Turing

"And when it comes to mathematics, you must realize that this is the human mind at the extreme limit of its capacity." (H. Robbins)

"…so reduce the use of the brain and calculate!" (E. W. Dijkstra)

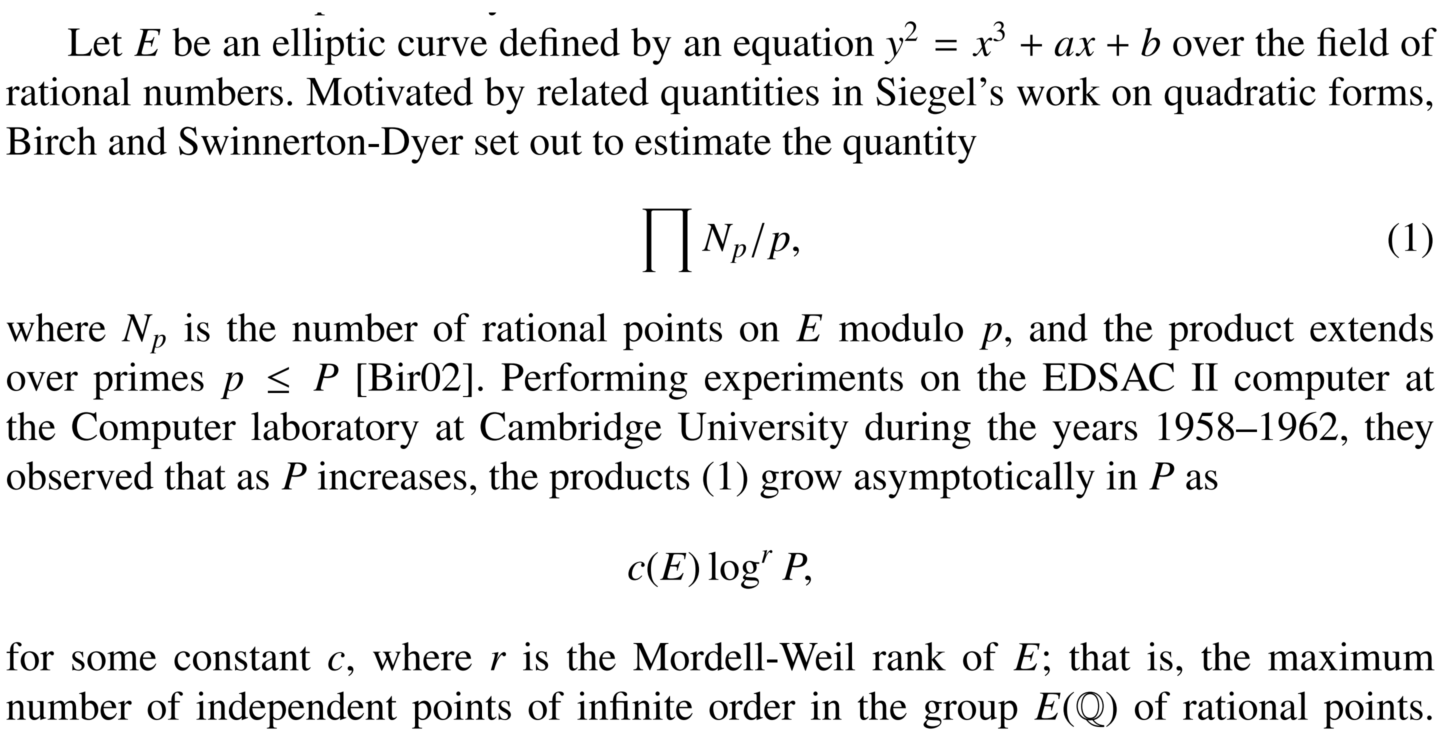
"The fact that a brain can do it seems to suggest that the difficulties [of trying with a machine] may not really be so bad as they now seem." (A. Turing)

Mathematics in the Age of the Turing Machine

Thomas C. Hales

September 19, 2011

Birch and Swinnerton-Dyer



Let E be an elliptic curve defined by an equation $y^2 = x^3 + ax + b$ over the field of rational numbers. Motivated by related quantities in Siegel's work on quadratic forms, Birch and Swinnerton-Dyer set out to estimate the quantity

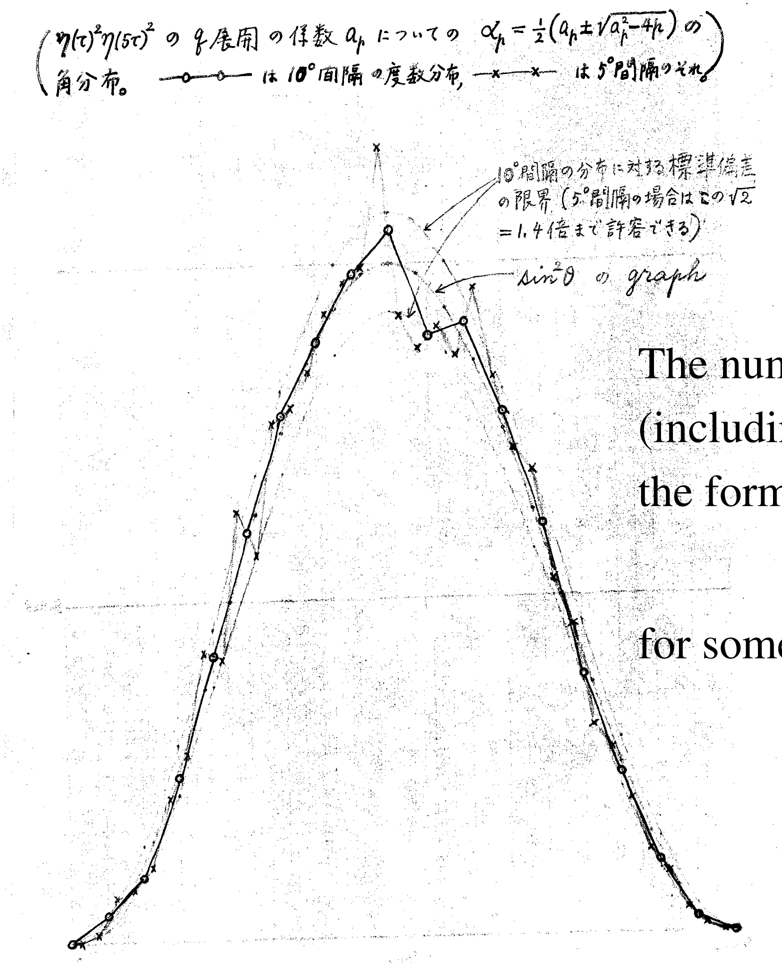
$$\prod N_p/p, \tag{1}$$

where N_p is the number of rational points on E modulo p, and the product extends over primes $p \leq P$ [Bir02]. Performing experiments on the EDSAC II computer at the Computer laboratory at Cambridge University during the years 1958–1962, they observed that as P increases, the products (1) grow asymptotically in P as

$$c(E) \log^r P,$$

for some constant c, where r is the Mordell-Weil rank of E; that is, the maximum number of independent points of infinite order in the group $E(\mathbb{Q})$ of rational points.

Sato-Nagashima-Namba (1962)



The number of points N_p modulo a prime number p (including the point at infinity) on an elliptic curve over $\mathbb{Q}$ has the form

$$1 + p - 2\sqrt{p}\cos\theta_p.$$

for some real number $0 \leq \theta_p \leq \pi$.

Euler conjectured (1769) that a fourth power cannot be the sum of three positive fourth powers, that a fifth power cannot be the sum of four positive fifth powers, and so forth. In 1966, a computer search [LP66] on a CDC 6600 mainframe uncovered a counterexample

$$27^5 + 84^5 + 110^5 + 133^5 = 144^5,$$

which can be checked by hand (I dare you). The two-sentence announcement of this counterexample qualifies as one of the shortest mathematical publications of all times. Twenty years later, a more subtle computer search gave another counterexample [Elk88]:

$$2682440^4 + 15365639^4 + 18796760^4 = 20615673^4.$$

$$2^8 \cdot 3^7 \cdot 5^6 \cdot 7 \cdot 11 \cdot 31 \cdot 37 \cdot 67.$$

In 1973, Sims proved the existence of this group in a long unpublished manuscript that relied on many specialized computer programs. By 1999 , the calculations had become standardized in group theory packages, such as GAP and Magma [HS99]. Eventually, computer-free existence and uniqueness proofs were found [MC02], [AS92].

The Catalan conjecture (1844) asserts that the only solution to the equation

$$x^m - y^n = 1,$$

in positive integers x, y, m, n with exponents m, n greater than 1 is the obvious

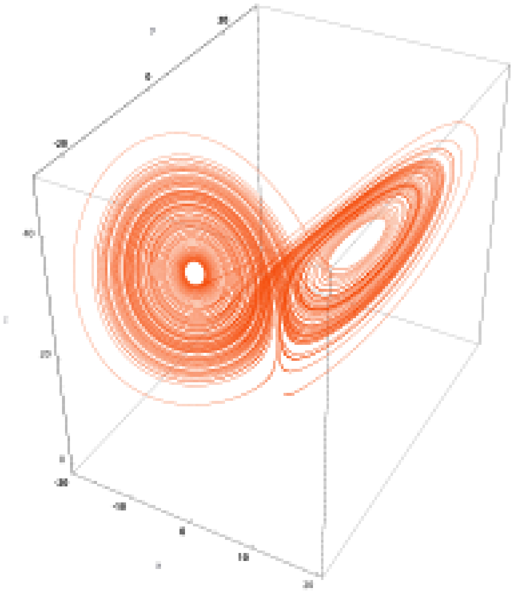
$$3^2 - 2^3 = 1.$$

That is, 8 and 9 are the only consecutive positive perfect powers. By the late 1970s, Baker's methods in diophantine analysis had reduced the problem to an astronomically large and hopelessly infeasible finite computer search. Mihăilescu's proof (2002) of the Catalan conjecture made light use of computers (a one-minute calculation), and later the computer calculations were entirely eliminated [Mih04], [Met03].

