

A REVISION OF THE PROOF OF THE KEPLER CONJECTURE

THOMAS C. HALES, JOHN HARRISON, SEAN MCLAUGHLIN, TOBIAS NIPKOW,
STEVEN OBUA, AND ROLAND ZUMKELLER

ABSTRACT. The Kepler conjecture asserts that no packing of congruent balls in three-dimensional Euclidean space has density greater than that of the face-centered cubic packing. The original proof, announced in 1998 and published in 2006, is long and complex. The process of revision and review did not end with the publication of the proof. This article summarizes the current status of several long-term initiatives to reorganize the original proof into a more transparent form and to provide a greater level of certification of the correctness of the computer code and other details of the proof. A final part of this article lists errata in the original proof of the Kepler conjecture.

CONTENTS

Introduction	2
Part 1. Formal Proof Initiatives	3
1. The Flyspeck project	3
2. Blueprint edition of the Kepler conjecture	5
3. Formalizing the ordinary mathematics	7
4. Standard ML reimplementations of code	10
5. Proving nonlinear inequalities with Bernstein bases	12
6. Tame graph enumeration	16
7. Verifying linear programs	19
Part 2. Errata in the Original Proof	21
8. Biconnected graphs	21
9. Errata listing	34
References	37

INTRODUCTION

In 2006, *Discrete and Computational Geometry* devoted an issue to the proof of the Kepler conjecture on sphere packings, which asserts that no packing of congruent balls in three-dimensional Euclidean space can have density greater than that of the face-centered cubic packing [22], [13], [14], [15], [5].

The proof is long and complex. The editors' forward to that issue remarks that "the reviewing of these papers was a particularly enormous and daunting task." "The main portion of the reviewing took place in a seminar run at Eötvös University over a 3 year period. Some computer experiments were done in a detailed check. The nature of this proof, consisting in part of a large number of inequalities having little internal structure, and a complicated proof tree, makes it hard for humans to check every step reliably. Detailed checking of specific assertions found them to be essentially correct in every case tested. The reviewing process produced in the reviewers a strong degree of conviction of the essential correctness of the proof approach, and that the reduction method led to nonlinear programming problems of tractable size."

The process of review and revision did not end when the proof was published. This article summarizes the current status of several long-term initiatives to reorganize the original proof into a more transparent form and to provide a greater level of certification of the correctness of the computer code and other details of the proof.

The article contains two parts. The first part describes various initiatives to give a formal proof of the Kepler conjecture. The second part gives errata in the original proof of the Kepler conjecture. Most of these errata are minor. The most significant new argument appears in a separate section (Section 8). It finishes an incomplete argument in the original proof asserting that there is no loss in generality in assuming (for purposes of the main estimate) that subregions are simple polygons. The incomplete argument was detected during the preparation of the blueprint edition of the proof, which is described in Section 2.

In this article, the *original proof* refers to the proof published in [22].

Part 1. Formal Proof Initiatives

1. THE FLYSPECK PROJECT

The purpose of a long-term project, called the Flyspeck project, is to give a formal proof of the Kepler conjecture. This section makes some preliminary remarks about formal proofs and gives a general overview of the current status of this project.

1.1. Formal proof. A formal proof is a proof in which every logical inference has been checked, all the way back to the foundational axioms of mathematics. No step is skipped no matter how obvious it may be to a mathematician. A formal proof may be less intuitive, and yet is less susceptible to logical errors. Because of the large number of inferences involved, a computer is used to check the steps of a formal proof.

It is a large labor-intensive undertaking to transform a traditional proof into a formal proof. The first stage is to expand the traditional proof in greater detail. This stage fills in steps that a mathematician would regard as obvious, works out arguments that the original proof leaves to the reader, and supplies the assumed background knowledge. In a final stage, the detailed text is transcribed into a computer-readable format inside a computer proof assistant. The proof assistant contains mathematical axioms, logical rules of inference, and a collection of previously proved theorems. It validates each new lemma by stepping through each inference. No other currently available technology is able to provide levels of certification of a complex mathematical proof that is remotely comparable to that available by formal computer verification. A general overview of formal proofs can be found at [7, 21, 27].

Proof assistants differ in detail in the way they treat the formalization of a theorem that is itself a computer verification (such as the proof of the four color theorem or the proof of the Kepler conjecture). In general, a formal proof of a computer verification can be viewed as a formal proof of the correctness of the computer code used in the verification. That is, the formal proof certifies that the the code is a bug-free implementation of its specification.

1.2. Formal proof of the Kepler conjecture. As mentioned above, the purpose of the Flyspeck project is to give a formal proof of the Kepler conjecture. (The project name *Flyspeck* comes from the acronym *FPK*, for the *Formal Proof* of the *Kepler* conjecture.) This is the most complex formal

proof ever undertaken. We estimate that it may take about twenty work-years to complete this formalization project.

The Flyspeck project is introduced in the article [12]. The project page gives the latest developments [17]. The project is now at an advanced stage; in fact, we estimate that the project is now about half-way complete. One of the main purposes of this article is to present a summary of the current status of this project.

In the original proof of the Kepler conjecture, there was a long mathematical text and three major pieces of computer code. The written part of the proof has been substantially revised with aims of the Flyspeck project in mind. Section 2 compares this revised text with the original. There is now a good match between the mathematical background assumed in the text and the mathematical material that is available in the proof assistant HOL Light. Section 3 describes the current level of support in HOL Light for the formalization of Euclidean space and measure theory. In the years following the publication of the original proof, S. McLaughlin has reworked and largely rewritten the entire body of code in a form that is more transparent and more amenable to formalization. Section 4 points out some difficulties in verifying the computer code in its original form and documents the reimplementa-

There have been three Ph.D. theses on the Flyspeck project, one devoted to each of the three major pieces of computer code. The first piece of computer code uses interval arithmetic to verify nonlinear inequalities. R. Zumkeller's thesis develops nonlinear inequality proving inside the proof assistant Coq [48]. Section 5 gives an example of this work. The second piece of computer code enumerates all tame graphs. (The definition of tameness is rather intricate; its key property is that the set of tame graphs includes all graphs that give a potential counterexample to the conjecture.) G. Bauer's thesis, together with subsequent work with T. Nipkow, completes the formal proof of the enumeration of tame graphs [36]. Section 6 gives a summary of this formalization project. The third piece of computer code generates and runs some 10^5 linear programs. These linear programs show that none of the potential counterexamples to the Kepler conjecture are actual counterexamples. S. Obua's thesis develops the technology to generate and verify the linear programs inside the proof assistant Isabelle [38]. Section 7 describes this research.

The ultimate aim is to develop a complete formal proof of the Kepler conjecture within a single proof assistant. Because of the scope of the problem and the number of researchers involved, different proof assistants have been used for different parts of the proof: HOL Light for background in Euclidean geometry and the text, Coq for nonlinear inequality verification, and Isabelle/HOL for graph enumeration and linear programming. This raises

the issue of how to translate a formal proof automatically from one proof assistant to another. Implementations of automated translation among the proof assistants HOL, Isabelle, and Coq can be found at [39], [32], [46], [4].

2. BLUEPRINT EDITION OF THE KEPLER CONJECTURE

The *blueprint edition* of the proof of the Kepler conjecture is a second-generation proof that contains far more explicit detail than the original proof. The blueprint edition is available at [18, 19]. Many proofs have been significantly simplified and systematized. It has been written in a manner to permit easy formalization. As its name might suggest, this version is intended as a blueprint for the construction of a formal proof. This section compares the blueprint edition with the original.

2.1. Lemmas in elementary geometry. A collection of about 200 lemmas that can be expressed in elementary terms has been extracted from the original proof and placed in a separate collection [19]. This has several advantages. First of all, these lemmas, although elementary, are precisely the parts of the original proof that put the greatest burden on the reader's geometrical intuition. (Many of these lemmas deal with the existence or non-existence of configurations of several points in \mathbb{R}^3 subject to various metric constraints.) Also, the lemmas can be stated without reference to the Kepler conjecture and all of the machinery that has been introduced to give a proof. Finally, the proofs rely on similar methods and are best considered together [16]. Section 3 on *Enhanced Automation* gives an approach to proving the lemmas in this collection.

These lemmas can be expressed in the first-order language of the real numbers; that is, they can be expressed in the syntax of first-order logic with equality (allowing quantifiers $\forall x$, $\exists x$ with variables running over the real numbers), the real constants 0, 1, and ring operations (+), (−), (·) on \mathbb{R} . In fact, L. Fejes Tóth's statement of the Kepler conjecture as an optimization problem in a finite number of variables can itself be expressed in the first-order language of the real numbers. (For this, the truncation used by L. Fejes Tóth must be modified slightly so that the truncated Voronoi cells are polyhedra.) Thus, it should come as no surprise that many of the intermediate lemmas in the proof can also be expressed in this manner.

For example, consider the statement asserting the existence of a circumcenter of a triangle: if three points in the plane are not collinear, then there exists a point in the plane that is equidistant from all three. This can be expressed in elementary terms as follows: for every (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , if there do not exist t_1, t_2, t_3 for which $t_1(x_1, y_1) + t_2(x_2, y_2) + t_3(x_3, y_3) = (0, 0)$ and

$t_1 + t_2 + t_3 = 1$, then there exists (x, y) such that

$$(x - x_1)^2 + (y - y_1)^2 = (x - x_2)^2 + (y - y_2)^2 = (x - x_3)^2 + (y - y_3)^2.$$

2.2. Background material. As we mentioned above, there is now a close match between what the blueprint edition assumes as background and what the proof assistant *HOL Light* provides, as described in Section 3. The blueprint edition develops substantial background material in trigonometry, measure and integration, hypermaps, and fans (a geometric realization of a hypermap). It turns out that only a small number of *primitive* volumes need to be computed for the proof of the Kepler conjecture. These primitives include the volume of a ball, a tetrahedron, and right circular cone. No line or surface integrals are required.

A hypermap consists of three permutations e, n, f (on a finite set D) that compose to the identity $e \circ n \circ f = I$. A hypermap is the combinatorial structure used by Gonthier in his formal proof of the four color theorem. In 2005, the proof of the Kepler conjecture was rewritten in terms of hypermaps, because it is better suited for formal proofs than planar graphs.

The Jordan curve theorem (JCT) has been formalized as a step in the Flyspeck project [20]. In fact, something weaker than the JCT is needed. The project only needs the JCT for curves on the surface of a unit sphere consisting of a finite number of arcs of great circles; that is, a spherical polygonal version of the JCT. In the proof of the Kepler conjecture, the combinatorial structure of a cluster of spheres is encoded as a hypermap. An Euler characteristic calculation, based on the JCT, shows that these hypermaps are planar. The planarity of these hypermaps is a crucial property that is used in the enumeration of tame graphs (Section 6). The background chapter in the blueprint edition contains detailed proofs of these facts.

The blueprint edition contains several introductory essays that introduce the main concepts in the proof, including the algorithms implemented by the computer code.

The formalization of the blueprint text started in 2008 with work of J. Rute (CMU) and the Hanoi Flyspeck group. The current members of this group are Trần Nam Trung, Nguyễn Tất Thắng, Hoàng Lê Trường, Nguyễn Quang Trường, Vũ Khắc Kỷ, Nguyễn Anh Tâm, Nguyễn Tuyên Hoàng, Nguyễn Đức Phương, Vương Anh Quyền, Phan Hoàng Chơn, and managed by Tạ Thị Hoài An. The blueprint formalization is still at an early stage.

3. FORMALIZING THE ORDINARY MATHEMATICS

This section describes some of the work on formalizing Euclidean space and measure theory, and the development of further proof automation, which should be useful in this endeavor.

The computer code in Flyspeck has so far received the lion's share of the formal effort. This is entirely reasonable since there are, or at least were, real questions about the feasibility of reproducing these results in a formal way. However, the Flyspeck proof includes a large amount of 'ordinary' mathematics, which also needs to be formalized. Here we are on fairly safe ground in principle, because by now we understand the formalization of mainstream mathematics quite well [47]. It is safe to predict that this formalization can be done, and we can even hope for a reasonably accurate estimate of the effort involved. Nevertheless, the formalization is certainly non-trivial and will require considerable work.

