Linear Programs for the Kepler Conjecture (extended abstract)

Thomas C. Hales*

University of Pittsburgh

Abstract. The Kepler conjecture asserts that the densest arrangement of congruent balls in Euclidean three-space is the face-centered cubic packing, which is the familiar pyramid arrangement used to stack oranges at the market. The problem was finally solved in 1998 by a long computer proof. The Flyspeck project seeks to give a full formal proof of the Kepler conjecture. This is an extended abstract for a talk in the formal proof session of ICMS-2010, which will describe the linear programming aspects of the Flyspeck project.

The original proof of the Kepler conjecture in 1998 was about three hundred pages long and relied long computer calculations that were done by custom computer code [5]. The amount of custom computer code was originally estimated to be about 40,000 lines, but later estimates give a number closer to 180,000 lines [4]. The larger figure includes a large difficult-to-estimate number of duplicated lines.

The computer code consists of three separate programs. The first is a program that generates all hypermaps up to isomorphism with prescribed properties. (A hypermap, defined below, is a combinatorial structure that conveniently encodes the structure of a planar graph.) The output of this program is a set X of about 18 thousand hypermaps. The second is a program that proves nonlinear inequalities. The third program generates and solves linear programs.

It is this third program that is the subject of this extended abstract. This third program was considered from a formal proof perspective in Obua's thesis [7]. The thesis gives a formal verification of over 90% of the cases. Why did he not complete the remaining 10% of the cases? He did not encounter any difficulties with the technology. Indeed, his work demonstrates that the formal proof of linear programs is entirely feasible.

Rather, the difficulties are with the slopppy documentation in the original proof of the Kepler conjecture. The linear programming part of the proof is most poorly documented part of the proof. The original linear programs were generated in interactive Mathematica sessions [3]. There are voluminous notes about these interactive sessions, but these notes are not in the form of executable code. In the end, the remaining cases were not formalized because it was not clear what precisely was to be formalized.

* Research supported by NSF grant 0804189 and a grant from the Benter Foundation.

1 Linear Programs

The linear programming part of the proof has been reworked from the start with the formal proof in mind [6]. The methods described in this extended abstract treat 99.93% of the cases (that is, all but 12 hypermaps).

The linear programs are specified in the *MathProg* language, which is a domain specific language designed for linear programming in the GNU Linear Programming Kit (GLPK) [1]. MathProg is a subset of the *AMPL* language [2]. According to the design of MathProg, the linear program is split into two parts: a model and data. The model contains the declarations of variables, parameters, and indexing sets. There is a single model file shared by all linear programs. There is a separate data file for each linear program. The model is further divided into a header and a body.

The computer code for this project has been written in Objective Caml. The data files as well as the body of the model are automatically generated. The computer code has not yet been formalized, but it is now in a formalization-ready state. There are a few different parts to the code.

- (200 lines) interface with GLPK,
- (200 lines) MathProg format model header,
- (80 lines) model body code generator,
- (300 lines) data generator and control flow.

The are two additional data files:

- (4MB) a set X of about 18 thousand hypermaps,
- (1000 lines) archive of nonlinear inequalities (in HOL Light format).

This is a significant improvement over the original program from the 1998 proof of the Kepler conjecture, which involved several thousand lines of computer code, 3GB of data, and long interactive sessions.

2 The Main Theorem

As we mentioned above, in the original proof of the Kepler conjecture, it is difficult even to state the theorem that has been proved by the linear programming part of the proof. Here we give a simple statement that captures the linear programming part of the proof of the Kepler conjecture.

Definition 1. *A hypermap is a finite set D with two permutations $n, f : D \to D$. A third permutation e is defined by the relation $enf = I$. We represent the hypermap as a tuple $x = (D, e, n, f)$.*

Let X be the set of over 18 thousand hypermaps that is mentioned above. Each of these hypermaps can be augmented by various *markings* to produce what we call marked hypermaps. Let Y be the finite set of all marked hypermaps and let $Y_x \subset Y$ be those that come from $x \in X$. Certain subsets of Y_x are called *covers* of x. Associated with each marked hypermap $y \in Y$ is a polyhedron $P(y)$. The main theorem about linear programs takes the following form.

Theorem 1. *For every hypermap $x \in X$, there exists a cover $U \subset Y_x$ such that for every $y \in U$, the polyhedron $P(y)$ is empty.*

The proof is by a direct construction carried out by computer. Each polyhedron is explicitly presented as a finite system of linear inequalities. Linear programming methods show that each system of linear inequalities has no solutions. The existence of a cover is established by a direct computer search. It takes about 2.5 hours to run the program on a laptop computer.

Finally, we describe the relationship between this theorem and the Kepler conjecture. The Kepler conjecture is initially expressed as a statement about packings of congruent balls in an unbounded region of space. Various reduction arguments reduce the proof of the conjecture to a conjecture about packings of finitely many balls.

The proof of the Kepler conjecture is by contradiction. If the Kepler conjecture is false, then there exists a finite packing V of at most 15 balls that has various remarkable properties. The balls in the packing form the set of nodes of a graph (V, E), and the combinatorial properties of the graph can be encoded as a hypermap x in the set X. With a counterexample V in hand, for every cover U of x, it is possible to find a marked hypermap $y \in U$ for which the polyhedron $P(y)$ is nonempty. The existence of a counterexample V contradicts the theorem, which asserts on the contrary that $P(y)$ is empty.

References

1. *GLPK (GNU Linear Programming Kit)*, `http://www.gnu.org/software/glpk/`.
2. Rober Fourer, David M. Gay, and Brian W. Kernighan, *The AMPL book*, Brooks/Cole, 2002.
3. T. C. Hales, *Computer resources for the Kepler conjecture*, 2003, `http://www.math.princeton.edu/\~\!annals/KeplerConjecture/` (2005 snapshot).
4. T. C. Hales, J. Harrison, S. McLaughlin, T. Nipkow, S. Obua, and R. Zumkeller, *A revision of the proof of the Kepler Conjecture*, DCG (2009).
5. Thomas C. Hales and Samuel P. Ferguson, *The Kepler conjecture*, Discrete and Computational Geometry **36** (2006), no. 1, 1–269.
6. Thomas C. Hales, Alexey Solovyev, and Hoang Le Truong et al., *The Flyspeck Project*, 2014, `https://github.com/flyspeck/flyspeck/`.
7. Steven Obua, *Flyspeck II: The basic linear programs*, Ph.D. thesis, Technische Universität München, 2008, `http://deposit.d-nb.de/cgi-bin/dokserv?idn=992033632&dok_var=d1&dok_ext=pdf&filename=992033632.pdf` `http://mediatum2.ub.tum.de/doc/645669/645669.pdf`.