Bailey, Borwein, and Plouffe found an algorithm for calculating the *n*th binary digit of π directly: it jumps straight to the *n*th digit without first calculating any of the earlier digits. They understood that to design such an algorithm, they would need an infinite series for π in which powers of 2 controlled the denominators. They did not know of any such formula, and made a computer search (using the PSLQ lattice reduction algorithm) for any series of the desired form. Their search unearthed a numerical identity

$$\pi = \sum_{n=0}^{\infty} \left(\frac{4}{8n+1} - \frac{2}{8n+4} - \frac{1}{8n+5} - \frac{1}{8n+6} \right) \left(\frac{1}{16} \right)^{n},$$

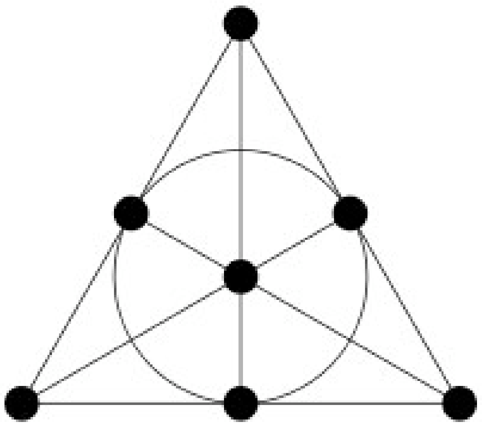
Lorenz attractor



Lorenz (1963) encountered chaos as he ran weather simulations on a Royal McBee LGP-30 computer. Tucker has solved Smale’s fourteenth problem (strange attractors in the Lorenz equations) by computer, recognized by the Moore Prize (2002) and the EMS Prize (2004)

A finite projective plane of order $n > 1$ is defined to be a set of $n^2 + n + 1$ lines and $n^2 + n + 1$ points with the following properties:

1. Every line contains $n + 1$ points;
2. Every point is on $n + 1$ lines;
3. Every two distinct lines have exactly one point of intersection;
4. Every two distinct points lie on exactly one line.

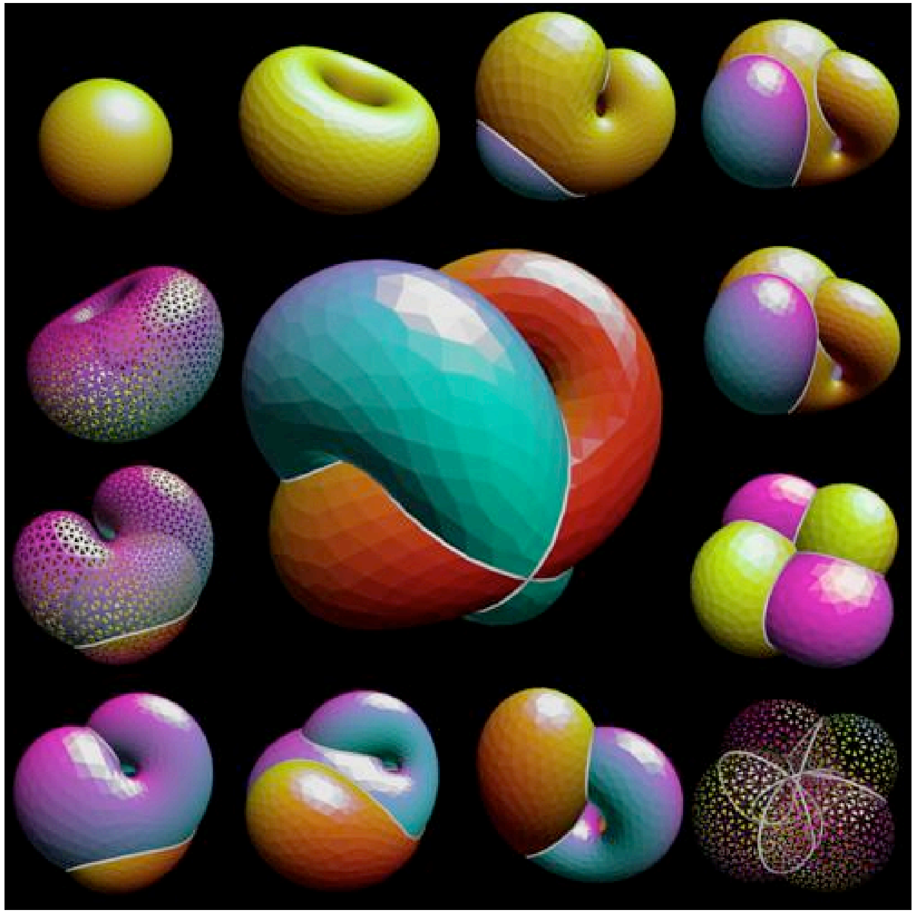


The smallest integers $n > 1$ that are *not* prime powers are

$$6, \ 10, \ 12, \ 14, \ 15, \ \ldots$$

The brute force approach to this conjecture is to eliminate each of these possibilities in turn. The case $n = 6$ was settled in 1938. Building on a number of theoretical advances [MST73], Lam eliminated the case $n = 10$ in 1989, in one of the most difficult computer proofs in history [LTS89]. This calculation was executed over a period of years on multiple machines and eventually totaled about 2000 hours of Cray-1A time.

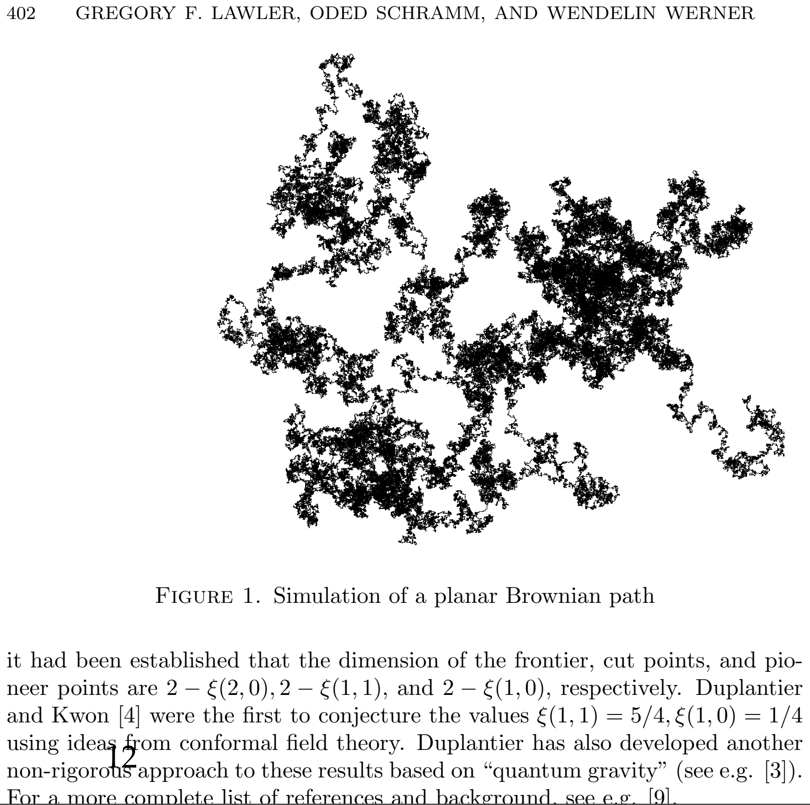
Sphere eversion



Mandelbrot’s 4/3 conjecture

Fejes Toth at Fields

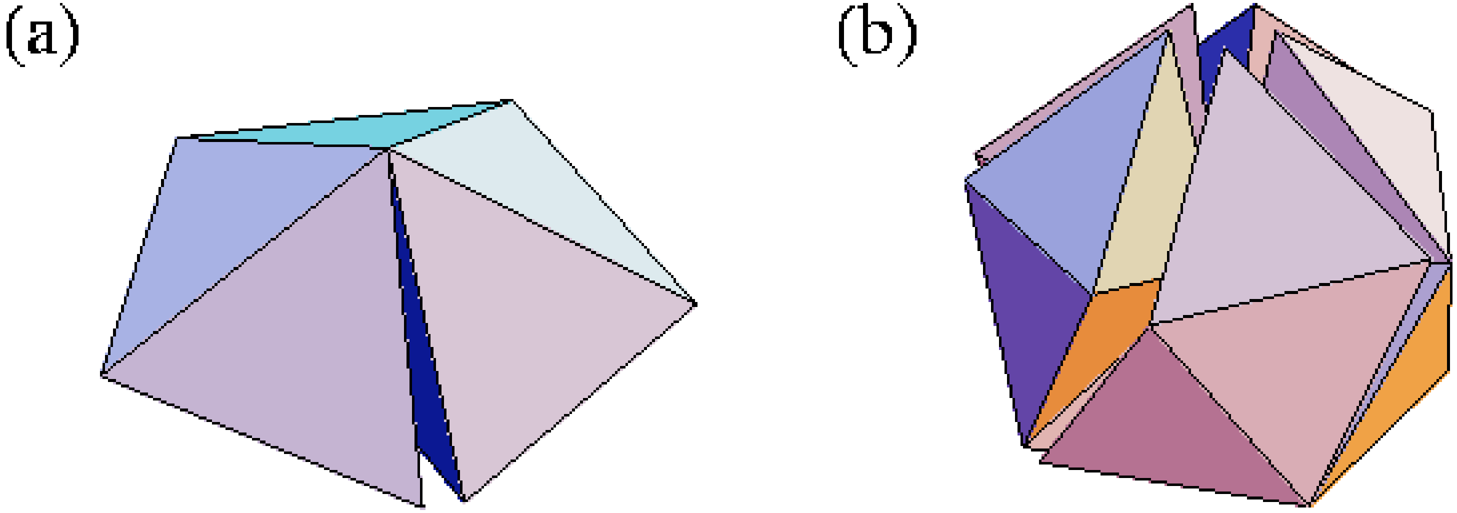
“The notion that these conjectures might have been reached by pure thought – with no picture – is simply inconceivable.. . . I had my programmer draw a very big sample [Brownian] motion and proceeded to play with it.” – Mandelbrot