3.1. Formalizing Euclidean space. Much of our work has been devoted to developing a solid general theory of Euclidean space \mathbb{R}^N [25]. For Flyspeck, we invariably just need the special case \mathbb{R}^3 . While some concepts, e.g. vector cross products, are specific to \mathbb{R}^3 , most of the theory has been developed for general \mathbb{R}^N so as to be more widely applicable. The theorem prover HOL Light [24] is based on a logic without dependent types, but we can still encode the index N as a type (roughly, an arbitrary indexing type of size N). This means that theorems about specific sizes like 3 really are just type instantiations of theorems for general N stated with polymorphic type variables. The theory contains the following:

- Basic properties of vectors in \mathbb{R}^N , linear operators and matrices, dimensions of vector subspaces and other bits of linear algebra. For example, the following is a formal statement of the theorem that a square matrix A' is a left inverse to another one A iff it is a right inverse. Note that the double use of the same type variable N constrains the theorem to square matrices:

$$\vdash \forall A:\text{real}^N{}^N \ A':\text{real}^N{}^N. (A ** A' = \text{mat } 1) \Leftrightarrow (A' ** A = \text{mat } 1)$$

- Metric and topological notions like distances, open sets, closure, compactness and paths. Some of these are very general, others are more specific to Euclidean space. Some results include the Heine-Borel theorem, the Banach fixed-point theorem and Brouwer's fixed-point theorem. The following is a formal statement that continuous functions preserve connectedness.

$$\vdash \forall f:\text{real}^M \rightarrow \text{real}^N \ s. f \text{ continuous_on } s \wedge \text{connected } s \Rightarrow \text{connected}(\text{IMAGE } f \ s)$$

- Properties of convex sets, convex hulls, cones etc. Results include Helly's theorem, Carathéodory's theorem, and various classic results about separating and supporting hyperplanes. The following states a simpler but not entirely trivial result that convex hulls preserve compactness.

`|- $\forall s:\text{real}^N \rightarrow \text{bool}.$ compact s \Rightarrow compact(convex hull s)`

- Sequences and series of vectors and uniform convergence, Fréchet derivatives and their properties, up to various forms of the inverse function theorem, as well as specific 1-dimensional theorems like Rolle's theorem and the Mean Value Theorem. Here is the formal statement of the chain rule for Fréchet derivatives.

`|- $\forall f:\text{real}^M \rightarrow \text{real}^N$ g: $\text{real}^N \rightarrow \text{real}^P$ f' g'.`
`(f has_derivative f') (at x) \wedge`
`(g has_derivative g') (at (f x))`
`\Rightarrow ((g o f) has_derivative (g' o f')) (at x)`

3.2. Formalizing measure theory. Although the basic Euclidean theory is an important foundation, and many of the concepts like ‘convex hull’ are used extensively in the Flyspeck mathematics, perhaps the most important thing to formalize is the concept of volume.

We define integrals of general vector-valued functions over subsets of \mathbb{R}^N , using the Kurzweil-Henstock gauge integral definition. We develop all the usual properties such as additivity and the key monotone and dominated convergence theorems. We also develop a theory of *absolutely* integrable functions, where both f and $|f|$ are gauge integrable; this is known to coincide with the Lebesgue integral. Here is a formal statement of the simple theorem that integration preserves linear scaling:

`|- $\forall f:\text{real}^M \rightarrow \text{real}^N$ y s h: $\text{real}^N \rightarrow \text{real}^P$.`
`(f has_integral y) s \wedge linear h \Rightarrow ((h o f) has_integral h(y)) s`

Using this integral applied to characteristic functions, we develop a theory of (Lebesgue) measure, which of course gives volume in the 3-dimensional case. The specific notion ‘measure zero’ is formalized as `negligible`, and we also have a general notion of a set having a finite measure, and a function `measure` to return that measure when it exists. For example, this is the basic additivity theorem:

`|- $\forall s t.$ measurable s \wedge measurable t \wedge DISJOINT s t`
`\Rightarrow measure(s UNION t) = measure s + measure t`

We have proved that various ‘well-behaved’ sets such as bounded convex ones and compact ones, or more generally those with negligible frontier (boundary) are measurable, e.g.

`| - ∀s:real^N->bool. bounded s ∧ negligible(frontier s) ⇒ measurable s`

The main lack at the moment is a set of results for actually computing the measures of specific sets, as needed for Flyspeck. We can evaluate most basic 1-dimensional integrals by appealing to the Fundamental Theorem of Calculus, but we need to enhance the theory of integration with stronger Fubini-type results so that we can evaluate multiple integrals by iterated one-dimensional integrals. This work is in progress at the time of writing.

3.3. Enhanced automation. Using coordinates, many non-trivial geometric statements in \mathbb{R}^3 , or other Euclidean spaces of specific finite dimension, can be reduced purely to the elementary theory of reals. This is known to be decidable using quantifier elimination [44, 3, 28]. However, in practice this is often problematic because quantifier elimination for nonlinear formulas is inefficient. The problem is particularly severe if we want to have any kind of *formal* proof, as we do in Flyspeck, since producing such a proof induces further slowdowns [30, 33]. With this in mind, we have explored a different approach to the case of purely universally quantified formulas [26], based on ideas of Parrilo [40]. This involves reducing the initial problem to semidefinite programming, solving the SDP problem using an external tool and reconstructing a ‘sum-of-squares’ (SOS) certificate that can easily be formally checked.

For example, suppose we wish to verify that if a quadratic equation $ax^2 + bx + c = 0$ has a real root, then $b^2 \geq 4ac$. Using the SDP solver we find an algebraic certificate $b^2 - 4ac = (2ax + b)^2 - 4a(ax^2 + bx + c)$, from which the required fact follows easily: $(2ax + b)^2 \geq 0$ because it is a square, and $4a(ax^2 + bx + c) = 0$ because x is a root, and so we deduce $b^2 - 4ac \geq 0$. This method seems very useful for automating routine nonlinear reasoning in a way that is easy and quick to formally verify, so that we don’t have to rely on the correctness of a complicated program. It is even capable of solving the coordinate forms of some of the simpler Flyspeck inequalities directly, though it seems unlikely to be competitive with customized nonlinear optimization methods as described in Section 5. For example, one simple Flyspeck inequality is the following, which after being reduced to a real problem with 9 variables (three coordinates for each point) is solved by SOS in a second:

$$\|u-v\| \geq 2 \wedge \|u-w\| \geq 2 \wedge \|v-w\| \geq 2 \wedge \|u-v\| < \sqrt{8} \Rightarrow \|w-(u+v)/2\| > \|u-v\|/2$$

A quite different approach to geometric theorem proving is to work in the setting of a general real vector space or normed real vector space. In this case, other decision methods are available [43]. In particular, one of these decision procedures that we have implemented in HOL Light can sometimes handle simple forms of spatial reasoning in a purely ‘linear’ way, and so be much more efficient than the direct reduction to coordinates, even if we do in fact have a specific dimension in mind. One real example from formalizing complex analysis is the following in \mathbb{R}^2 :

$$|\|w - z\| - r| = d \wedge \|u - w\| < d/2 \wedge \|x - z\| = r \Rightarrow d/2 \leq \|x - u\|$$

4. STANDARD ML REIMPLEMENTATION OF CODE

This section describes a reimplementation of the computer code used in the proof of the Kepler conjecture. The code has been substantially redesigned to avoid various difficulties with the original implementation.

4.1. Code. The original proof of the Kepler conjecture relies significantly on computation. Computer code is used extensively and is central both to the correctness of the result and to a thorough understanding of the proof.

There are four major difficulties with understanding and verifying the original code base. The first and most glaring difficulty is simply the amount of code. At the website [10] that posts the code for the original proof there are well over 50,000 lines of programs in Java, C++, and Mathematica (among others). This represents only the calculations that Hales did himself. Samuel Ferguson also completed many of the calculations with an entirely different code base¹ of 137,000 lines of C. By contrast, the proof of the four color theorem by Robinson *et al.* [42] is less than 3,000 lines of C.

The second difficulty is in the organization of the code. The calculations were done over the space of four years and involved thousands of executions of a multitude of independent programs. Section 1.2 identifies three main computational tasks: tame graph generation, linear program bounding, and nonlinear inequality verification. Each of these main tasks consists of several subtasks. For example, verifying the inequalities required dozens of relatively complicated preprocessing phases where second derivatives of the relevant functions were bounded over fixed domains. As another example, many linear programs were solved only after a branch and bound period which were recorded in voluminous log files. In an attempt to organize the complex web of calculations, Hales devised a labeling scheme to

¹There is a large amount of code copying in Ferguson’s code, resulting in a much larger code base. The number of distinct lines is difficult to measure

uniquely identify the calculations. However, even now some computations relied upon by the proof are difficult to find in the original source code. To locate, for instance, computation “CALC-821707685” from the original proof [22, p.159], one can search on the website [10] in vain. While records of the computations were made, it is not always an easy matter to find them without guidance from the authors.

The third difficulty lies in the complexity of the implementation. For instance, the software developed to prove the inequalities upon which the original proof rests is relatively complicated. Processing power at the time (1994-1998) was just barely capable of completing the computations requested. To keep the length of execution to days or weeks instead of months or years, the code is extensively optimized. The optimizations were often implemented without comment in the source and in some cases are difficult to understand.

The final difficulty is that the original code uses C and C++ to carry out *interval arithmetic* calculations based on *floating point arithmetic*. In the process, it explicitly sets the IEEE 754 [29] rounding modes on the processor’s floating point unit. While floating point is desirable for its speed, there are difficulties with using floating point for software that requires a very high level of rigor such as that supporting mathematical proof. The first is that reasoning about floating point instructions requires a relatively deep understanding of the machine architecture [35]. For instance, setting the rounding mode changes the state of the processor itself. Such an instruction has a global effect on all subsequent floating point computations. In the original code base the rounding modes are explicitly changed at least 400 times. Moreover, compilers, libraries, and even processors are notorious for unsound implementations of the 754 standard.

4.2. Reimplementation. In 2004 we decided to reimplement the original code base. We decided that the new implementation should not require floating point numbers and rounding modes. Though speed was important, we wanted the code to be independent of any particular interval arithmetic implementation. This meant we could use a fast floating point implementation of interval arithmetic for our daily work, but could use a slower but more trustworthy implementation such as MPFI [41] to double check important computations. We also wanted to organize the new implementation such that any of the many computations upon which the proof relies could be evaluated from a single interface. This would allow Flyspeck developers to find and easily check the text of the proof during the formalization process. Finally, we wished to bring the computational aspects of the Kepler conjecture closer to the level of simplicity and clarity necessary for formalization by a proof assistant. We began this work in the spirit of Robinson *et al.* [42], which simplified the original code of Appel and Haken [1], and was

used by Gonthier to construct the fully formal proof [7] in the Coq proof assistant.

We chose Standard ML for the reimplementaion for a number of reasons. It has a formal definition [34], and thus programs have a meaning apart from the particular compiler used. It has an efficient compiler, named MLton [45]. (Our reimplementaion runs between 50% and 200% the speed of the original implementation compiled with GCC.) MLton has the ability to use external libraries written in languages other than SML with relative ease. This allowed us, from one programming environment, to control multiple linear programming solvers, interval arithmetic implementations, and nonlinear optimization packages. SML has an expressive module system, and thus it was simple to write our code with respect to an abstract type of interval arithmetic. Thus we could use multiple independent implementations with ease. As of 2008, most of the code has been completely rewritten in SML and is executable from a single command-line program. The code is freely available at the project website [31].

In the original implementation, the myriad computations were done with many different programs written in a half dozen programming languages. The results of these computations are not always easy to find or interpret. Now all the computations are executed from the same source, with organized output. In addition to giving us added confidence that the original computations were sound, we have a fairly complete suite of software support for the Flyspeck project. We are now in the process of organizing and reevaluating the thousands of computations upon which the proof depends.

5. PROVING NONLINEAR INEQUALITIES WITH BERNSTEIN BASES

The hardest computational part of the original proof of the Kepler conjecture is the verification of a list of about a thousand nonlinear inequalities. This section presents a technique aimed at proving them, based on polynomial approximation and Bernstein bases. We feel that this approach better fits the requirements of formal proof, as outlined in Section 4.2. We hope to refine the method to cover all Flyspeck inequalities.

