least one f_i is analytic around x (and takes a negative value).

The inequalities are handled by using interval arithmetic, i.e. numbers are approximated by an upper and a lower bound, for example $[3.14, 3.15]$ as an approximation of π. This is only possible if the arithmetic operations are defined over intervals: Let $\mathbb{IR}$ denote the set of intervals over the real numbers. The interval extension $f : \mathbb{IR} \rightarrow \mathbb{IR}$ of $f : \mathbb{R} \rightarrow \mathbb{R}$ satisfies

$$\forall I \in \mathbb{IR},\ \{f(x) : x \in I\} \subset F(I)$$

and can easily be extended to $\mathbb{R}^k$.

Arithmetic operations are similarly expanded for intervals, for example the sum of intervals

$$[a_1, b_1] \oplus [a_2, b_2] = [a, b]$$

for some $a \leq a_1 + a_2$ and $b \geq b_1 + b_2$.[5] Other arithmetic operations are defined analogously.
This natural way to define interval expansions can be imprecise and often needs improvement. An easy way to improve them is to devide the interval into subintervals and evaluate the function on all of them. Though this is a possibility, it

[5] Keep in mind that $\oplus$ is not defined as $[a_1 + a_2, b_1 + b_2]$ since there might be differences due to imprecision.

increases the number of operations significantly, especially for multivariate functions.
For this reason the procedure implemented in OCaml and HOL Light works with interval extensions based on Taylor approximations instead. Even then it is sometimes required to work with subdivisions to get sufficiently good approximations. Another advantage of Taylor polynomials is the possibility to easily prove monotonicity, by expanding the derivative over an interval. If the target interval of the derivative does not contain zero, the maximum value over the interval occurs on its boundary. This can be used to reduce the verification of the inequality on a rectangle (the cartesian product of intervals the inequality is defined on) of dimension k to a rectangle of dimension $k-1$.

Through partitioning of the domains the several hundred nonlinear inequalities become more than 23000. With a verification time of approximately 5000 hours in HOL Light this is the most laborious part of the proof[6] (the other two sets of calculations can be verified in less than a day). This is a problem, because the inequalities have to be obtained in several sessions, but in the regular version of HOL Light it is not possible to transfer a theorem (in this case the correctness of the inequality) without a reconstruction of the whole proof. Therefore a slightly modified version of HOL Light was used to combine the results that were obtained in parallel sessions of calculation.

5.3 Tame classification

Classifying tame plane graphs was the first major success of the Flyspeck project. The tame plane graphs encode the possible counterexamples to the Kepler Conjecture as plane graphs. The computer-generated list of tame graphs is collected in a text file called the *archive* and can be imported into the proof. The goal is to formalize the following completeness theorem in Isabelle/HOL

$$\vdash \text{"}g \in \text{PlaneGraphs" and "tame } g\text{" shows "fgraph } g \in_{\simeq} \text{Archive"},$$

which means that every tame plane graph is isomorphic to a graph appearing in the archive. The formalization of the graph includes a list its faces (represented by their nodes, which are again represented by integer indices). fgraph is a function that reduces the graph to the list of faces. To show the completeness of the archive it is neccessary to enumerate all tame plane graphs. This is done through the functional programming language in HOL and leads to a set of graphs called *TameEnum*. To prove completeness of *TameEnum* one has to show that it contains

[6]But since most of the vericfications were done in parallel with 32 cores it took less than a week in total.

every tame graph and that every graph it contains is isomorphic to a graph in the archive. Formally that looks like

$$\vdash \text{fgraph ' TameEnum } \subseteq_{\simeq} \text{Archive}$$

This formula can automatically be proven by Isabelle through computational reflection.

5.3.1 General enumeration of plane graphs

Plane graphs are defined algorithmically by starting with a polygon and subsequentially adding loops to it. This leads to a natural way of enumerating them. The initial polygon serves as a seed graph which creates a partition of the plane with a final (outer) and a nonfinal (inner) component (final means the algorithm is not allowed to further divide the face). The nonfinal component can then be subdivided in several ways to create a new graph and so on. Through this process a forrest of graphs is defined whose leaves are final graphs. An executable function called *next_plane* maps a graph to the set of graphs obtainable by dividing one face. In Isabelle/HOL the set of final graphs reachable from some seed graph in finitely many steps is called *PlaneGraphs*.