402 GREGORY F. LAWLER, ODED SCHRAMM, AND WENDELIN WERNER

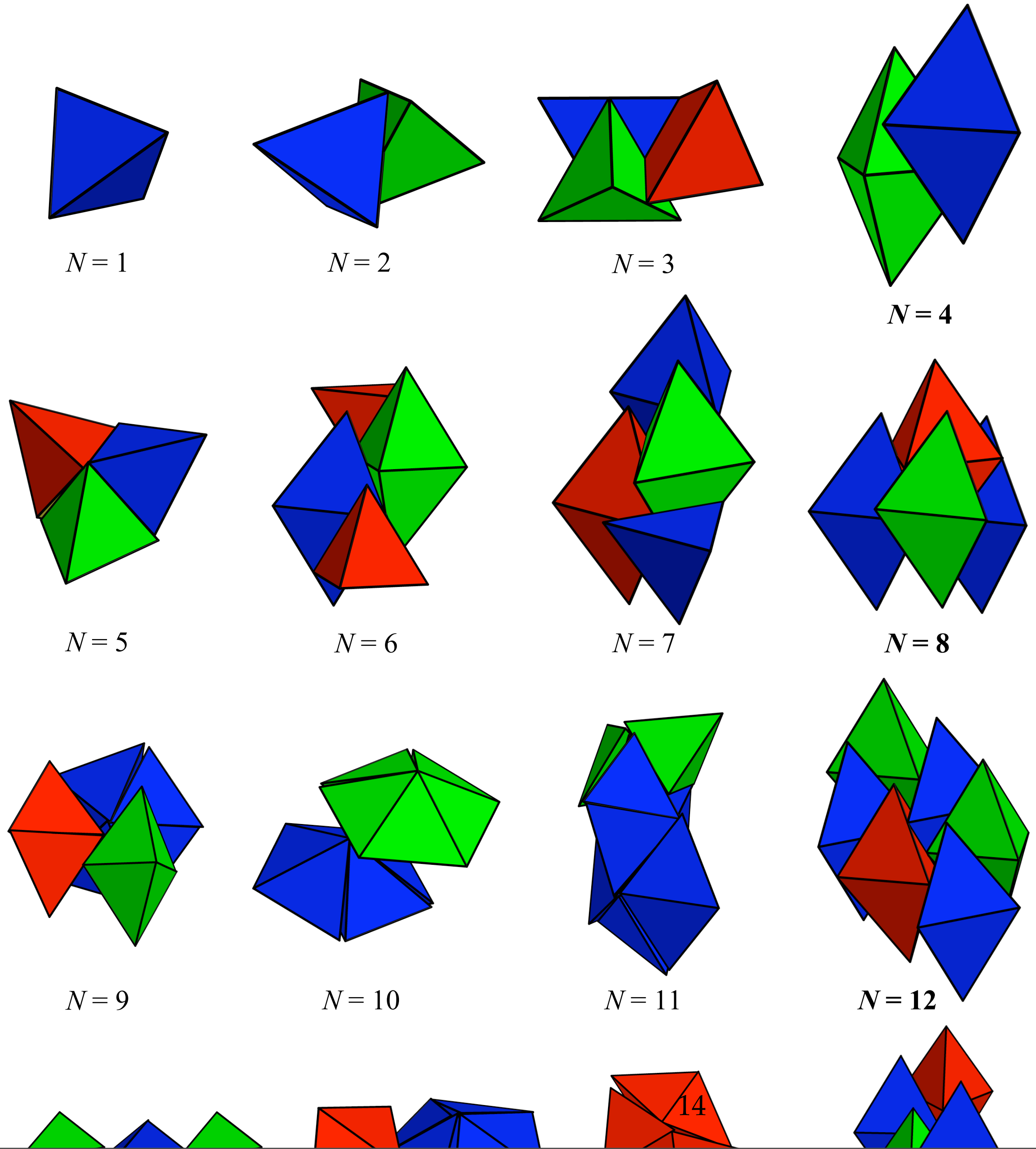
FIGURE 1. Simulation of a planar Brownian path

it had been established that the dimension of the frontier, cut points, and pioneer points are $2 - \xi(2,0), 2 - \xi(1,1)$, and $2 - \xi(1,0)$, respectively. Duplantier and Kwon [4] were the first to conjecture the values $\xi(1,1) = 5/4, \xi(1,0) = 1/4$ using ideas from conformal field theory. Duplantier has also developed another non-rigorous approach to these results based on “quantum gravity” (see e.g. [3]). For a more complete list of references and background, see e.g. [9].


(a)
(b)

Chen-Engel-Glotzer packings

Fejes Toth at Fields



$N = 1$ $N = 2$ $N = 3$ **$N = 4$**

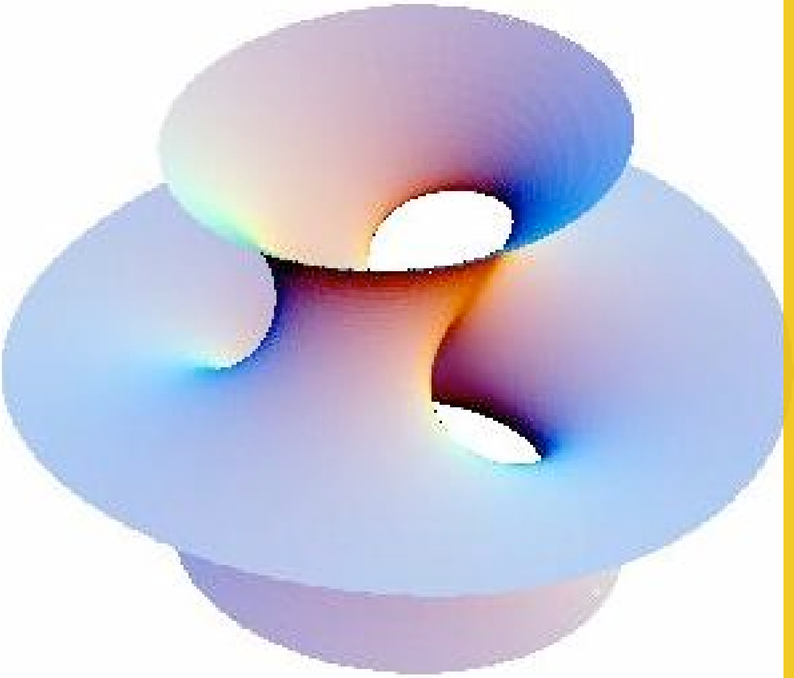
$N = 5$ $N = 6$ $N = 7$ **$N = 8$**

$N = 9$ $N = 10$ $N = 11$ **$N = 12$**

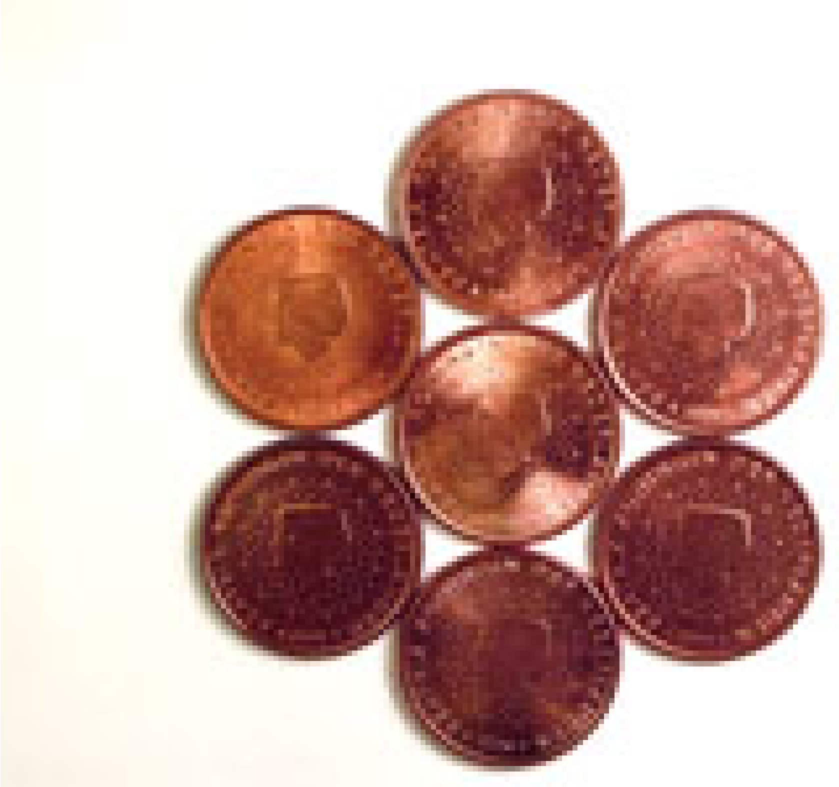
400-year anniversary of the Kepler conjecture (1611)