We exhibit the method on a single inequality CALC-586468779. The original proof contains the following definitions [9]:

$$\begin{aligned} \text{pt} &:= -\frac{\pi}{3} + 4 \arctan \frac{\sqrt{2}}{5} \\ \delta_{\text{oct}} &:= \frac{\pi - 4 \arctan \frac{\sqrt{2}}{5}}{2\sqrt{2}} \\ \Delta(y) &:= \frac{1}{2} \begin{vmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & y_3^2 & y_2^2 & y_1^2 \\ 1 & y_3^2 & 0 & y_4^2 & y_5^2 \\ 1 & y_2^2 & y_4^2 & 0 & y_6^2 \\ 1 & y_1^2 & y_5^2 & y_6^2 & 0 \end{vmatrix} \\ a_0(y) &:= y_1 y_2 y_3 + \frac{1}{2}(y_1^2 y_2 + y_1 y_2^2 + y_1^2 y_3 + y_2^2 y_3 + y_1 y_3^2 \\ &\quad + y_2 y_3^2 - y_1 y_4^2 - y_2 y_5^2 - y_3 y_6^2) \\ a_1(y) &:= a_0(y_1, y_5, y_6, y_4, y_2, y_3) \\ a_2(y) &:= a_0(y_2, y_4, y_6, y_5, y_1, y_3) \\ a_3(y) &:= a_0(y_4, y_5, y_3, y_1, y_2, y_6) \\ \gamma(y) &:= -\frac{\delta_{\text{oct}}}{6} \sqrt{\Delta(y)} + \frac{2}{3} \sum_{i=0}^3 \arctan \frac{\sqrt{\Delta(y)}}{a_i(y)} \end{aligned}$$

The statement of the inequality is:

$$(1) \quad \forall y \in [2, 2.51]^6. \gamma(y) \leq \text{pt}$$

Define the difference of two intervals by $[a_1, b_1] - [a_2, b_2] = [a_1 - b_2, b_1 - a_2]$. Interval arithmetic, used to prove the inequalities in the original proof, suffers from the *dependency problem*: the minimum and maximum of the formula $x - x$ are overestimated because $[a, b] - [a, b] = [a - b, b - a]$, although $x - x$ is clearly 0. Subdividing $[a, b]$ into $[a, \frac{a+b}{2}]$ and $[\frac{a+b}{2}, b]$, and then re-evaluating the formula yields an improved result. However, depending on the problem, the number of required subdivisions can be very large. This is why checking some inequalities takes a very long time.

5.1. From geometrical functions to polynomials. Fortunately, better methods than interval arithmetic are available, if the function under consideration is polynomial. A quick look at γ tells us that (1) is not polynomial, since it has occurrences of $\sqrt{\cdot}$, $1/\cdot$, and \arctan . Can it nevertheless be reduced to a polynomial problem? Two strategies come to mind:

First, algebraic laws such as $\sqrt{a} \leq b \Leftrightarrow a \leq b^2$ (if $b \geq 0$) and $\frac{a}{b} \leq c \Leftrightarrow a \leq bc$ (if $b > 0$) can often be used to eliminate occurrences of $\sqrt{\cdot}$ and $1/\cdot$. The list of trigonometric identities is endless. For our example, Vega's rule

$\arctan a + \arctan b = \arctan \frac{a+b}{1-ab}$ seems useful. Unfortunately, this technique quite often yields huge expressions that are difficult to deal with by virtue of their sheer size. Also, an algebraic transformation to a polynomial problem may simply be impossible (we suspect that this is the case for (1)).

A second technique is based on replacing γ with a polynomial g that dominates it, but is still smaller than pt . Clearly, if there exists a g such that

$$(2) \quad \forall y \in [2, 2.51]^6. \gamma(y) \leq g(y)$$

and

$$(3) \quad \forall y \in [2, 2.51]^6. g(y) \leq \text{pt},$$

then by transitivity (1) holds.

Such a polynomial g can be obtained by replacing $\sqrt{\cdot}$, $1/\cdot$ and \arctan with polynomial approximations. We only need to ensure that we use upper approximations for positive occurrences and lower approximations for negative ones. Only occurrences whose arguments contain variables need to be replaced, since e.g. $\sqrt{2}$ is a (constant) polynomial itself.

In the definition of γ the function \arctan occurs positively, so it is replaced by an upper approximation $\overline{\arctan}$. The term $\frac{\sqrt{\Delta(y)}}{a_i(y)}$ is first unfolded to $\sqrt{\Delta(y)} \cdot \frac{1}{a_i(y)}$. Both the square root and reciprocal occur positively again here, so they can be replaced by upper approximations $\overline{\text{sqrt}}$ and $\overline{\text{rcp}}$, respectively. This yields $\overline{\arctan}(\overline{\text{sqrt}}(\Delta(y)) \cdot \overline{\text{rcp}}(a_i(y)))$ in all four summands. There remains only $\sqrt{\cdot}$ occurring negatively after $-\frac{\delta_{\text{oct}}}{6}$, which is to be replaced by a lower approximation $\underline{\text{sqrt}}$.

We choose the following approximations:

$$\begin{aligned} \overline{\arctan}(t) &:= \arctan \frac{\sqrt{2}}{5} + \frac{25}{27} \left(t - \frac{\sqrt{2}}{5} \right) \\ \overline{\text{rcp}}(t) &:= \frac{1}{4} - \frac{37t}{1600} + \frac{t^2}{1000} - \frac{13t^3}{640000} + \frac{t^4}{6400000} \\ \underline{\text{sqrt}}(t) &:= 8\sqrt{2} + \frac{3}{64(\pi - 4 \arctan \frac{\sqrt{2}}{5})} (t - 128) \\ \overline{\text{sqrt}}(t) &:= 8\sqrt{2} + \frac{1}{16\sqrt{2}} (t - 128) \end{aligned}$$

These approximations are valid on the domain (1). For example,

$$\forall t \in \Delta([2, 2.51]^6). \sqrt{t} \leq \overline{\text{sqrt}}(t).$$

This can be established by elementary means, knowing that $\Delta([2, 2.51]^6) \subseteq [128; 501]$. The latter can be shown automatically by the method outlined in the next subsection.

In summary, we arrive at the following definition of g :

$$g(y) := -\frac{\delta_{\text{oct}}}{6} \sqrt{\Delta(y)} + \frac{2}{3} \sum_{i=0}^3 \overline{\arctan(\sqrt{\Delta(y)} \cdot \overline{\text{rcp}}(a_i(y)))}$$

In form, it is almost identical to the definition of γ . Our construction of g therefore ensures (2). Moreover, the approximations were chosen (using polynomial interpolation) in a way such that

$$\gamma(2, 2, 2, 2, 2, 2) = g(2, 2, 2, 2, 2, 2) = \text{pt.}$$

This is important, because otherwise (3) cannot hold.

5.2. Bounding polynomials. In order to prove (1), it remains to be shown that $g(y) \leq \text{pt}$. This can be done with the help of Bernstein polynomials. We briefly outline the case of a single variable x here.

The i th Bernstein basis polynomial of order k is defined as

$$B_i^k(x) := \binom{k}{i} x^i (1-x)^{k-i}.$$

For a polynomial p and a vector $b \in \mathbb{R}^k$, if

$$p(x) = \sum_{i=0}^k b_i \cdot B_i^k(x)$$

then b is called the *Bernstein representation* of p . In this case

$$\forall x \in [0; 1]. p(x) \leq \max_i b_i.$$

This property is tremendously useful: it gives us an upper bound on p , namely the largest coefficient of p 's Bernstein representation. By a change of variable we can reduce any interval to $[0, 1]$. The generalization to the multivariate case is straightforward [6, 48].

In order to bound a polynomial it thus suffices to convert it into Bernstein representation. This can be done by a matrix multiplication (the Bernstein basis of order k forms a basis of the vector space of all polynomials of degree up to k). For practical purposes it is however crucial to use a more efficient algorithm (cf. [6, 48]).

Note that g contains irrational coefficients. This is a consequence of requirement (5.1) and cannot be avoided. However, we were able to choose

the approximation polynomials in a way such that the transcendental parts can be factored out (hence the occurrence of $\pi - 4 \arctan \frac{\sqrt{2}}{5}$ in the definition of `sqrt`). As can be easily checked with symbolic algebra software, the polynomial $p(y) := \sqrt{2}(g(y) - pt)$ has rational coefficients! It can thus be converted to a Bernstein representation without rounding, using the algorithm presented in [48]. With this method the only divisions are by powers of 2, which can be efficiently represented using dyadic numbers.

The polynomial p consists of 12945 monomials and has total degree 18. A prototype implementation in Haskell returns 0 as the maximum for p in about ten minutes. Thus $\sqrt{2}(g(y) - pt) \leq 0$ and $g(y) \leq pt$.

6. TAME GRAPH ENUMERATION

Tame graphs are particular plane graphs that represent potential counterexamples to the Kepler conjecture. The *Archive* is a list of over 5000 plane graphs. The original proof generates the Archive with the help of a Java program that enumerates all plane graphs. Tameness is defined in Section 18 and the enumeration is sketched in Section 19 of [22]. This section sketches the formally machine-checked proof of Claim 3.13 and Theorem 19.1 in the original proof [22]:

Theorem 1. *Any tame plane graph is isomorphic to a graph in the Archive.*

There are two potential sources of errors in the original proof: the publications only sketch the details of the enumeration, and the referees only made a passing glance at the implementation, consisting of more than 2000 lines of Java.

We recast the Java program for the enumeration of all tame graphs in logic, proved its completeness with the help of an interactive theorem prover, ran it, and compared the output to the Archive. It turns out that the original proof was right, the Archive is complete, although redundant (there are at most 2771 tame graphs). Doing all this inside a logic and a theorem prover requires two things:

- The logic must contain a programming language. We used Church's *higher-order logic* (HOL) based on λ -calculus, the foundation of functional programming. Programs in HOL are simply sets of recursion equations, i.e. pure logic.
- The programming language contained in the logic must be efficiently executable and such executions must count as proofs. The theorem prover that we used, Isabelle/HOL [37] fulfills this criterion. If all

functions that appear in a term are either data, e.g. numbers, or functions defined by recursion equations, Isabelle/HOL offers the possibility to evaluate this term t in one big and relatively efficient step to a value v , giving rise to the theorem $t = v$.

The enumeration of all tame graphs generates 23 million plane graphs — hence the need to perform massive computations in reasonable time.

Now we give a top-level overview of the formalization and proof of completeness of the enumeration of tame graphs in HOL. For details see [36]. The complete machine-checked proof, over 17000 lines, is available online in the Archive of Formal Proofs at afp.sf.net [2].

6.1. Plane graphs. Following the original proof, we represent finite, undirected, plane graphs as lists (= finite sets) of faces and faces as lists of vertices. Note that by representing faces as lists they have an orientation. The enumeration of plane graphs requires an additional distinction between *final* and *non-final* faces. Hence a face is really a pair of a list of vertices and a Boolean. A plane graph is *final* iff each of its faces is. In final graphs we can ignore the Boolean component of the faces.

6.2. Enumeration of plane graphs. The original proof characterizes plane graphs by an executable enumeration and sketches a proof of completeness of this enumeration. We have followed the original proof and taken this enumeration as the definition of planarity. The enumeration of plane graphs in the original proof proceeds inductively: you start with a seed graph with two faces, the final outer one and the (reverse) non-final inner one. If a graph contains a non-final face, it can be subdivided into a final face and any number of non-final ones. Because a face can be subdivided in many ways, this process defines a tree of graphs. By construction the leaves must be final graphs, and they are the plane graphs we are interested in: any plane graph of n faces can be generated in $n - 1$ steps by this process, adding one (final) face at a time. For details see [22] or [36].

The enumeration is parameterized by a natural number p which controls the maximal size of final faces in the generated graphs. The seed graph $Seed_p$ contains two $(p+3)$ -gons and the final face created in each step may at most be a $(p+3)$ -gon. As a result, different parameters lead to disjoint sets of graphs.

The HOL formalization defines an executable function $next-plane_p$ that maps a graph to a list of graphs, the successor graphs reachable by subdividing

one non-final face. The plane graphs are the final graphs reachable from $Seed_p$ via $next-plane_p$ for some p .