5.3.2 Enumeration of the tame graphs

There are two crucial properties for a graph to fulfill to be called tame: The faces have to be triangles or hexagonal and the admissible weight has to be bounded. One can show that this is also sufficient to prove the finiteness (up to isomorphy) of tame plane graphs. The enumeration is derived from the enumeration of plane graphs with elimination of all non-tame final graphs and nonfinal graphs that do not produce any tame ones. The *pruning*[7] criteria can be implemented into theprogram with various degrees of complexity. The weaker the criteria the easier they are to justify, but also the longer is the running time of the algorithm and vice versa. For the formalization process a modified version of the function *next_plane* called *next_tame* is used, that applies the described pruning procedure. To guarantee correctness the used algorithm works with approximations that never eliminate a tame graph, but might produce nontame graphs, so no possible counterexample is missed. As a result some fake counterexamples may be produced. Luckily these get eliminated in later steps. The set of tame graphs produced with the function *next_tame* is called *TameEnum*.

While most of the proof of the Kepler Conjecture was done in HOL Light, the tame

[7]This means eliminating graphs from the tree.

graph classification was done in Isabelle/HOL, since HOL Light does not permit computational reflection and it is not possible to import results from Isabelle to HOL Light automatically. For this reason both systems are used and HOL Light treats the parts proven by Isabelle as assumptions. Formally this is represented as

```
|- import_tame_classifiation <=>
        (!g. g IN PlaneGraphs /\ tame g ==>
         fgraph g IN_simeq archive)
```

The right-hand side is exactly what was expressed above. The terms *PlaneGraphs, tame, archive, In_simeq* and *fgraph* in HOL Light are basically just translations of the corresponding definitions in Isabelle.

5.4 Linear programs

The infinitely large potential counterexamples are first reduced to finite ones and then encoded as tame planar hypermaps. For every hypermap there is a list of inequalities (mostly nonlinear) the corresponding counterexamples have to satisfy. If one can show that the system of inequalities is infeasible, the counterexample can be eliminated.
By replacing nonlinear terms in the inequalities with new variables it is possible to obtain linear relaxations. Showing that a linear program is infeasible proves inconsistency of the original inequalities and the nonexistence of the contravening packings associated with the hypermap. Since there are essentially finitely many tame planar hypermaps (as shown in the tame graph classification theorem) all possible counterexamples can successively be ruled out.
Generating the linear programs from the nonlinear inequalities is the first step of their verification procedure. To demonstrate the transformation from nonlinear to linear inequalities we have a look at the following example:
Say one has the inequalities $x + x^2 \leq 3$ and $x \geq 2$. Substitute $y := x^2$. This changes the first inequality to $x + y \leq 3$ and the second implies $y \leq 4$ and thus $x + y \geq 6$, a contradiction.

Instead of solving inequalities over irrational constants directly, the proof assistant solves an implied rational inequality. For instance $x \geq \pi$ implies $x \geq 3.14$ (and so on). These are then modified to inequalities over integer coefficients by multiplying with powers of 10.

The second and last step is the introduction of free variables with values depending on the properties of the associated tame planar hypermap. Thus every

hypermap produces a linear program that is checked for feasibility. But due to the approximate values of the integer inequalities only about half of the inequalities can directly be proven to be infeasible. For the other half it is necessary to split the inequality into several cases which lead to more precise relaxations, i.e. introduce new inequalities of the form $x \leq a$ and $x \geq a$ and check these separately for a constant a. This is automatically done by the proof assistant.
There are ultimately 43078 cases to consider, which can be verified in about 15 hours on a 2.4 GHz computer.
While most of the linear programs are generated and solved automatically, some require a manual formal proof.
Applying this procedure for all possible counterexamples and subsequently eliminating them concludes the proof of the Kepler Conjecture.

6 Conclusion

In this proof the proof assistants mostly replace the human referees, since deeper mathematical insight is scarcely needed. Instead the proof can be checked by sufficiently trained users of HOL Light and Isabelle. The most important parts for reviewers to check are whether or not the right theorems are formalized and if all assumed axioms are permissable. The biggest issue with this proof is that it was not obtained in a single session but in several, over two different proof assistants. This makes it essential to meticulously survey the translation of statements from Isabelle to HOL Light.
What made the Kepler Conjecture unprovable for a few hundred years was the sheer amount of counterexamples one has to refute. Solving of all the calculations demanded a calculation capacity not available for most of the time since the question arose, and even then the verification was immensely intricate.

References

- Hales, T., Adams, M., Bauer, G., Dang, T., Harrison, J., Hoang, L., . . . Zumkeller, R. (2017). *A formal proof of the Kepler Conjecture.* Forum of Mathematics, Pi, 5, E2. doi:10.1017/fmp.2017.1

- Hales, T. C., *Dense Sphere Packings: a Blueprint for Formal Proofs*, London Mathematical Society Lecture Note Series, 400 (Cambridge University Press, 2012)

- George G. Szpiro. *Die Keplersche Vermutung.* Springer-Verlag, 2011

Pictures

- https://hexnet.org/content/close-packing-spheres

- https://upload.wikimedia.org/wikipedia/commons/2/2e/Closepacking.svg