Fejes Toth at Fields

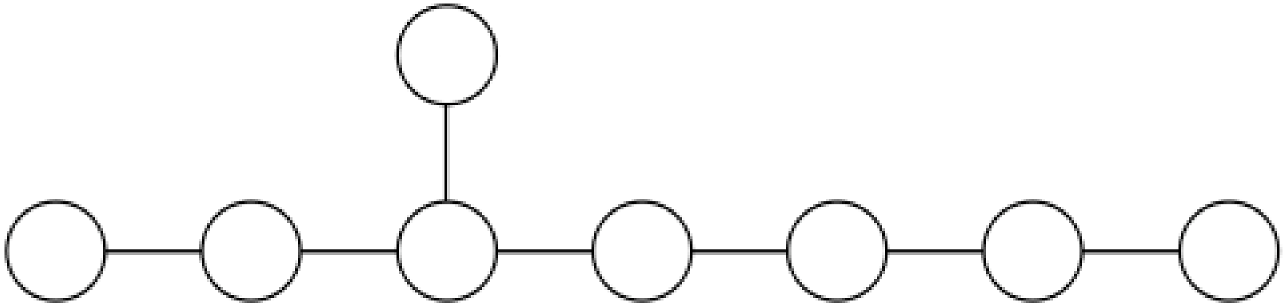
The Costa surface



Kissing numbers



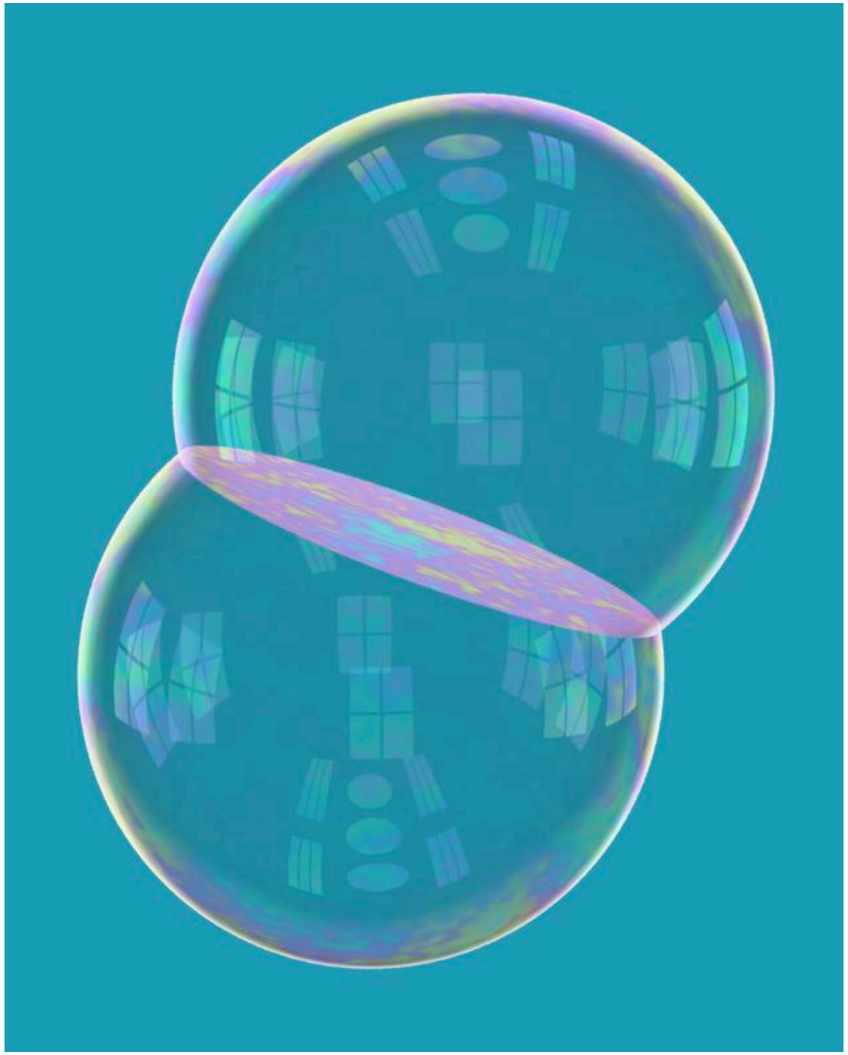
The E8 lattice and Lie group



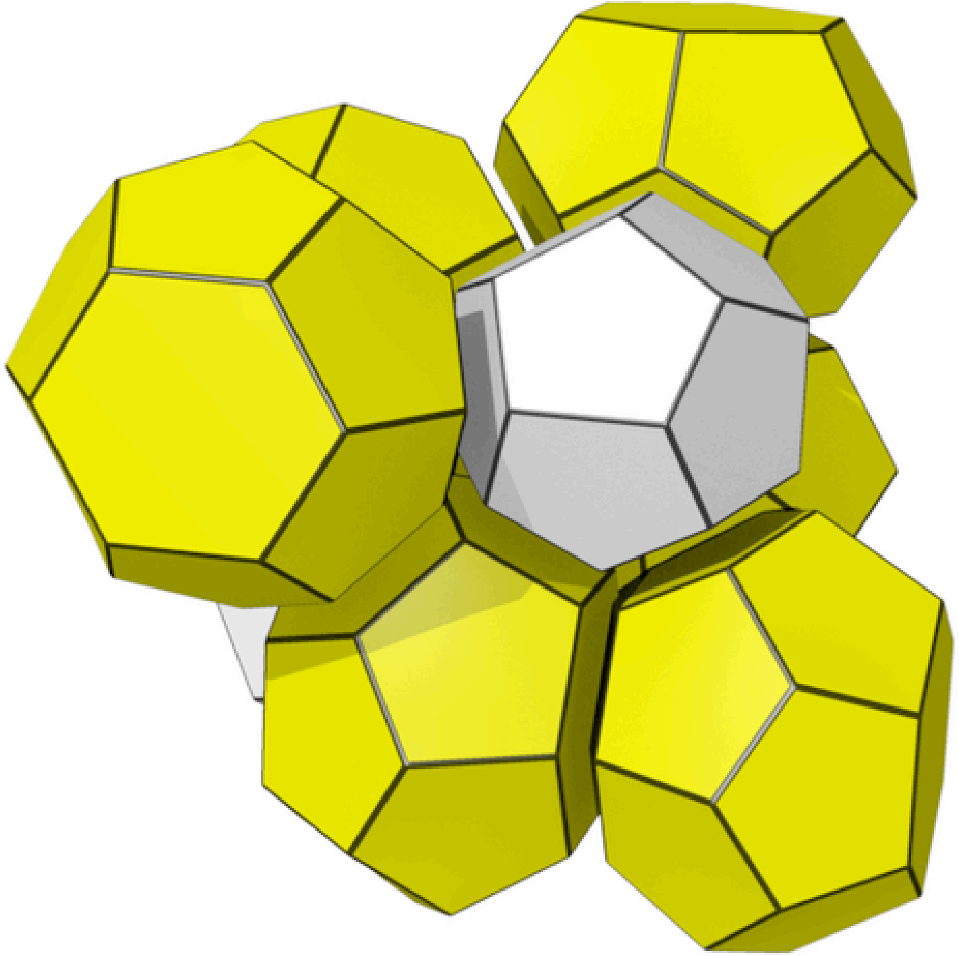
By 2007, a computer had completed the character table of E_8. Since there are infinitely many irreducible characters and each character is an analytic function on (a dense open subset of) the group, it is not clear without much further explanation what it might even mean for a computer to output the full character table as a 60 gigabyte file.

The Atlas project brings the computer to bear on some abstract parts of mathematics that have been traditionally largely beyond the reach of concrete computational description, including infinite dimensional representations of Lie groups, intersection cohomology and perverse sheaves. Vogan's account of this computational project was awarded the 2011 Conant Prize of the AMS.

Double -bubble problem



Phelan-Weaire



Rogers-Ramanujan identities via q-WZ

The famous Rogers-Ramanujan identities

$$1+\sum_{k=1}^{\infty}\frac{q^{k^2+ak}}{(1-q)(1-q^2)\cdots(1-q^k)}=\prod_{j=0}^{\infty}\frac{1}{(1-q^{5j+a+1})(1-q^{5j-a+4})},\qquad a=0,1.$$

Zeilberger says I shouldn't waste time "dotting i's for the sake of a Princeton professor."

"There are so many open problems left to do, Tom, so don't waste your time trying to find a "formal proof" version to Kepler. . . Let's be happy with the current standards of rigor in informal human mathematical discourse, and use computers with that level." (Zeilberger opinion 94)

For me, the reasons for turning to formal proof are much more complex. Simply put, we cannot build skyscrapers out of adobe bricks (that is, informal discourse and ordinary programming tools).

The use of computers in mathematics is a done deal. That day has already dawned. The Kepler conjecture reached the limits of what can be done without better computational tools. It is up to us now to build the desperately needed reinforced steel to support our structures.

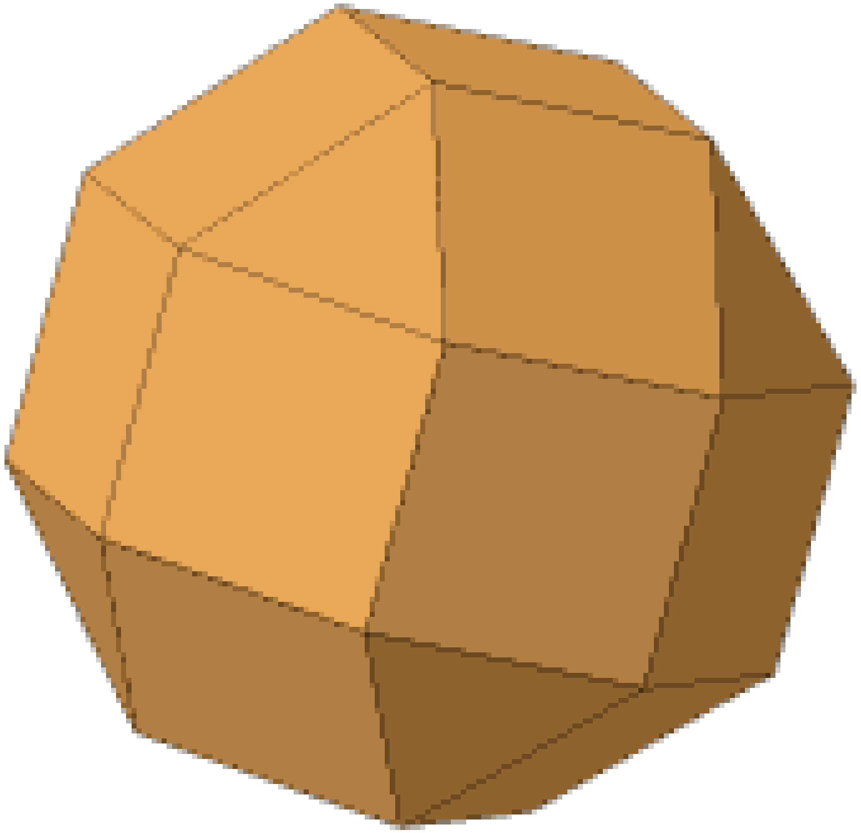
We have reckless trust in computers

But what about the Flash Crash on Wall Street that brought a 600 point plunge in the Dow Jones in just 5 minutes at 2:41 pm on May 6, 2010? According to the New York Times [NYT10], the flash crash started when a mutual fund used a computer algorithm "to sell $4.1 billion in futures contracts." The algorithm was designed to sell "without regard to price or time.. . .[A]s the computers of the high-frequency traders traded [futures] contracts back and forth, a 'hot potato' effect was created." When computerized traders backed away from the unstable markets, share prices of major companies fluctuated even more wildly. "Over 20,000 trades across more than 300 securities were executed at prices more than 60% away from their values just moments before" [SEC10] Throughout the crash, computers followed algorithms to a T, to the havoc of the global economy.