6.3. Enumeration of tame graphs. The definition of tameness in the original proof is already quite close to a direct logical formulation. Hence the HOL formalization is very close to this. Of course pictures of graphs had to be translated into formulae, taking implicit symmetries in pictures into account. We found one simplification: in the definition of an admissible weight assignment one can drop condition 3 (a condition on the 4-circuits in graphs) without changing the set of tame graphs. What facilitated our work considerably was that a number of the eight tameness conditions in the original proof are directly executable. The details are described elsewhere [36].

The enumeration of tame graphs is a modified enumeration of plane graphs where we remove final graphs that are definitely not tame, and prune the search tree at non-final graphs that cannot lead to tame graphs anymore. The published description [22] is deliberately sketchy, and the precise formulation of the pruning criteria is based on the original Java programs. This is the most delicate part of the proof because we need to balance effectiveness of pruning with simplicity of the completeness proof: weak pruning criteria are easy to justify but lead to unacceptable run times of the enumeration, sophisticated pruning techniques are difficult to justify formally. Since computer-assisted proofs are still very laborious, simplifying those proofs was of prime importance. In the end, the HOL formalization defines a function $next-tame_p$ from a graph to a graph list. It computes the list of plane successor graphs $next-plane_p g$ and post-processes it as follows:

- (1) Remove all graphs from the list that cannot lead to tame graphs because of lower bound estimates for the total admissible weight of the final graph.
- (2) Finalize all triangles in all of the graphs in the list (because every 3-cycle in a tame graph must be a face).
- (3) Remove final graphs that are not tame from the list.

A necessary but possibly not sufficient check for tameness is used in the last step. Hence the enumeration may actually produce non-tame graphs. This is unproblematic: in the worst case a fake counterexample to the Kepler conjecture is produced, but we do not miss any real ones.

Although we have roughly followed the procedure of the original proof, we have simplified it in many places. In particular we removed the special treatment of $Seed_0$ and $Seed_1$, which is a fairly intricate optimization that turned out to be unnecessary.

The following completeness theorem is the key result:

Theorem 2. *If a tame and final graph g is reachable from $Seed_p$ via $next-plane_p$ then g is also reachable from $Seed_p$ via $next-tame_p$.*

Each step $next-tame_p$ is executable and an exhaustive enumeration of all graphs reachable from a seed graph is easily defined on top of it. We call this function $tameEnum_p$. By definition, tame graphs may contain only triangles up to octagons, which corresponds to the parameters $p = 0, \dots, 5$.

6.4. Archive. In order to build on the above enumeration of all tame graphs without having to rerun the enumeration, the results of running $tameEnum_p$ with $p = 0, \dots, 5$ are put into an Archive and isomorphic graphs are eliminated. This results in 2771 graphs, as opposed to 5128 in the original proof. The reasons are twofold: there are many isomorphic copies of graphs in the Archive and it contains a number of non-tame graphs, partly because, for efficiency reasons, the original proof did not enforce all tameness conditions in its Java program. The new reduced Archive is also available online [2].

Finally we can prove Theorem 1: if g is tame plane graph, Theorem 2 and the definition of $tameEnum$ tell us that g must be contained in $tameEnum_p$ for some $p = 0, \dots, 5$. Hence it suffices to enumerate $tameEnum_p$, $p = 0, \dots, 5$, and check that, modulo graph isomorphism, the result is the same as the Archive. This is a proposition that can be proved by executing it (because the HOL formalization also includes a verified executable test for graph isomorphism which we do not discuss).

7. VERIFYING LINEAR PROGRAMS

This section reports on the current state of the formal verification of the linear programming part of the proof of the Kepler conjecture. The results of the linear programming in the original proof are recorded in several gigabytes of log files. This section presents the formalization of the generation and bounding of these linear programs in the mechanical proof assistant Isabelle [37]. A more detailed version of the material presented here can be found in [38].

This formalization relies on the archive of tame graphs. Each tame graph (except the graphs associated with the face-centered cubic and hexagonal close packings) represents a potential counterexample to the Kepler conjecture. Each potential counterexample obeys certain constraints. The original proof refutes each potential counterexample by building a linear program from the constraints and showing that these linear programs are infeasible.

A structure, which we call a *graph system*, makes this precise in a formal environment. An instance of a graph system is a tame graph which obeys the constraints listed in the definition of a graph system. The current formalization does not use all of the constraints of the original proof but only those that do not require branch and bound strategies. Our current notion of graph system can be viewed as a detailed formalization of what is called *basic linear programs* in [22, §23.3]. Because the current formalization does not capture not all constraints of the original proof, we cannot hope to refute all potential counterexamples. Nevertheless we manage to refute most of them.

We represent tame graphs as *hypermaps*. As mentioned in Section 2, a hypermap is just a finite set D of *darts* together with three permutations on D : the edge, the face, and the node permutation that compose to the identity $e \circ n \circ f = I$. This representation greatly simplifies the axiomatization of a graph system. For a detailed description of how we represent hypermaps and for a complete list of all of the axioms of a graph system see [38, §4].

Figure 1 summarizes how we generate and solve the linear programs. We apply the axioms of a graph system to each tame graph. This results in a large Isabelle theorem which is a conjunction of linear equalities and inequalities. We then normalize this conjunction to bring it into the form of a matrix inequality

$$(4) \quad Ax \leq b.$$

The entries of A and b are symbolic expressions which contain various real constants (such as π) that are needed to define the axioms of a graph system. In order to apply linear programming, we need to replace A and b with numerical approximations. We achieve this via a formalization of interval arithmetic in Isabelle, and arrive at numerical matrices A' , A'' and b' for which we have the formally proven Isabelle theorems

$$(5) \quad A' \leq A \leq A'', \quad b \leq b'$$

We then apply a simple preprocessing step which gives us formally proven a-priori bounds x' and x'' for x such that

$$(6) \quad x' \leq x \leq x''.$$

It is then possible to obtain a certificate from an external linear programming solver like GLPK (Gnu Linear Programming Kit) that allows us to formally reach a contradiction from $Ax \leq b$. The beauty of a certificate is that we can use results obtained from an untrusted source (in our case this untrusted source is a heavily optimized linear program solver programmed in C) in a trusted and completely mechanically verifiable way.

In this way, we proved the inconsistency of 2565 of the graph systems but failed to prove the inconsistency of the remaining 206. This yields a success

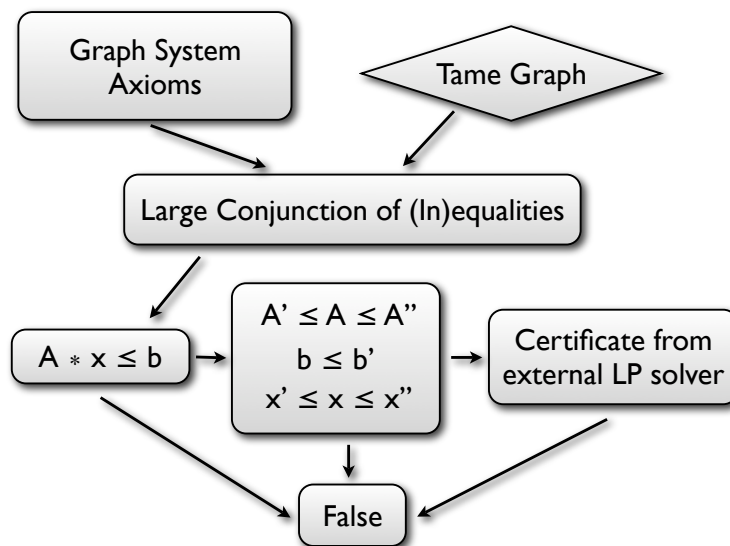


FIGURE 1. Refuting a potential counterexample to the Kepler conjecture

rate of about 92.5%. Future work will extend the notion of graph system and generate linear programs that take all the constraints of the original proof into account.

Part 2. Errata in the Original Proof

8. BICONNECTED GRAPHS

This section gives further detail to the argument of [22, §12.7 p.131]. There it is claimed that the proof of the main estimate [22, Theorem 12.1] can be reduced to polygonal standard regions. This claim is correct. However, the justification of this claim is not complete in the original proof. This section gives complete justification of the claim.

In the original proof, the boundaries of standard regions may fail to be simple polygons. In fact, the failure of the boundaries to be polygons may be quite severe: the boundaries may contain multiple components and articulation vertices. An *articulation* vertex in a graph is a vertex whose removal increases the number of connected components of the graph. A connected graph is *biconnected* if it contains no articulation vertex.

In the original proof, each standard region is first prepared by a process of erasure (described in [22, §11.1]), then deformed in a way to transform its boundary into a simple polygon. The deformation is made in such a way that the values of two key functions $\text{vor}_{0,R}$ and $\tau_{0,R}$ are left unchanged. These are the functions that enter into the main estimate [22, Theorem 12.1].

What the original proof fails to consider is a hypothetical situation such as that shown in Fig. 2, where the rigid movement of set of vertices (such as the illustrated triangle with vertex w) is blocked by a nearby vertex v that is not visible from w , when the distance between v and w drops to the minimum value 2. This section presents a proof that this hypothetical situation does not occur.

The strategy of the proof in this section is to produce a biconnected graph. In a biconnected planar graph with at least three vertices, each face is a simple polygon. Thus, by producing a biconnected graph, we achieve our objective. In the original proof of the Kepler conjecture, biconnected graphs are not mentioned by name. Nevertheless, most of the graphs that occur in the late stages of the original proof are biconnected.

In more detail, the proof in this section will produce a sequence of admissible deformations of a decomposition star D in such a way that the individual standard regions R are preserved in number and identity (but not in shape) by the deformations. The deformations will preserve the values of $\text{vor}_{0,R}(D)$ and $\tau_{0,R}(D)$, for each standard region R . The deformations will change the combinatorial structure of the graph formed by all the subregions of the decomposition star. At the conclusion of the deformation, the decomposition star D will have a graph (formed by all the subregions of the decomposition star) that is biconnected. As in the original proof, the decomposition star is first simplified by a process of erasure, before the deformations begin.

In the remainder of this section, we adopt notation, definitions, and conventions without further comment from [22].

8.1. Context. We work in the following restrictive context for Theorem 3 and its proof. We fix a packing centered at a vertex at the origin. As described in Section 12.6 of the original proof, we assume that all upright quarters are erased, except loops (that is, those surrounded by anchored simplices). Let U be the set of (non null) vertices of height at most 2.51. As usual, we say two edges $\{u_1, u_2\}$ and $\{u'_1, u'_2\}$ *cross* if the interiors of the triangles formed by $\{0, u_1, u_2\}$ and $\{0, u'_1, u'_2\}$ intersect.

We form the set of edges E' between vertices in U , consisting of

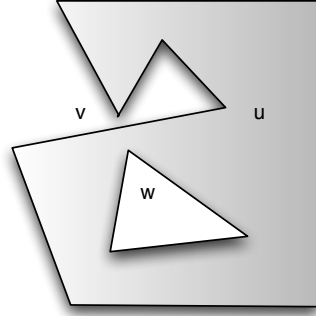


FIGURE 2. In this figure (not to scale), segments represent geodesic arcs on the unit sphere. The shaded region is a subregion whose boundary is not a simple polygon. We wish to rigidly slide the triangle containing the vertex w until a new visible distinguished edge forms (say from w to u). We will show that this rigid motion does not decrease the distance between two vertices (say w and v) to the minimum distance 2.

- all standard edges; that is, $\{v, w\} \subseteq U$ such that $0 < \|v - w\| \leq 2.51$.
- all edges $\{v, w\} \subseteq U$ of an anchored simplex, whenever the upright diagonal of the anchored simplex is an unerased loop.
- all edges $\{v, w\} \subseteq U$ such that $0 < \|v - w\| \leq \sqrt{8}$, where $\{v, w\}$ does not cross any other edge in previous two items. (If two of these edges cross, pick only one of them. This can only happen with conflicting diagonals of a quad cluster.)

These edges do not cross. A *special simplex* $\{0, u, v, w\}$ has one edge $\{v, w\}$ of length at least $\sqrt{8}$, called the *special edge*. A special edge has length at most 3.2. The other vertex u is called a special vertex (or corner). Let $E = E' \setminus S$, where S is the set of special edges (that is, the edges of special simplices shared with an anchored simplex). The projection of the line segments formed by E to the unit sphere is a planar graph. The complement of this graph in the unit sphere is a disjoint union of connected components. The closures of these connected components are called subregions.