Year	Theorem	Proof System	Formalizer	Traditional Proof
1986	First Incompleteness	Boyer-Moore	Shankar	Gödel
1990	Quadratic Reciprocity	Boyer-Moore	Russinoff	Eisenstein
1996	Fundamental - of Calculus	HOL Light	Harrison	Henstock
2000	Fundamental - of Algebra	Mizar	Milewski	Brynski
2000	Fundamental - of Algebra	Coq	Geuvers et al.	Kneser
2004	Four Color	Coq	Gonthier	Robertson et al.
2004	Prime Number	Isabelle	Avigad et al.	Selberg-Erdös
2005	Jordan Curve	HOL Light	Hales	Thomassen
2005	Brouwer Fixed Point	HOL Light	Harrison	Kuhn
2006	Flyspeck I	Isabelle	Bauer-Nipkow	Hales
2007	Cauchy Residue	HOL Light	Harrison	classical
2008	Prime Number	HOL Light	Harrison	analytic proof

Incorrect proofs of correct statements are so abundant that they are impossible to catalogue. Kempe's claimed proof of the four-color theorem stood for more than a decade before Heawood refuted it [Mac01, p. 115]. "More than a thousand false proofs [of Fermat's Last Theorem] were published between 1908 and 1912 alone" [Cor10]. Ralph Boas, former executive editor of Math Reviews, once remarked that proofs are wrong "half the time" [Aus08]. Many published theorems are like the hanging chad

Pseudo rhombic cuboctahedron



Theorems that are calculations or enumerations are especially prone to error. Feynman laments, “I don’t notice in the morass of things that something, a little limit or sign, goes wrong.. . . I have mathematically proven to myself so many things that aren’t true.” Elsewhere, Feynman describes two teams of physicists who carried out a two-year calculation of the electron magnetic moment and independently arrived at the same predicted value. When experiment disagreed with prediction, the discrepancy was eventually traced to an arithmetic error made by the physicists, whose calculations were not so independent as originally believed. Pontryagin and Rokhlin erred in computing stable homotopy groups of spheres. Little’s tables of knots from 1885 contains duplicate entries that went undetected until 1974. In enumerative geometry, in 1848, Steiner counted 7776 plane conics tangent to 5 general plane conics, when there are actually only 3264.

In HOL Light we trust

To what extent can we trust theorems certified by a proof assistant such as HOL Light? There are various aspects to this question. Is the underlying logic of the system consistent? Are there any programming errors in the implementation of the system? Can a devious user find ways to create bogus theorems that circumvent logic? Are the underlying compilers, operating system, and hardware reliable?

- Is the underlying logic of the system consistent? YES
- Are there any programming errors in the implementation of the system? NO
- Can a devious user find ways to create bogus theorems that circumvent logic? YES
- Are the underlying compilers, operating system, and hardware reliable? SOMEWHAT

Hacking HOL

- Strings are *mutable*.
- *Object magic* defeats the type system.
- There are further Pollack inconsistencies: Substitute a variable with name `n<0 ∧ 0' for t in $\exists n.\ t < n$ to obtain a visual inconsistency $\exists n.\ n < 0\ \wedge\ 0 < n$.

Soft errors

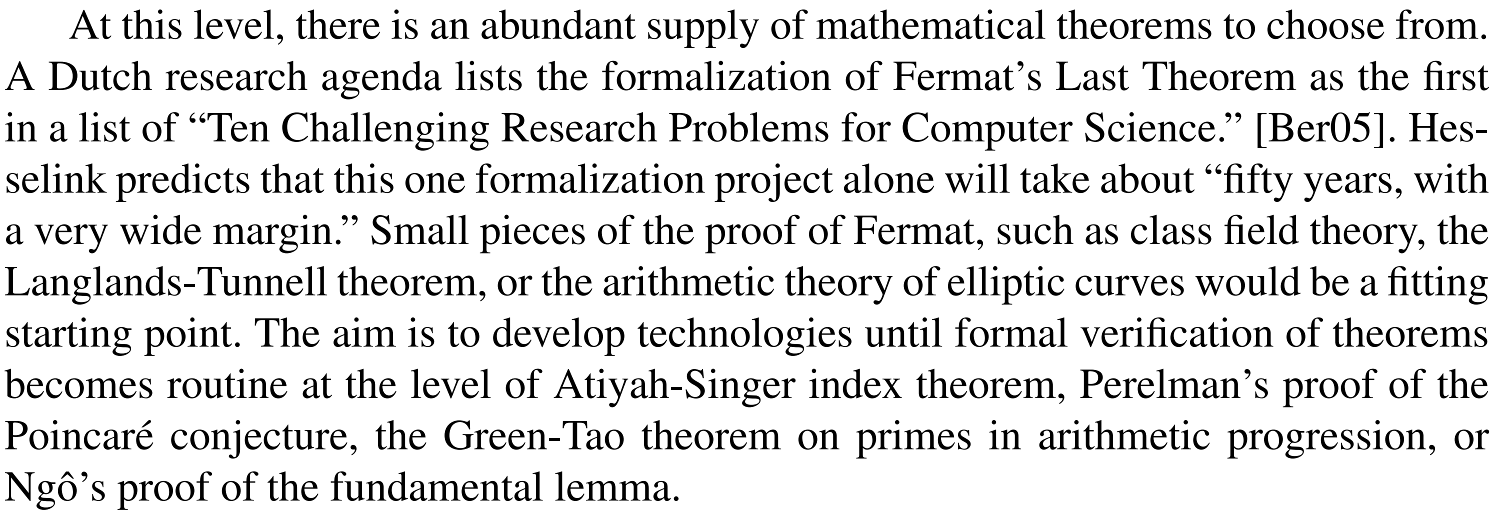
As an example, we will calculate the expected number of soft errors in one of the mathematical calculations of Section 1.17. The Atlas Project calculation of the E_8 character table was a 77 hour calculation that required 64 gigabytes RAM [Ats]. Soft errors rates are generally measured in units of failures-in-time (FIT). One FIT is defined as one error per 10^9 hours of operation. If we assume a soft error rate of 10^3 FIT per Mbit, (which is a typical rate for a modern memory device operating at sea level[15] [Tez04]), then we would expect there to be about 39 soft errors in memory during the calculation:

$$\frac{10^3 \text{ FIT}}{1 \text{ Mbit}} \cdot 64 \text{ GB} \cdot 77 \text{ hours} = \frac{10^3 \text{ errors}}{10^9 \text{ hours Mbit}} \cdot (64 \cdot 8 \cdot 10^3 \text{ Mbit}) \cdot 77 \text{ hours} \approx 39.4 \text{ errors.}$$

The Feit-Thompson theorem (Gonthier style)

```
Structure finGroupType Type := FinGroupType {
    element :> finType;
          1 : element;
     $^{-1}$ : element $\to$ element;
          * : element $\to$ element $\to$ element;
      unitP : $\forall\, x,\ \ 1 * x = x$;
       invP : $\forall\, x,\ \ x^{-1} * x = 1$;
       mulP : $\forall\, x_1\ \ x_2\ \ x_3,\ \ x_1 * (x_2 * x_3) = (x_1 * x_2) * x_3$
}.
```

Future Challenges...



At this level, there is an abundant supply of mathematical theorems to choose from. A Dutch research agenda lists the formalization of Fermat's Last Theorem as the first in a list of "Ten Challenging Research Problems for Computer Science." [Ber05]. Hesselink predicts that this one formalization project alone will take about "fifty years, with a very wide margin." Small pieces of the proof of Fermat, such as class field theory, the Langlands-Tunnell theorem, or the arithmetic theory of elliptic curves would be a fitting starting point. The aim is to develop technologies until formal verification of theorems becomes routine at the level of Atiyah-Singer index theorem, Perelman's proof of the Poincaré conjecture, the Green-Tao theorem on primes in arithmetic progression, or Ngô's proof of the fundamental lemma.

The Language of Mathematics

Ganesalingam's thesis is the most significant linguistic study of the language of mathematics to date. Ganesalingam was awarded the 2011 Beth Prize for the best dissertation in Logic, Language, or Information.

- infix (e.g. +),
- postfix (e.g. factorial !),
- prefix (cos).
- subscripted infix operators ($x +_n y$),
- multi-symboled operators $[\ :\ ]$,
- prefixed words (R-module),
- text within formulas $\{(a, b) \mid a \text{ is a factor of } b\}$,
- unusual script placement ${}^L G$,
- chained relations $a < b < c$,
- ellipses $1 + 2 + \cdots + n$,
- contracted forms $x, y \in \mathbb{N}$,
- exposed formulas ("for all $x > 0, \ldots$").

Automating proof assistants: Harrison’s Groebner basis in HOL Light

$$
\begin{aligned}
y_1^2 &= 1 + a x_1^2 + b x_1^4, \\
x_2 y_1 &= x_1, \\
y_2 y_1^2 &= (1 - b_1^4), \\
y_1 &\neq 0
\end{aligned}
$$

then (x_2, y_2) lies on a second elliptic curve

$$
y_2^2 = 1 + a' x_2^2 + b' x_2^4,
$$

Computer algebra within proof assistants (Kaliszyk and Wiedijk)

```
 In1 := (3 + 4 DIV 2) EXP 3 * 5 MOD 3
Out1 := 250
 In2 := vector [&2; &2] - vector [&1; &0] + vec 1
Out2 := vector [&2; &3]
 In3 := diff (diff (\x. &3 * sin (&2 * x) + &7 + exp (exp x)))
Out3 := \x. exp x pow 2 * exp (exp x) + exp x * exp (exp x) + -- &12 * sin (&2 * x)
 In4 := N (exp (&1)) 10
Out4 := #2.7182818284 + ... (exp (&1)) 10 F
 In5 := 3 divides 6 /\ EVEN 12
Out5 := T
 In6 := Re ((Cx (&3) + Cx (&2) * ii) / (Cx (-- &2) + Cx (&7) * ii))
Out6 := &8 / &53
```

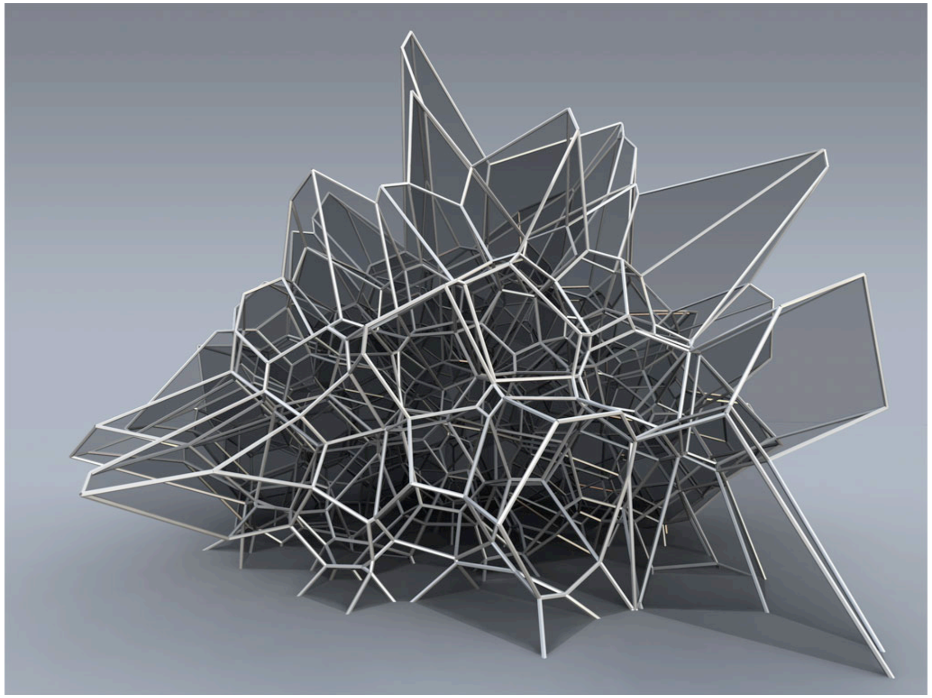
The Stanley sequence

$$\left\lceil \frac{2}{2^{1/n}-1} \right\rceil - \left\lfloor \frac{2n}{\log 2} \right\rfloor, \quad n = 1, 2, 3, \ldots$$

starts out as the zero sequence, but remarkably first gives a nonzero value when n reaches $777, 451, 915, 729, 368$ and then again when $n = 140, 894, 092, 055, 857, 794$.

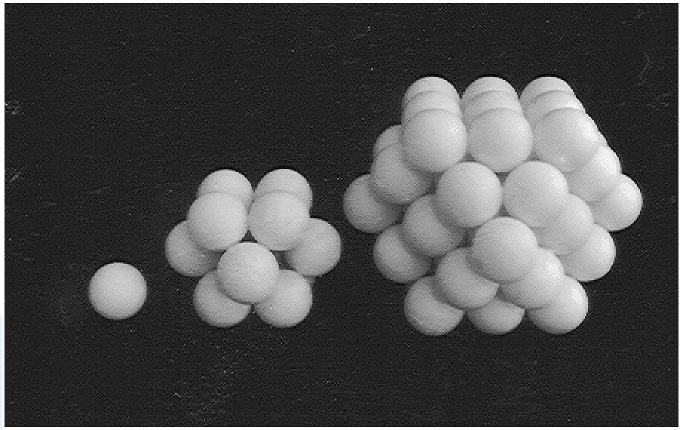
Wednesday: dodecahedral conjecture
Friday: Fejes Toth's contact conjecture

Fejes Toth at Fields



- The Kepler conjecture asserts that the densest packing of congruent balls in $\mathbb{R}^3$ is achieved by the familiar "cannonball" arrangement.
- The Kepler Conjecture was formulated in the booklet "The six-cornered snowflake," presented as a gift on New Year's day 1611 to Kepler's patron Lord Wacker von Wackenfels.

Kepler asks why a snowflake has six sides. This leads to honeycombs, pomegranates, and then sphere packings.



- The first proof was presented (by Ferguson and H. in 1998) and published in 2006.
- A project called Flyspeck seeks to give a formal proof of the theorem, which involves a computer verification of every single logical inference in the proof, all the way back to the fundamental axioms of mathematics.
- The Flyspeck project is about 80% complete.

cluster fly (Herrick)

General comments on formalization:

- Computers have become the medium of choice for the foundations of mathematics.
- Research on formalization might profit from greater participation from mathematicians.
- Two valuable activities are Bourbakization and the Rising Sea.
- (Almost all of my formalization work has gone into the Bourbakization of the proof of the Kepler conjecture.)

The Flyspeck project (expected to take about 20 work years) is about 75% – 80% complete. The project has four parts:

1. The text part of the proof is contained in an unpublished manuscript “Dense Sphere Packings: a formal blueprint.” Formalization is being done by a team of researchers (Harrison, Nguyen Quang Truong, Solovyev, Hoang Le Truong, Tran Nam Trung, and several others).
2. The first computer program (plane graph generation) was formalized by G. Bauer and T. Nipkow.
3. The second computer program (linear programming) is nearly formalized by S. Obua and A. Solovyev.
4. The third computer program (nonlinear inequality proving) is work in progress.

The Bourbakization of a web of conjectures related to the Kepler conjecture…First conjecture: a variation on Fejes Tóth's kissing problem estimate (1953). Let 14 nonoverlapping balls of diameter 1 be given with centers P_i, $i = 0, \ldots, 13$. Let

$$a = 7/\sqrt{27} \approx 1.347$$

Is

$$\sum_{i=1}^{13} P_0P_i \geq 12 + a \approx 13.347?$$

Here is a variant. Let

$$L(h) = \begin{cases} \frac{h_0 - h}{h_0 - 1} & h \leq h_0 \\ 0 & h \geq h_0. \end{cases}$$

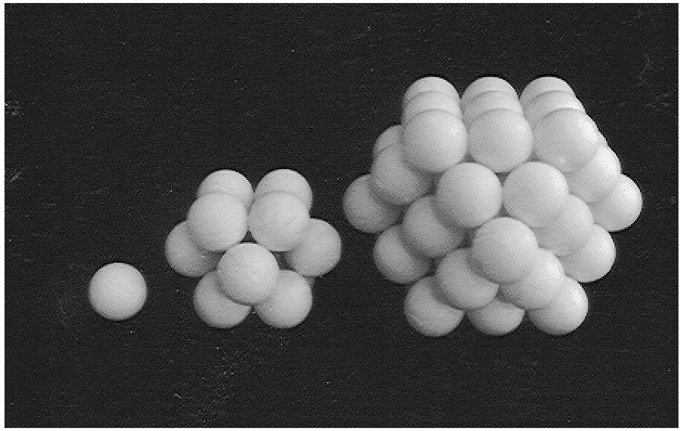
where $h_0 = 1.26$.

Conjecture 1 (L12)**.** *Let $P_0, \ldots, P_N$ be the centers of N nonoverlapping balls. Set $h_i = P_0 P_i$. Then*

$$\sum_{i=1}^{N} L(h_i) \leq 12.$$

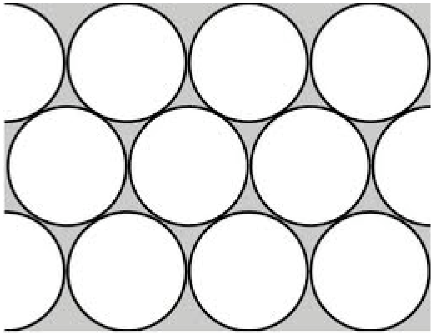
(If $N = 13$ and h_0 is increased to a, then it becomes Fejes Tóth's kissing number conjecture from 1953.)

Conjecture 2 (Kepler (1611))**.** *The densest packing of congruent balls in* $\mathbb{R}^3$ *is attained (non-uniquely) by the face-centerd cubic packing.*



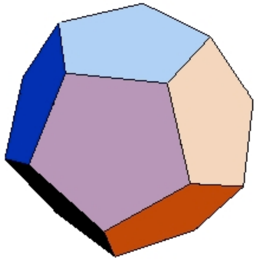
Conjecture 3 (Fejes Tóth's full contact conjecture (1969))**.** *In 3-space a packing of equal balls such that each enclosed ball is touched by 12 others consists of hexagonal layers.*

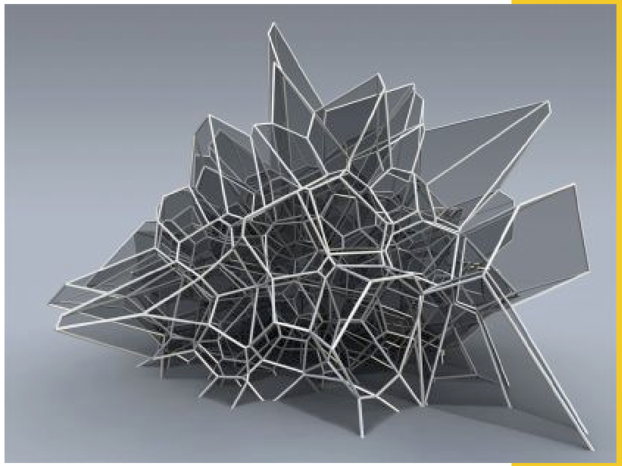
(The corresponding problem in the plane is trivial. If each unit disk in the plane touches 6 others then it must be the regular hexagonal packing of disks.)



Conjecture 4 (K. Bezdek's strong dodecahedral conjecture (2000))**.** *In every packing of congruent balls in $\mathbb{R}^3$, the surface area of every Voronoi cell is at least that of the (circumscribing) regular dodecahedron.*

(The strong dodecahedral conjecture implies the weak dodecahedral conjecture, which was proved by S. McLaughlin in 1998, and published last year.)





Theorem 1. *The L12 conjecture (the variant of FT's kissing number estimate from 1953) implies all of the other conjectures:*

1. *L12 implies the Kepler conjecture.*
2. *L12 implies FT's full contact conjecture.*
3. *L12 implies the strong dodecahedral conjecture.*

- The proof of this theorem relies on computer.
- About 500 automatically proved nonlinear inequalities are involved.
- The inequalities are specified in a formal proof system (HOL Light).
- From the formal specification, computer code is automatically generated that checks them numerically (by a gradient descent algorithm) and then checks them rigorously (by interval arithmetic).
- (The interval arithmetic has still not been formalized. This is biggest part of the remaining 20% – 25% of the flyspeck project.)
- The amount of computer code has been reduced from 187K lines of code to well under 10K.

What is the status of inequality L12?

- I have an incomplete proof of L12.
- The only missing piece of L12 are 1 additional nonlinear inequalities that are currently being verified by computer. This remaining inequality is similar to but slightly more difficult than the other 500.
- What this means is that I am reasonably confident that I can make an announcement of every one of the conjectures sometime within the coming month.