We call a loop subregion one that contains an unerased loop $\{0, v\}$. If R is a loop subregion, then there are no enclosed vertices of height ≤ 2.51 over the subregion. The corners of R are the anchors of the upright diagonal together with the special corners of the subregion. The subregion R is star convex with center point at the projection of v to the unit sphere. It follows that the boundary of R is a simple polygon.

The graph Γ with vertices U and edges E is not necessarily connected. The aim is to deform U (and D) to create a biconnected graph (without changing the values of $\text{vor}_{0,R}(D)$ and $\tau_{0,R}(D)$ for *standard regions* R ; the values for individual subregions will change). Once the graph Γ is biconnected, the subregions are simple polygons as desired.

The deformation of U is admissible if it satisfies the following three conditions.

- If the graph Γ is not connected, the deformation acts by a rotation about the origin on the vertices of a single chosen connected component of Γ , leaving all other vertices of U fixed. (For example, in Figure 2, the entire triangle with vertex w may be rotated.) In particular, $\|v\|$, for $v \in U$, is constant.
- If the graph Γ is connected, but not biconnected, with a chosen articulation vertex a , then the deformation acts by a rotation about the axis $\{0, a\}$ on the vertices of a single component of $\Gamma \setminus \{a\}$, leaving all other vertices of U fixed.
- The distance between v, w remains at least 2, for all $v, w \in U$.

The deformation stops when either of the following halting conditions are met:

- (1) $\|v - w\| \leq \sqrt{8}$, where the edge $\{v, w\} \notin E'$, with $\{v, w\} \subseteq U$, does not cross any edge in E' , or
- (2) $\|v - w\|$ decreases to 2 for some $v, w \in U$, and $\{v, w\}$ crosses an edge in E' .

We warn that the deformation is allowed to assume configurations in which the distance between some $u \in U$ and the upper endpoint of an unerased upright diagonal d is less than 2. This is not a problem because the vertex u does not end up as a corner of the loop subregion to which the upright diagonal d belongs. (In particular, the first halting condition prevents a new special vertex from being added to a loop subregion.) Thus, when a single subregion is considered in isolation, all distances between pairs of vertices in that subregion are at least 2.

In the restrictive context that has been described in this section, we have the following result.

Theorem 3. *If the graph $\Gamma = (U, E)$ is not biconnected, then a nontrivial admissible deformation of U exists. The second halting condition never occurs. The admissible deformation can always be continued until the first halting condition holds.*

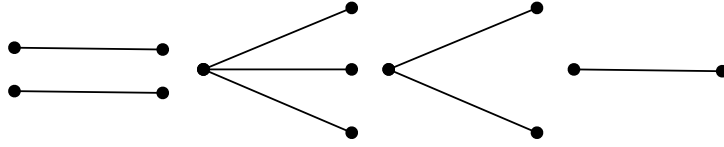


FIGURE 3. Theorem 3 breaks into various cases: two edges with distinct endpoints, three or more edges, two edges, or a single edge passing through the triangle $\{0, v, w\}$.

We break the proof of Theorem 3 into cases according to the number and combinatorial structure of the edges $\{u_1, u_2\}$ that pass through the triangle $\{0, v, w\}$. Lemma 4 shows that we cannot have two such edges $\{u_1, u_2\}$ and $\{u'_1, u'_2\}$ with distinct endpoints. It follows that there is an endpoint u_2 shared by every edge that passes through the triangle $\{0, v, w\}$. Lemma 9 shows that there cannot be three or more edges. Finally, the cases of one or two edges $\{u_1, u_2\}$ are treated in further subsections (Fig. 3). Each subsection is organized around one of the main cases (with introductory Section 8.2): the four graphs of Fig. 3 are treated in the four Sections 8.3, 8.4, 8.5, 8.6, respectively.

When U is deformed until the first halting condition holds, a new edge $\{v, w\}$ can be added to E . The theorem is repeated to add further edges, until a biconnected graph is obtained. The remainder of this section explains why the second halting condition never occurs. For this, we assume on the contrary, that $\|v - w\| = 2$, where $\{v, w\}$ belong to different connected components of Γ (if Γ is not connected) and different connected components of $\Gamma \setminus \{a\}$ (if a is an articulation vertex of a connected graph Γ). In short, we say that v, w belong to different bicomponents. We may also assume that $\{v, w\}$ crosses some edge of E' .

8.2. Preliminary lemmas. The first lemma shows that a vertex does not cross over an edge during a deformation; the halting conditions for the vertex are met before it reaches the edge. Recall that the term *geometric considerations* refers to a specific collection of methods introduced in [22, §4.2] to prove the existence and non-existence of various simple configuration of points in \mathbb{R}^3 .

Lemma 1. *Let $S = \{0, v, u_1, u_2\}$ is a set of four distinct points in \mathbb{R}^3 whose pairwise distances are at least 2. Suppose that $\|u\| \leq 2.51$, for all $u \in S$. If the segment $\{0, v\}$ meets the segment $\{u_1, u_2\}$, then $\|v - u_1\|, \|v - u_2\| \leq 2.51$.*

Proof. This follows by geometric considerations. □

Lemma 2. *Let $S = \{v, w, u_1, u_2, v_0\}$ be a set of five distinct points in \mathbb{R}^3 whose pairwise distances are at least 2. Suppose that the segment $\{v, w\}$ passes through $\{v_0, u_1, u_2\}$. Assume*

$$\begin{aligned} \|u_1 - u_2\| &\leq 3.2 \\ \|v_0 - u\| &\leq 2.51, \text{ for } u \in S \end{aligned}$$

Then $\|v - w\| > 2$.

Lemma 3. *Let $S = \{v, w, u_1, u_2, v_0\}$ be a set of five distinct points in \mathbb{R}^3 whose pairwise distances are at least 2. Suppose that the segment $\{u_1, u_2\}$ passes through $\{v_0, v, w\}$. Assume*

$$\begin{aligned} \|u_1 - u_2\| &\leq 2.91 \\ \|v_0 - u\| &\leq 2.51, \text{ for } u \in S \end{aligned}$$

Then $\|v - w\| > 2$.

Proof. Both lemmas follow by geometric considerations. □

By these two lemmas, we may assume that for each edge $\{u_1, u_2\}$ that $\{v, w\}$ crosses, we have that $\{u_1, u_2\}$ passes through $\{0, v, w\}$ and that $\|u_1 - u_2\| > 2.91$. This means that the edge $\{u_1, u_2\}$ is an edge of an anchored simplex, so that the edge is special or bounds a loop subregion. This loop subregion will provide the key to the proof, because we will show that it prevents the distance between v and w from becoming 2, as assumed (Fig. 4).

Remark 1. *As an aside, we mention that this issue of biconnected graphs is a major issue only in the proof of the Kepler conjecture and not in the proof of the dodecahedral conjecture [23]. In the proof of the dodecahedral conjecture there are no loops of anchored simplices, and without loops there are no difficulties: edges of length at most $\sqrt{8}$ can be used instead of the set E' . Lemmas 2 and 3 suffice.*

8.3. Edge crossings with distinct endpoints.

Lemma 4. *There does not exist a set $S = \{0, v, w, u_1, u_2, u'_1, u'_2\}$ of seven distinct points in \mathbb{R}^3 whose pairwise distances are at least 2 and that satisfies the following conditions.*

- *The edges $\{u_1, u_2\}$ and $\{u'_1, u'_2\}$ do not cross.*
- *The edges $\{u_1, u_2\}$ and $\{u'_1, u'_2\}$ both pass through $\{0, v, w\}$.*
- *$\|u\| \leq 2.51$ for all $u \in S$.*
- *$\|v - w\| = 2$.*
- *$\|u_1 - u_2\|, \|u'_1 - u'_2\| \leq 3.2$.*
- *u_1 and u'_1 lie in the same half-space bounded by the plane $\{0, v, w\}$.*

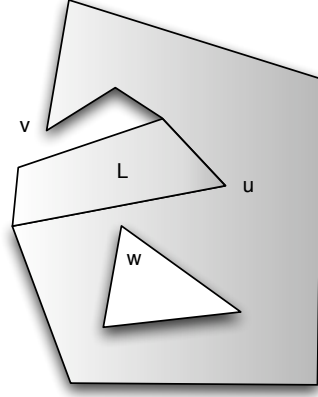


FIGURE 4. In this figure (not to scale), segments represent geodesic arcs on the unit sphere. The shaded regions represent subregions. The situation of Fig. 2 does not exist (as will be shown). There is an intervening loop subregion L that forces v and w to be more than the minimum distance apart.

- The directed segment from v to w crosses the segment $\{u'_1, u'_2\}$ before the segment $\{u_1, u_2\}$.

Note that the last two conditions can always be achieved by suitable labels on the points $\{u_1, u_2, u'_1, u'_2\}$.

Proof. This is a direct consequence of Lemmas 6 and 8. \square

Lemma 5. *Let $S = \{u, v, w\}$ be a triangle such that each side has length at least 2, and such that $\|u - v\| \leq 2.51$, $\|u - w\| \leq 2.51$, $\|v - w\| = 2$. Let X be the set of points in the convex hull of S that have distance at least 1.2 from each vertex of S . Then the diameter of X is less than 1.044.*

Proof. Let $x = \|u - v\|$ and $y = \|u - w\|$. Assume $x \geq y$. As u moves away from v , along a fixed line through v and an initial position u_0 , the region X expands. Thus, we may assume that $x = 2.51$. The boundary of X is a polygonal curve consisting of line segments and concave arcs of circles. The diameter is realized by the distance between two vertices $p_i(y)$ of the polygonal curve. We consider two cases according to $y \leq 2.4$ and $y \geq 2.4$ because the structure of the polygonal curve changes at $y = 2.4$. We calculate $\|p_i(y) - p_j(y)\|$ directly, checking for each (i, j) that the distances are less than 1.044. \square

Lemma 6. *Let $S = \{0, v, w, u_1, u_2, u'_1, u'_2\} \subseteq \mathbb{R}^3$ be a configuration of seven points that satisfies the conditions of Lemma 4. Then there exist a point p on the segment $\{u_1, u_2\}$ and a point p' on the segment $\{u'_1, u'_2\}$ such that $\|p - p'\| < 1.044$.*

Proof. Let $u \in \{0, v, w\}$. By the metric constraints, the distance from u to the segment $\{u_1, u_2\}$ (resp. $\{u'_1, u'_2\}$) is at least

$$\sqrt{2^2 - (3.2/2)^2} = 1.2.$$

Let p (resp. p') be the point of intersection of the segment $\{u_1, u_2\}$ (resp. $\{u'_1, u'_2\}$) with the convex hull of $\{0, v, w\}$. By Lemma 5, we have $\|p - p'\| < 1.044$. \square

Lemma 7. *Let $S = \{u_1, u'_1, u_2, u'_2\} \subseteq \mathbb{R}^3$ be a set of four distinct points such that the distance between each pair of points is at least 2. Assume that*

$$\|u_1 - u_2\| \leq 3.2, \quad \|u'_1 - u'_2\| \leq 3.2.$$

If any of the following conditions hold:

- (A): $\|u_1 - u'_2\|, \|u'_1 - u_2\| \geq 2.91$;
- (B): $\|u_1 - u'_1\|, \|u_2 - u'_2\| \geq 2.85, \quad \|u_1 - u'_2\| \geq 2.91$; or
- (C): $\|u_1 - u'_1\| \geq 3.64, \quad \|u_1 - u'_2\| \geq 2.91$;

then every point on the segment $\{u_1, u_2\}$ has distance greater than 1.044 from every point on the segment $\{u'_1, u'_2\}$.

Proof. Assume for a contradiction that the assumptions hold and that the conclusion is false for some configuration. The metric constraints can be used to show that the segments $\{u_1, u_2\}$ and $\{u'_1, u'_2\}$ can be stretched along their axes without decreasing any edge length. Thus we may assume without loss of generality that $\|u_1 - u_2\| = \|u'_1 - u'_2\| = 3.2$. Decreasing one dihedral angle of the simplex S at a time, we may move the segments closer together, until all four edges $\{u_i, u'_j\}$ attain their minimum length. Then we have three rigidly determined simplices (A), (B), (C) (with equality constraints). An explicit coordinate calculation of the distance between the two segments shows that the distance is greater than 1.044 in each case. \square

Lemma 8. *Let $S = \{0, v, w, u_1, u_2, u'_1, u'_2\} \subseteq \mathbb{R}^3$ be a configuration of seven points that satisfies the conditions of Lemma 4. Then there do not exist a point p on the segment $\{u_1, u_2\}$ and a point p' on the segment $\{u'_1, u'_2\}$ such that $\|p - p'\| < 1.044$.*

Proof. Assume for a contradiction that such p, p' exist. In Lemma 7, we may assume that none of the conditions A, B, C hold. By obvious symmetry,

without loss of generality, we may assume by case A that $\|u_2 - u'_1\| < 2.91$. By Lemma 3, this implies that $\{u'_1, u_2\}$ does not pass through $\{0, v, w\}$. By the conditions of the lemma, the segment $\{u_1, u'_1\}$ does not meet the plane $\{0, v, w\}$, so that $\{u_1, u'_1\}$ does not pass through $\{0, v, w\}$. This means that the triangle $\{u_1, u'_1, u_2\}$ is linked around $\{0, v, w\}$; and some edge of $\{0, v, w\}$ passes through $\{u_1, u'_1, u_2\}$. Up to symmetry there are two cases: (1) $\{0, v\}$ passes through $\{u_1, u'_1, u_2\}$, or (2) $\{v, w\}$ passes through $\{u_1, u'_1, u_2\}$.

In the first case, recall that $\{u'_1, u'_2\}$ does not cross $\{u_1, u_2\}$. So u'_2 is enclosed over $(0, \{u_1, u'_1, u_2\})$. Then $\{u'_2, u_1\}$ and $\{u'_2, u'_1\}$ pass through $\{0, v, w\}$. By Lemma 3, we have $\|u'_2 - u_1\|, \|u'_2 - u'_1\| \geq 2.91$. As we are assuming that C does not hold, we have $\|u_1 - u'_1\| \leq 3.64$. We claim that $\{0, u'_2\}$ passes through $\{u_1, u'_1, u_2\}$. Otherwise, u'_2 lies in the convex hull of $\{0, u_1, u'_1, u_2\}$ and a coordinate calculation shows that the upper bounds on the edges of the simplex $\{0, u_1, u'_1, u_2\}$ are inconsistent with the lower bounds on the distances from u'_2 to the vertices of the simplex. Since $\{0, u'_2\}$ passes through $\{u_1, u'_1, u_2\}$ we may use geometric considerations to show that $\|u'_2\| > 2.51$. This is contrary to the assumptions of Lemma 4.

In the second case, $\{v, w\}$ passes through $\{u_1, u'_1, u_2\}$ and through $\{u'_1, u'_2, u_2\}$. Geometric considerations give $\|u_1 - u'_1\| > 2.85$ and $\|u_2 - u'_2\| > 2.85$. Assuming that B does not hold gives $\|u_1 - u'_2\| < 2.91$. The first paragraph of the proof now gives that $\{u_2, u'_2, u_1\}$ links around the triangle $\{0, v, w\}$. The edge $\{v, w\}$ does not pass through $\{u_2, u'_2, u_1\}$. (This can be seen by drawing the relative positions of $p(u)$, for $u \in S$, in the projection p of the points to a plane orthogonal to $\{v, w\}$.) Thus, we are in the first case, which has already been treated. \square

8.4. Triple edge crossings. By Lemma 4, there is a common endpoint u_2 such that every edge of E' that passes through $\{0, v, w\}$ has u_2 as an endpoint. Next we show that there cannot be three such edges.

Lemma 9. *There does not exist a set of seven distinct points*

$$S = \{0, v, w, u_1, u'_1, u''_1, u_2\}$$

in \mathbb{R}^3 that satisfies the following conditions.

- *The distance between each pair of distinct points in S is at least 2.*
- *The edges $\{u_1, u_2\}$, $\{u'_1, u_2\}$, and $\{u''_1, u_2\}$ pass through $\{0, v, w\}$.*
- *$\|u\| \leq 2.51$, for all $u \in S$.*
- *$\|v - w\| = 2$.*
- *$\|u - u_2\| \leq 3.2$, for $u = u_1, u'_1, u''_1$.*

Proof. Assume S exists. We may pivot w around the axis $\{0, v\}$ until $\|w - u_2\| \leq 2.51$. (The metric constraints on edge lengths show that the condition

$\Delta > 0$ is preserved for the simplices $\{u, u_2, v, w\}$ and $\{u, u_2, w, 0\}$, for $u = u_1, u'_1, u''_1$, throughout this pivot.) A similar pivot of v gives $\|v - u_2\| \leq 2.51$. We may order the vertices in cyclic order around $\{0, u_2\}$ as

$$(w_1, w_2, w_3, w_4, w_5) = (w, u_1, u'_1, u''_1, v),$$

so that setting $d(w_i, w_j) = \text{dih}(0, u_2, w_i, w_j)$, we have

$$d(w_1, w_5) = \sum_{i=1}^4 d(w_i, w_{i+1}) \geq d(w_2, w_3) + d(w_3, w_4).$$

Interval calculations² give $d(w_2, w_3), d(w_3, w_4) \geq 0.7$ and $d(w_1, w_5) < 1.4$. We obtain an immediate contradiction:

$$1.4 > d(w_1, w_5) \geq 0.7 + 0.7.$$

□

8.5. Double edge crossings. This subsection treats the case of two edges crossings in the proof of Theorem 3. We continue to assume the general context of Theorem 3. As usual, the edge $\{u_1, u_2\} \in E'$ crosses $\{v, w\}$.

Lemma 10. *Let $\{u_1, u_2, w, v\}$ be a set of four distinct points in \mathbb{R}^3 (in the given context). Assume that $\|v - u_i\| \leq 2.51$, for $i = 1, 2$. Assume that no edge of E' crosses $\{v, w\}$ in the open half-space A containing v bounded by the plane $\{0, u_1, u_2\}$. (That is, $\{u_1, u_2\}$ is the first edge to cross $\{v, w\}$, moving from v toward w .) Assume there is a loop subregion L along $\{u_1, u_2\}$ on the A -side of $\{u_1, u_2\}$. Then $\{u_1, u_2\}$ is a special edge of E' with corner v . In particular, $\{u_1, u_2\} \notin E$, so that it is not a bounding edge of a subregion.*

Proof. Assume for a contradiction that $\{u_1, u_2\}$ is not special. Note that loop subregions have simple polygonal boundaries and remain rigid under all the deformations. In particular, the upright diagonal, special corners, and so forth remain rigidly positioned with respect to the corners of the subregion.

Since there are no further edges crossing $\{v, w\}$, the subregion L extends to v . Hence v is a corner of the subregion L . It is either an anchor or a special corner (with respect to L). However, it cannot be a special corner, by the assumption that $\{u_1, u_2\}$ is not a special edge. Hence it is an anchor. Also, u_1 and u_2 are anchors.

To reach a contradiction, we consider possible locations of the upright diagonal $\{0, u\}$, and show that it has nowhere to go (Figure 5). Since $\{u_1, u_2\}$ is an edge, the points u_1, u_2 are consecutive anchors (excluding region B). Also, the upright diagonal of any unerased loop has at least four anchors (say

²CALC-2799256461, CALC-5470795818

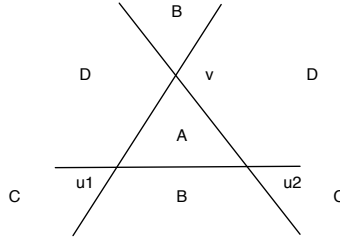


FIGURE 5. The upright diagonal u cannot be placed over any of the regions A, B, C, D . The lines (not to scale) represent geodesic arcs on the sphere passing through the pairs of points in $\{p(u_1), p(u_2), p(v)\}$, where p denotes projection to the unit sphere.

v, u_1, u_2, w). Moreover, some anchored simplex around the upright diagonal is not an upright quarter (because of the edge $\{u_1, u_2\}$). Geometric considerations based on these constraints (say $\|w - u_1\| \geq 2.51$) show that the fourth anchor w is not in A . This prevents u from being located over the region A . A vertex u_j cannot be enclosed over an upright quarter $\{0, u, v, u_i\}$. This excludes region C . Finally, an edge $\{v, u_i\}$ of length at most 2.51 cannot pass through a triangle $\{0, u, u_j\}$ of sides at most 2.51, 2.51, $\sqrt{8}$ (excluding D). \square

Lemma 11. *Let $\{u_1, u_2, w, v\}$ be a set of four distinct points in \mathbb{R}^3 (in the given context). Assume that $\|v - u_i\| \leq 2.51$, for $i = 1, 2$. Assume that no edge of E crosses $\{v, w\}$ in the open half-space A containing v bounded by the plane $\{0, u_1, u_2\}$. (That is, $\{u_1, u_2\}$ is the first edge to cross $\{v, w\}$, moving from v toward w .) Then both edges $\{u_1, v\}$ and $\{u_2, v\}$ belong to E . In particular, there is a circuit of the graph Γ through v, u_1, u_2 .*

Proof. If there is a loop subregion L along $\{u_1, u_2\}$ on the A -side of $\{u_1, u_2\}$, Lemma 10 implies that $\{u_1, u_2\}$ is a special edge of E' with corner v . In particular, $\{u_1, v\}$ and $\{u_2, v\}$ are edges of E . The conclusion follows in this case.

Now assume that there is no loop subregion along $\{u_1, u_2\}$ on the A side of $\{u_1, u_2\}$.

Let S be the finite set of points of U enclosed over the simplex $\{0, u_1, u_2, v\}$. We show by contradiction that S is empty. The plane $\{0, v, w\}$ separates S into a disjoint union $S = S_1 \cup S_2$, according to those in the same half-space as u_i , $i = 1, 2$. We form the convex hull of the projection p to the unit sphere of the points $S_i \cup \{u_i, v\}$. As in [22, §12.13], form a sequence of

geodesic arcs on the unit sphere from $p(u_i)$ to $p(v)$. Let $p(w_i)$, for $w_i \in S_i$, be the other endpoint of the arc starting at $p(u_i)$ (or set $w_i = v$ if $S_i = \emptyset$). For some $w' \in \{w_1, w_2\} \setminus \{v\}$, the edges $\{w', u_i\}$ do not cross any edges of E . Furthermore, geometric considerations show that $\|w' - u_i\| \leq 2.51$, for $i = 1, 2$. By the criteria for forming edges of E , we must have $\{w', u_i\} \in E$ for $i = 1, 2$. This contradicts the assumption that $\{v, w\}$ does not cross any edges of E over A . Hence $S = \emptyset$.

Since $S = \emptyset$, the edges $\{v, u_i\}$ do not cross any edges of E . By the criteria for forming edges of E , they belong to E . This completes the proof. \square

We are ready to prove the next major case of Theorem 3. We continue to work in the general context of that theorem, with v and w in different bicomponents of the graph Γ .

Lemma 12. *In this context, there does not exist a set of six points $\{0, v, w, u_1, u'_1, u_2\}$ where $\{u_1, u_2\}$ and $\{u'_1, u_2\}$ pass through $\{0, v, w\}$ and such that*

$$\|u_1 - u_2\| \leq 3.2, \quad \|u'_1 - u_2\| \leq 3.2.$$

Proof. We may assume that $\{u_1, u'_1\}$ are ordered so that the cyclic order around $\{0, u_2\}$ is $(w_1, w_2, w_3, w_4) = (v, u'_1, u_1, w)$. By the previous results, there are at most two edges that pass through $\{0, v, w\}$ in this manner. In particular, the part of the line segment $\{v, w\}$ between the crossings of $\{u_1, u_2\}$ and $\{u'_1, u_2\}$ lies in a single subregion L .