Observations:

- The proof that L12 implies Kepler is adapted from a recent paper by C. Marchal that M12 implies Kepler.
- The proof of L12 (modulo the 1 inequality) is adapted from the 1998 proof of the Kepler conjecture.
- The Flyspeck project is formalizing L12 and (L12 implies Kepler).
- We would have none of these new theorems without the impetus from formal mathematics to push us towards a radical simplification of the original long computer proof.

The Flyspeck project (expected to take about 20 work years) is about 75% – 80% complete. The project has four parts:

1. The text part of the proof is contained in an unpublished manuscript “Dense Sphere Packings: a formal blueprint.” Formalization is being done by a team of researchers (Harrison, Nguyen Quang Truong, Solovyev, Hoang Le Truong, Tran Nam Trung, and several others).

2. The first computer program (plane graph generation) was formalized by G. Bauer and T. Nipkow.

3. The second computer program (linear programming) is nearly formalized by S. Obua and A. Solovyev.

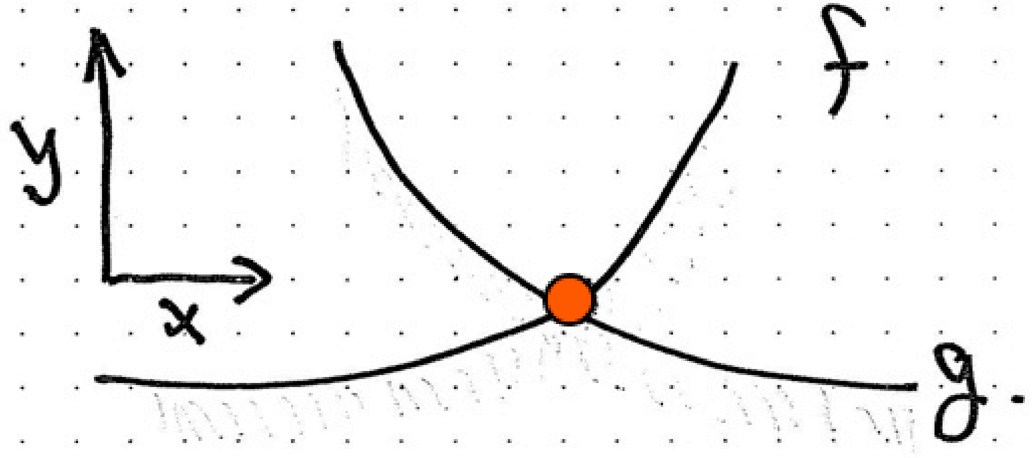
4. The third computer program (nonlinear inequality proving) is work in progress.

Graph Generation:

- The formalization of the computer program that classifies planar graphs was the first success of the Flyspeck project (G. Bauer and T. Nipkow)
- Nipkow visited Pittsburgh in August to update the formal proof so that it reflects the revised proof of the Kepler conjecture.
- The computer programs makes the classification up to plane graph isomorphism of all planars graphs with specific properties. There are about 25K such graphs.
- In doing so, he uncovered a bug in my original code (that went unexercised in the original proof). The bug was an uninitialized structure that gets used in symmetry reductions.

Linear programming

Kepler 2011

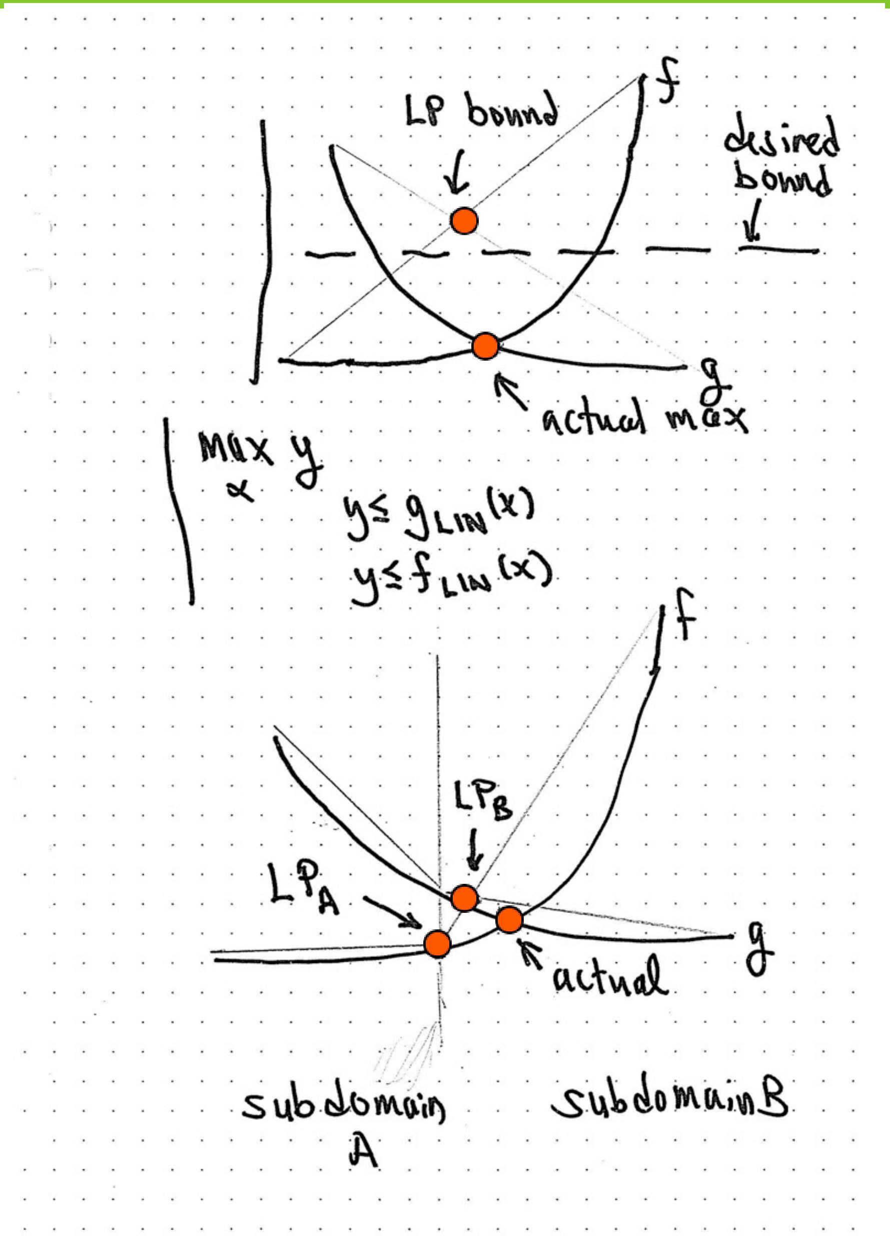


$$\max_{x} \; y$$

s.t. $y \le g(x)$ and

$y \le f(x)$

$\max_{x} \; y.$

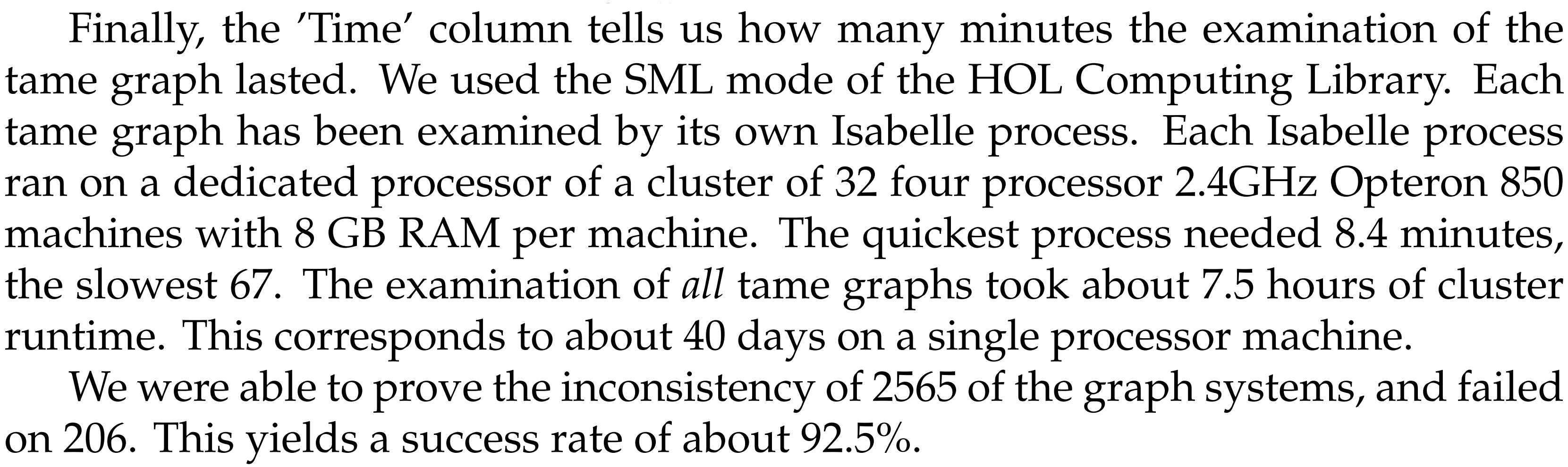


$$\max_{x} y$$
$$y \leq g_{LIN}(x)$$
$$y \leq f_{LIN}(x)$$

Obua’s Thesis

- Does the “basic linear programs”
- Uses an external floating-point module
- Most of the compute time goes into graph combinatorics
- Obua worked out the linear programming issues (checking certificates inside the proof assistant, compensating for floating point errors).

Benchmarks from Obua's Thesis.