We claim that we do not have $\|v - u'_1\|, \|v - u_2\|, \|w - u_1\|, \|w - u_2\| \leq 2.51$. Otherwise, we break into two cases and derive a contradiction as follows. Either (1) L is a loop subregion, or (2) L is not a loop subregion, but both regions adjacent to L (along edges $\{u_1, u_2\}, \{u'_1, u_2\}$) are loop subregions. In case (1), $\{u_1, u_2, u'_1\}$ are corners of the loop subregion L . Hence, they lie on a circuit in Γ formed by the corners of that loop subregion. If $\|v - u_2\|, \|v - u'_1\| \leq 2.51$, then by Lemma 11, the points $\{v, u'_1, u_2\}$ lie on a circuit in Γ . A similar conclusion holds if corresponding inequalities hold $\|w - u_2\|, \|w - u_1\| \leq 2.51$. If all four inequalities hold, then these circuits put v, w in the same bicomponent of Γ , which is contrary to hypothesis. In case (2), then by Lemma 11, $\{v, u_2\}, \{v, u'_1\} \in E$, so $\{u'_1, u_2\}$ is a special edge and L is a loop subregion. This is contrary to the assumption of case (2). Hence we may assume by symmetry and without loss of generality that $\|v - u'_1\| \geq 2.51$ or $\|v - u_2\| \geq 2.51$.

We may stretch along the edges $\{u_2, u_1\}, \{u_2, u'_1\}$, moving u_1, u'_1 , until $\|u_2 - u_1\| = \|u_2 - u'_1\| = 3.2$. We may add inequality

$$\|u_2\| \leq 2.23,$$

for otherwise by geometric considerations $\|u_2 - u'_1\| > 3.2$. Similarly, $\|u'_1\| \leq 2.23$. If $\|u_2 - v\| \geq 2.51$, we may pivot v toward u_2 around the axis $\{0, w\}$ until $\|u_2 - v\| \leq 2.51$. Similarly, we may assume that $\|u_2 - w\| \leq 2.51$. Set $d(i, j) = \text{dih}(0, u_2, w_i, w_j)$. Then interval arithmetic calculations³ give the contradiction:

$$1.3 > d(1, 4) = d(1, 2) + d(2, 3) + d(3, 4) > 0.5 + 0.8 + 0 = 1.3.$$

□

8.6. Single edge crossings. This subsection treats the proof of Theorem 3 in the case of a single edge crossing $\{u_1, u_2\}$. This is the final case of the proof. We continue to assume the notation and general context of that theorem. In particular, v and w lie in different bicomponents of the graph Γ .

Lemma 13. *Let $\{0, u_1, u_2, v, w\}$ be a set of five distinct points such that $\{u_1, u_2\}$ is the only edge of E' that crosses $\{v, w\}$. Then $\|v - w\| > 2$.*

Proof. We assume for a contradiction that $\|v - w\| = 2$. We consider four cases depending on lengths.

Case 1: $\|u - u_i\| \leq 2.51$, for $i = 1, 2$ and $u = v, w$. By Lemma 11, there are circuits running through $\{u, u_1, u_2\}$, for $u = v, w$. This is contrary to the assumption that v, w lie in different bicomponents of the graph Γ . (In the remaining cases, there is no loss in generality to assume $\|w - u_2\| \geq 2.51$.)

Case 2: $\|w - u_2\| \geq 2.51$, $\|v - u_1\| \geq 2.51$. Geometric considerations give the contradiction $\|u_1 - u_2\| > 3.2$.

Case 3: $\|w - u_2\| \geq 2.51$, $\|v - u_2\| \geq 2.51$. Geometric considerations gives the contradiction $\|u_1 - u_2\| > 3.2$.

Case 4: $\|w - u_2\| \geq 2.51$, $\|v - u_i\| \leq 2.51$, for $i = 1, 2$. The edge $\{u_1, u_2\}$ cannot be a special edge of E' . Otherwise, v, w are corners of the same loop subregion. This contradicts the running assumption that these two vertices belong to separate bicomponents of the graph Γ . By Lemma 10, there is no loop subregion along $\{u_1, u_2\}$ on the v -side. Since $\{u_1, u_2\}$ has length greater than $\sqrt{8}$, there is a loop subregion L bounded by the edge, and it must then lie on the w -side. Thus, w is a corner of L and the circuit of Γ described by the boundary of L passes through w, u_1, u_2 . By Lemma 11, there is a circuit of Γ through v, u_1, u_2 . Hence, v, w lie in the same biconnected component, which is contrary to the running assumption. □

³CALC-7431506800, CALC-5568465464, CALC-4741571261, CALC-6915275259

9. ERRATA LISTING

The abridged version of the Kepler conjecture in the Annals [11] was generated by the same tex files as the unabridged version in [22]. For this reason, it seems that every correction to the abridged version should also be a correction to the unabridged version. We list errata in the unabridged version. The same list applies to corresponding passages in the abridged version.

Each correction gives its location in [22]. The location $\ell+n$ counts down from the top of the page, or if a section or lemma number is provided, it counts from the top of that organizational unit. The location $\ell-n$ counts up from the bottom of the page. Footnotes are not included in the count from the bottom. Every line containing text of any sort is included in the count, including displayed equations, section headings, and so forth. The material to the left of \rightsquigarrow indicates original text, and material to the right of the arrow gives replacement text. The original text and replacement text appear in italic. Comments about the corrections appear in roman. Please report further items to Hales.

In addition to the corrections to the text mentioned below, there have been some corrections to the computer code, including some typos in the listings of nonlinear inequalities. They are described in detail in [8].

9.1. Listing.

[p.47, Lemma 5.16] $Q \rightsquigarrow F$

[p.49, $\ell+2$] *supposed* \rightsquigarrow *suppose*

[p.63, Lemma 7.10] *S-system* \rightsquigarrow *Q-system*

[p.73][p.124] Some applications (such as Lemma 11.27) of Theorem 8.4 rely on the proof of the theorem, which is more general than the statement of the theorem. There are no errors or gaps here, but the wording of the theorem should have been phrased differently so that it would not have been necessary to refer to the proof. The proof is based on a finer decomposition into pieces. These finer pieces are sometimes used.

[p.75, Remark 8.11] *show* \rightsquigarrow *shows*

[p.78, $\ell-7$] *constraints* \rightsquigarrow *constraint*

[p.86, $\ell+14$] *Let $\{0, v\}$ be the diagonal of an upright quarter in the Q_0* \rightsquigarrow *Let v be a vertex with $2t_0 < |v| < \sqrt{8}$.* Section 9 assumes that the diagonal belongs to a quarter in the Q -system, but Lemma 10.14 uses these results when $\{0, v\}$ has 0 or 1 anchors. To make this coherent, we should assume throughout Section 9 that we have the weaker condition that whenever $\{0, v\}$ has two or more anchors, it is a

diagonal of a quarter in the Q -system. The proofs of Section 9 all go through in this context. (Lemma 9.7 is all that is relevant here.)

[p.87,Definition 9.3] In definition of $\Delta(v, W^e)$, we can have some Q (as in Fig 9.1) with negative orientation. In this case, $E_v \cap E_i$ can clip the other side. We want the object without clipping. $\Delta(v, W^e)$ should be understood as the unclipped object.

[p.88,Definition 9.6] The definition is poorly worded. First of all, it requires that the subscript to ϵ to be a vertex, but then in the displayed equation, it makes $w/2$ the subscript, which is not a vertex. To define ϵ' , move from $w/2$ along the ray through x' until an edge of the Voronoi cell is encountered. If v, w, u are the three vertices defining that edge, then set $\epsilon'_v(\Lambda, x) = u$. Degenerate cases, such as when two different edges are encountered at the same time, can be resolved in any consistent fashion. In [19], these degeneracies are avoided altogether, by replacing functions ϵ, ϵ' with sets Φ, Φ' .

[p.88,Lemma 9.7, $\ell+2$] w and $v \rightsquigarrow w$ and u

[p.88,L. 9.7,Claim 1] *with* $|w - w'| \leq 2t_0$, *and* \rightsquigarrow *with*

[p.88,L. 9.7, $\ell+5$] *Then:* \rightsquigarrow *Let* $R'_w = \{x \in R_w \cap (0, \{u, w\}) \mid \epsilon_0(x, \{u, w\}) = u\}$. *Assume that* R'_w *is not empty. Then:*

[p.88,L. 9.7,Claim 3] $R_w \rightsquigarrow R'_w$

[p.89, $\ell+2$] $\{w, v\} \rightsquigarrow \{w, u\}$

[p.92, $\ell+16, \ell+21$] $\max_j u_j \rightsquigarrow \max_j |u_j|$

[p.93, $\ell-4$] *obstructed from* $w \rightsquigarrow$ *obstructed from* w'

[p.93, $\ell-2$] *from some* \rightsquigarrow *for some*

[p.99, $\ell+1$] *start* \rightsquigarrow *star*

[p.105,Lemma 10.14] In the proof of the cases involving 0 or 1 anchor, a combination of the decompositions from Section 8.4 and Section 9 are used. These decompositions haven't been shown to be compatible. Instead, it is better to combine $\Delta(v, W)$ with t_0 -truncation on the rest of the quad-cluster. With a t_0 truncation, we no longer have the non-positivity results from Section 8. (The quoins give a positive contribution.) However, a routine calculation shows that the estimate on $\Delta(v, W)$ is sufficiently small so that we still obtain a constant less than -1.04 pt.

[p.116][p.121] Definition 11.7 allows masked flat in definition of 3-unconfined. Definition 11.24 requires no masked flats in the same definition. Use Definition 11.24 (no masked flats), rather than 11.7. Where masked flats occur, treat them with Lemma 11.23, parts (1) and (2).

[p.116, $\ell+1$] *Lemma 4.16* \rightsquigarrow *Lemma 4.17*

[p.117,before Lemma 11.9] *two others* \rightsquigarrow *three others*

[p.117,Def 11.8] $y_1 \rightsquigarrow y_1$

[p.119,Definition 11.5] By definition, we require a masked flat quarter to be a strict quarter.

[p.121] See p.116.

[p.121, ℓ -5] $0.2274 \rightsquigarrow 0.02274$

[p.123. flat case (2)] It is missing isolated quarters cut from the side. To fix this, in condition 2(f), $\eta_{456} \geq \sqrt{2} \rightsquigarrow \eta_{456} \geq \sqrt{2}$ or $\eta_{234} \geq \sqrt{2}$

[p.124] See p.73.

[p.126] Theorem 12.1 needs to be stated in a form that allows the application in pp.251-252 and Lemma 13.5. In these applications, the regions are smaller than standard regions. Yet in the statement of the theorem, the regions are standard regions. This is not a problem in practice, because the proof is at a much finer level of decomposition than standard regions. However, the wording should to be changed so that the theorem applies precisely.

[p.126] Theorem 12.1 should include $\sigma_R(D) \leq s_n$ with $s_3 = 1$ pt and $s_4 = 0$, and $\tau_R(D) \geq t_3 = 0$.

[p.131] Section 8 gives the deformation arguments that produce a biconnected graph.

[p.139,Lemma 12.18,proof, ℓ +3] $C_0(|v|, \pi) \rightsquigarrow C_0^u(|v|, \pi)$

[p.139,Lemma 12.18,proof, ℓ +6] $\tau_0(C_0^u(2t_0, \pi)) - \pi_{\max} \rightsquigarrow \tau_0(C_0^u(2.2, \pi)) - \pi_{\max}$

[p.144, ℓ +11, ℓ +17] $2t_0^2 \rightsquigarrow (2t_0)^2$

[p.146] $S_n^\pm \rightsquigarrow$ of 3-crowded, 3-undefined, and 4-crowded combinations

[p.148,§13.6] This entire section is misplaced. It belongs with §25.5 and §25.6.

[p.149,before 13.7] *the diagrams* \rightsquigarrow *Figs 25.1–25.4*

[p.149,p.156] The definition of δ_{loop} was accidentally dropped from the published version. Set $\delta_{loop}(4, 2) = 0.12034$ $\delta_{loop}(5, 1) = 0.24939$. These constants and their properties appear in the earlier 2002 arXiv preprint of the proof *The Kepler conjecture (Sphere Packings VI)*.

[p.156,Lemma 13.5, ℓ +4] *respectively for $\tau_R(D)$* \rightsquigarrow *respectively, for $\sigma_R(D)$ and $\tau_R(D)$*

[p.164, ℓ -1] *This shows... occur.* \rightsquigarrow *This completes the proof.*

[p.173, ℓ +4] Insert the subscript on b , as in Proposition 15.5, starting on page 173: $b \rightsquigarrow b_q$.

[p.182,Lemma 16.7] The bound of 0 has not been shown to hold on each half. This is not a direct consequence of Theorem 8.4 as claimed. This can be fixed as follows. Let v_1, v_3 be the corners giving the endpoints of the long edge of the acute triangle at 0, and let v_2, v_4 be the other two corners. If either vertex v_1, v_3 has height greater than 2.3 show that the vor_0 -scored quad cluster scores less than -1.04 pt. For this, we may use the deformations of Lemmas 12.10. The length of the diagonal along

the acute face remains fixed and at least $\sqrt{8}$. We claim these deformations produce a diagonal of length less than $\sqrt{8}$ between opposite corners of the quadrilateral. (If not, the deformations produce a rhombus with side 2 and diagonals both greater than $\sqrt{8}$, which is a geometric impossibility.) We cut the quad cluster along the diagonal of length $\sqrt{8}$ and continue with deformations until the top edges on each simplex are $(y_4, y_5, y_6) = (2, 2, \sqrt{8})$. We may apply CALC-474496219 and CALC-8990938295 to the two separate simplices to obtain the result. Now we may assume that the heights of v_1, v_3 are at most 2.3. If either height is at least 2.1, the result follows from CALC-5127197465, which gives the bound of 0 on each half.

Finally, we have the case where both heights are at most 2.1. We may apply dimension-reduction techniques so that each of the two remaining corners $v \neq v_1, v_3$ of the quad cluster either has height 2 or has distance 2 from v_1 or v_3 . We then reprove Lemma 16.8 without using the bound of 0 and Lemma 16.9 for tetrahedra without the bound -1.04 pt. This appears in CALC-1551562505 and CALC-3013446042.

If the dimension reduction drops the cross-diagonal $\{v_2, v_4\}$ all the way to $\sqrt{8}$, then we may swap diagonals and continue, until both diagonals are exactly $\sqrt{8}$. In this case, by the cases already considered, we may assume that each corner has height at most 2.1. Also, geometric considerations give that the other edges are at most 2.02:

$$\mathcal{E}(2, 2, 2, \sqrt{8}, 2.1, 2.1, 2, 2, 2.02) > \sqrt{8}.$$

The result follows in this case by CALC-4723770703.

[p.241] *Mixed* is defined so as to include the pure analytic case. In earlier articles, ‘mixed’ excludes the pure analytic. *mixed* \rightsquigarrow *mixed or pure*

[p.243, $\ell+13,\ell+14,\ell+15$] Delete three sentences: ‘Let v_{12} be ... We let ... Break the pentagon ...’

[p.248,last displayed formula] = \rightsquigarrow + so that it reads

$$\sum_i f_{R_i}(D) \leq \hat{\sigma}(Q_i) + \text{vor}_{R',0}(D) + \pi_R$$

[p.252,§25.7,Cases 2 and 3] ‘The flat quarter’ is mentioned, but there are no flat quarters that have been introduced into the context. This passage has been displaced from its original context.

[p.254, $\ell+7$] *to branch combine* \rightsquigarrow *to combine*

REFERENCES

- [1] K. Appel and W. Haken. The four color proof suffices. *The Mathematical Intelligencer*, 8(1):10–20 & 58, 1986.
- [2] G. Bauer and T. Nipkow. Flyspeck I: Tame graphs. In G. Klein, T. Nipkow, and L. Paulson, editors, *The Archive of Formal Proofs*. <http://afp.sf.net/entries/Flyspeck-Tame.shtml>, May 2006.

- [3] G. E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Automata theory and formal languages (Second GI Conf., Kaiserslautern, 1975)*, pages 134–183. Lecture Notes in Comput. Sci., Vol. 33. Springer, Berlin, 1975.
- [4] E. Denney. A prototype proof translator from HOL to Coq. In *TPHOLs '00: Proceedings of the 13th International Conference on Theorem Proving in Higher Order Logics*, pages 108–125, London, UK, 2000. Springer-Verlag.
- [5] S. P. Ferguson. Sphere packings V. *Discrete and Computational Geometry*, 36(1):1–269, 2006.
- [6] J. Garloff. Convergent bounds for the range of multivariate polynomials. In *Proceedings of the International Symposium on interval mathematics on Interval mathematics 1985*, pages 37–56, London, UK, 1985. Springer-Verlag.
- [7] G. Gonthier. Formal proof – the four colour theorem. *Notices of the AMS*, 55(11):1382–1393, December 2008.
- [8] T. C. Hales. Errata and revisions to the proof of the Kepler conjecture. <http://code.google.com/p/flyspeck/>.
- [9] T. C. Hales. Sphere packings, I. *Discrete Comput. Geom*, 17:1–51, 1997.
- [10] T. C. Hales. Kepler conjecture source code, 1998. <http://www.math.pitt.edu/~thales/kepler98/>.
- [11] T. C. Hales. A proof of the Kepler conjecture. *Annals of Mathematics*, 162:1065–1185, 2005.
- [12] T. C. Hales. Introduction to the Flyspeck project. In T. Coquand, H. Lombardi, and M.-F. Roy, editors, *Mathematics, Algorithms, Proofs*, number 05021 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2006. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany. <http://drops.dagstuhl.de/opus/volltexte/2006/432>.
- [13] T. C. Hales. Sphere packings III. *Discrete and Computational Geometry*, 36(1):1–269, 2006.
- [14] T. C. Hales. Sphere packings IV. *Discrete and Computational Geometry*, 36(1):1–269, 2006.
- [15] T. C. Hales. Sphere packings VI. *Discrete and Computational Geometry*, 36(1):1–269, 2006.
- [16] T. C. Hales. Some methods of problem solving in elementary geometry. In *LICS '07: Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science*, pages 35–40, Washington, DC, USA, 2007. IEEE Computer Society.
- [17] T. C. Hales. The Flyspeck Project, 2007. <http://code.google.com/p/flyspeck>.
- [18] T. C. Hales. Flyspeck: A blueprint for the formal proof of the Kepler conjecture. source files at <http://code.google.com/p/flyspeck/source/browse/trunk/>, 2008.
- [19] T. C. Hales. Lemmas in elementary geometry. source files at <http://code.google.com/p/flyspeck/source/browse/trunk/>, 2008.
- [20] T. C. Hales. The Jordan curve theorem, formally and informally. *American Math. Monthly*, 114(10):882–894, December 2007.
- [21] T. C. Hales. Formal proof. *Notices of the AMS*, 55(11):1370–1380, December 2008.
- [22] T. C. Hales and S. P. Ferguson. The Kepler conjecture. *Discrete and Computational Geometry*, 36(1):1–269, 2006.
- [23] T. C. Hales and S. McLaughlin. A proof of the Dodecahedral conjecture, 2008. <http://arxiv.org/abs/math/9811079>.
- [24] J. Harrison. HOL Light: A tutorial introduction. In M. Srivas and A. Camilleri, editors, *Proceedings of the First International Conference on Formal Methods in Computer-Aided Design (FMCAD'96)*, volume 1166 of *Lecture Notes in Computer Science*, pages 265–269. Springer-Verlag, 1996.

- [25] J. Harrison. A HOL theory of Euclidean space. In *Theorem proving in higher order logics*, volume 3603 of *Lecture Notes in Comput. Sci.*, pages 114–129. Springer, Berlin, 2005.
- [26] J. Harrison. Verifying nonlinear real formulas via sums of squares. In K. Schneider and J. Brandt, editors, *Proceedings of the 20th International Conference on Theorem Proving in Higher Order Logics, TPHOLs 2007*, volume 4732 of *Lecture Notes in Computer Science*, pages 102–118. Springer-Verlag, Kaiserslautern, Germany, 2007.
- [27] J. Harrison. Formal proof – theory and practice. *Notices of the AMS*, 55(11):1395–1406, December 2008.
- [28] L. Hörmander. *The analysis of linear partial differential operators. II*. Classics in Mathematics. Springer-Verlag, Berlin, 2005. Differential operators with constant coefficients, Reprint of the 1983 original.
- [29] IEEE Standards Committee 754. *IEEE Standard for binary floating-point arithmetic, ANSI/IEEE Standard 754-1985*. Institute of Electrical and Electronics Engineers, New York, 1985.
- [30] A. Mahboubi and L. Pottier. Elimination des quantificateurs sur les réels en Coq. In *Journées Francophones des Langages Applicatifs (JFLA)*, 2002. available on the Web from <http://www.lix.polytechnique.fr/~assia/Publi/jfla02.ps>.
- [31] S. McLaughlin. KeplerCode: computer resources for the Kepler and Dodecahedral Conjectures. <http://code.google.com/p/kepler-code/>.
- [32] S. McLaughlin. An interpretation of Isabelle/HOL in HOL Light. In U. Furbach and N. Shankar, editors, *IJCAR*, volume 4130 of *Lecture Notes in Computer Science*, pages 192–204. Springer, 2006.
- [33] S. McLaughlin and J. Harrison. A proof-producing decision procedure for real arithmetic. In *Automated deduction—CADE-20*, volume 3632 of *Lecture Notes in Comput. Sci.*, pages 295–314. Springer, Berlin, 2005.
- [34] R. Milner, M. Tofte, and R. Harper. *The Definition of Standard ML*. MIT Press, 1990.
- [35] D. Monniaux. The pitfalls of verifying floating-point computations. *TOPLAS*, 30(3):12, May 2008.
- [36] T. Nipkow, G. Bauer, and P. Schultze. Flyspeck I: Tame graphs. In U. Furbach and N. Shankar, editors, *Automated Reasoning (IJCAR 2006)*, volume 4130 of *Lect. Notes in Comp. Sci.*, pages 21–35. Springer-Verlag, 2006.
- [37] T. Nipkow, L. Paulson, and M. Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *Lect. Notes in Comp. Sci.* Springer-Verlag, 2002. <http://www.in.tum.de/~nipkow/LNCS2283/>.
- [38] S. Obua. *Flyspeck II: The Basic Linear Programs*. PhD thesis, Technische Universität München, 2008.
- [39] S. Obua and S. Skalberg. Importing HOL into Isabelle/HOL. In *Automated Reasoning*, volume 4130 of *Lecture Notes in Computer Science*, pages 298–302. Springer, 2006.
- [40] P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Math. Program.*, 96(2, Ser. B):293–320, 2003. Algebraic and geometric methods in discrete optimization.
- [41] N. Revol and F. Rouillier. Motivations for an arbitrary precision interval arithmetic and the MPFI library. *Reliable Computing*, 11(4):275–290, 2005.
- [42] Robertson, Sanders, Seymour, and Thomas. The four-colour theorem. *Journal of Combinatorial Theory, Series B*, 70, 1997.
- [43] R. M. Solovay, R. D. Arthan, and J. Harrison. Some new results on decidability for elementary algebra and geometry. submitted APAL, 2009.
- [44] A. Tarski. *A decision method for elementary algebra and geometry*. University of California Press, Berkeley and Los Angeles, Calif., 1951. 2nd ed.
- [45] S. Weekes. MLton. <http://mlton.org>.

- [46] F. Wiedijk. Encoding the HOL Light logic in Coq. <http://www.cs.ru.nl/~freek/notes/holl2coq.pdf>.
- [47] F. Wiedijk, editor. *The seventeen provers of the world*, volume 3600 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 2006. Foreword by Dana S. Scott, Lecture Notes in Artificial Intelligence.
- [48] R. Zumkeller. *Global Optimization in Type Theory*. PhD thesis, École Polytechnique Paris, 2008.

filename: revision.tex

PDF generated from LaTeX sources on December 29, 2008;

Repository Root: <https://flyspeck.googlecode.com/svn>

SVN *LastChangedRevision* : 721

Research supported by NSF grant 0804189.

(T. Hales) MATH DEPARTMENT, UNIVERSITY OF PITTSBURGH

E-mail address: hales@pitt.edu

(J. Harrison) INTEL CORPORATION, JF1-13, 2111 NE 25TH AVENUE HILLSBORO, OR 97124 USA

E-mail address: johnh@ichips.intel.com

(S. McLaughlin) CARNEGIE MELLON UNIVERSITY

E-mail address: seanmcl@gmail.com

(T. Nipkow) DEPARTMENT FOR INFORMATICS, TECHNISCHE UNIVERSITÄT MÜNCHEN

E-mail address: www.in.tum.de/~nipkow

(S. Obua) DEPARTMENT FOR INFORMATICS, TECHNISCHE UNIVERSITÄT MÜNCHEN

(R. Zumkeller) ÉCOLE POLYTECHNIQUE, PARIS