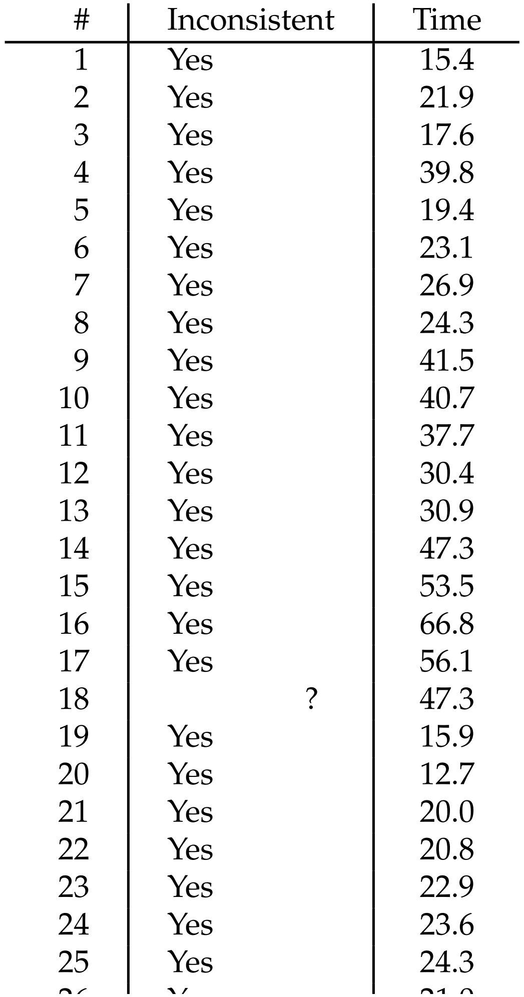


Finally, the 'Time' column tells us how many minutes the examination of the tame graph lasted. We used the SML mode of the HOL Computing Library. Each tame graph has been examined by its own Isabelle process. Each Isabelle process ran on a dedicated processor of a cluster of 32 four processor 2.4GHz Opteron 850 machines with 8 GB RAM per machine. The quickest process needed 8.4 minutes, the slowest 67. The examination of *all* tame graphs took about 7.5 hours of cluster runtime. This corresponds to about 40 days on a single processor machine.

We were able to prove the inconsistency of 2565 of the graph systems, and failed on 206. This yields a success rate of about 92.5%.

source: Obua's thesis

Benchmarks



#	Inconsistent	Time
1	Yes	15.4
2	Yes	21.9
3	Yes	17.6
4	Yes	39.8
5	Yes	19.4
6	Yes	23.1
7	Yes	26.9
8	Yes	24.3
9	Yes	41.5
10	Yes	40.7
11	Yes	37.7
12	Yes	30.4
13	Yes	30.9
14	Yes	47.3
15	Yes	53.5
16	Yes	66.8
17	Yes	56.1
18	?	47.3
19	Yes	15.9
20	Yes	12.7
21	Yes	20.0
22	Yes	20.8
23	Yes	22.9
24	Yes	23.6
25	Yes	24.3

source: Obua's thesis

2010 Reworking of the LPs

Floating Point Issues

- The combinatorics have been eliminated.
- A typical problem has 200 variables, 2000 constraints, 100, 000 linear programs.
- By LP theory, only 200 constraints are active on a problem with 200 variables.
- The matrix is sparse.
- Most cases need a single digit of precision.

2010 Reworking of the LPs

The current approach

- The linear programming is done in GLPK.
- There is an AMPL model that is indpendent of the hypermap. (It is the same model for all $25,000$ hypermaps.)
- There is a OCAML generated AMPL data file for each linear program.

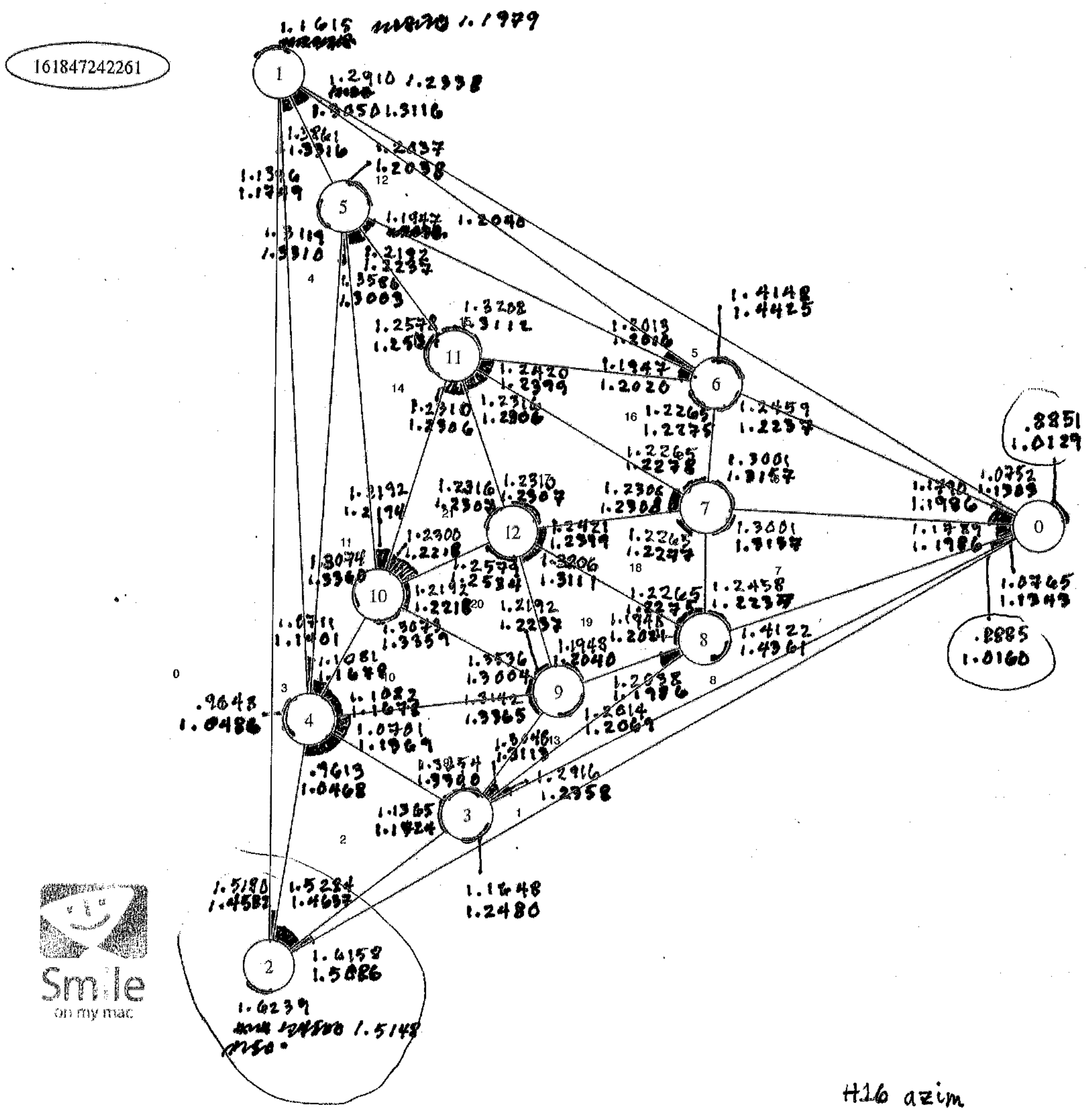
2010 Reworking of the LPs

Subdivision

- Subdivision of a problem that is already linear causes a needless blowup in the number of cases.
- An intelligent scheme for subdivision of the problem should be based on the location of the nonlinearities.

2010 Reworking of the LPs

Kepler 2011



H16 azim
new calc.

2010 Reworking of the LPs

Our approach is to compute all of the dihedral angles (based on optimal edge lengths returned by the linear program) and compare them to the linearized dihedral angles.

The angles are ranked by the size of the error.

Each angle is attached a weight, according to the number of subdivisions that have already occurred at that angle.

The angle with the largest weight error is used for subdivision.

2010 Reworking of the LPs

The number of subdivisions is limited by the specifications of the model. It is independent of the hypermap.

If all of the weights are zero, then no further subdivision is possible. A new inequality must be designed and added to the system.

2010 Reworking of the LPs

Adding new inequalities

- Several programs are used (all automated).
- A mathematica procedure based on heuristics is used to generate a candidate inequality.
- The inequality is shipped to cfsqp for testing by nonlinear optimization methods.
- A formal specification is automatically generated in HOL Light.
- The AMPL model is automatically updated with the new inequality. (The inequality is added to all linear programs.)

2010 Reworking of the LPs

- This work was all *informal*, but done with formalization in mind.
- At this point A. Solovyev took over the project and began to formalize the linear programming.
- He implemented linear program checking inside HOL Light.
- He optimized real arithmetic calculations inside HOL Light.
- He can now make a formal verification of a large-scale linear program in about 3 seconds. (Read/write operations rather than real arithmetic dominate the times.)
- Compare Obua's benchmarks of about 20 minutes per LP, even when performing real arithmetic outside the proof assistant.

Nonlinear inequalities

The Flyspeck project (expected to take about 20 work years) is about 75% – 80% complete. The project has four parts:

1. The text part of the proof is contained in an unpublished manuscript "Dense Sphere Packings: a formal blueprint." Formalization is being done by a team of researchers (Harrison, Nguyen Quang Truong, Solovyev, Hoang Le Truong, Tran Nam Trung, and several others).
2. The first computer program (plane graph generation) was formalized by G. Bauer and T. Nipkow.
3. The second computer program (linear programming) is nearly formalized by S. Obua and A. Solovyev.
4. The third computer program (nonlinear inequality proving) is work in progress.

Nonlinear inequalities

- Since September, I have been working on the informal proof of a collection of about 500 nonlinear inequalities.
- Testing of inequalities is done with a gradient descent program, developed at U. Maryland.
- Interval arithmetic verification is done by code developed for the 1998 proof of the Kepler conjecture. (A few thousand lines)
- The C++ code to test and verify each inequality is automatically generated from the formal specification. It automatically converts inequalities into an optimized form, splits piecewise analytic functions into analytic pieces, . . .
- There are other programs for informal proofs of nonlinear inequalities by Ferguson, McLaughlin, and Zumkeller.

Nonlinear inequalities

What were the challenges over recent months?

- The collection of nonlinear inequalities is heterogeneous. It took some work to make automated code generation to work uniformly on this collection.
- All but the last step of the automated code generation is done inside HOL Light. In particular, all of the major transformations are formally justified.
- Many of the nonlinear inequalities have naturally occurring instabilities: $1/0$, $0/0$, $\sqrt{0}$, piecewise continuity. They have all been transformed into C^∞ functions.
- The code can now deal with some sharp inequalities with interior extreme points. (The problem of equality.)

Nonlinear inequalities

Towards a formal verification of the nonlinear inequalities

- Soon this will be the only remaining piece of the Flyspeck project. It was always expected to be the most difficult part.
- The next step will to be to run a test case inside HOL, using the real interval arithmetic for HOL that A. Solovyev is developing.

Thank You!