Practical solutions to hard problems in 3-dimensional topology.

Nathan M. Dunfield
University of Illinois

Fields Institute, November 20, 2009

This talk available at http://dunfield.info/
Math blog: http://ldtopology.wordpress.com/
In contrast to higher dimensions, many properties of $M^3$ are algorithmically computable.

[Haken 1961] Whether a knot in $S^3$ is unknotted. More generally, find the simplest surface representing a class in $H_2(M; \mathbb{Z})$.


[Rubinstein-Thompson 1995] Whether $M$ is $S^3$. Casson showed this allows finding connected sum decompositions.

[Haken-Hemion-Matveev] Whether two Haken 3-manifolds are homeomorphic.
Thurston and Perelman: 3-manifolds have canonical decompositions into geometric pieces modeled on $\mathbb{E}^3$, $S^3$, $H^3$, $S^2 \times \mathbb{R}$, $H^2 \times \mathbb{R}$, Nil, Sol, $\widetilde{SL_2 \mathbb{R}}$.

The work of Perelman, Casson-Manning, Epstein et. al., Hodgson-Weeks, Jaco-Oertel, Haken-Hemion-Matveev, Casson, Rubinstein-Thompson, and others gives

**Thm.** There is an algorithm to determine if two compact 3-manifolds are homeomorphic.

Other directions: Heegaard Floer homology, quantum invariants...
How hard are these questions?

[Agol-Hass-Thurston 2002] The following is NP-complete:

Q: Given a manifold $M$, a knot $K$ in $\mathcal{T}^1$, and $g \in \mathbb{N}$, is there a surface $\Sigma \subset M$ with boundary $K$ and genus $\leq g$?

[Casson, Schleimer, Ivanov 2004] Recognizing the 3-sphere is in NP.
Normal surfaces meet each tetrahedra in a standard way:

and correspond to certain lattice points in a finite polyhedral cone in $\mathbb{R}^{7t}$ where $t = \#\mathcal{T}$:
Meta Thm. In an interesting class of surfaces, there is one which is normal. Moreover, one lies on a vertex ray of the cone.

E.g. The class of minimal genus surfaces whose boundary is a given knot.

Problem: the dimension grows linearly with $t$, and moreover there can be exponentially many vertex rays. In practice, limited to $t < 40$.

Worse, sometimes have a second step examining each $M \setminus \Sigma$ and looking for surfaces there, and that new manifold may be much more complicated than $M$ itself.
Thm. (Dunfield-Ramakrishnan 2007) There is a closed hyperbolic 3-manifold $M$ of arithmetic type, with an infinite family of finite covers $\{M_n\}$ of degree $d_n$, where the number $\nu_n$ of fibered faces of the Thurston norm ball of $M_n$ satisfies

$$\nu_n \geq \exp \left( 0.3 \frac{\log d_n}{\log \log d_n} \right) \quad \text{as } d_n \to \infty.$$ 

To prove this, we needed to compute the Thurston norm for a manifold with $\#T \approx 130$, and moreover show that it fibers over the circle!
Practical Trick 1: Finding the simplest surface representing some $\phi \in H^1(M; \mathbb{Z}) \cong H_2(M; \mathbb{Z})$.

Use a triangulation with only one vertex (cf. Casson, Jaco-Rubinstein). The $\phi$ comes from a unique 1-cocycle, which realizes $\phi$ as a piecewise affine map $M \to S^1$. 
Power of randomization: Trying several different triangulations usually yields the minimal genus surface.

Lower bounds on the genus come from (twisted) Alexander polynomials.

**Practical Trick 2:** Proving that $N = M \setminus \Sigma$ is $\Sigma \times I$.

Start with a presentation for $\pi_1(N)$ coming from a triangulation, and then simplify that it using Tietze transformations. With luck (i.e. randomization), one gets a one-relator presentation of a surface group. This gives $N \cong \Sigma \times I$ by [Stallings 1960].

To see that $N \not\cong \Sigma \times I$, try Alexander polynomials.

**Current work:** Can this work for other problems, e.g. finding incompressible surfaces?
Rank vs. genus (with Helen Wong)

A closed $M^3$ can always be constructed as

Consider

$$\text{rank}(M) = \text{min genus of a Heegaard splitting}$$
$$\text{genus}(M) = \text{min size of a gen set of } \pi_1 M$$

Clearly have $\text{rank}(M) \leq \text{genus}(M)$.

Q. Does $\text{rank}(M) = \text{genus}(M)$ for all hyperbolic 3-manifolds?

[Boileau-Zieschang 1984] There are Seifert fibered spaces with $\text{rank}(M) \neq \text{genus}(M)$.

Further graph manifold examples found by Weidmann and Schultens.
Searching for an example.

Computability in theory for $M^3$ hyperbolic:

- **Rank:** Yes [Kapovich-Weidmann 2004]
- **Genus:** Unknown, likely yes. Rubinstein and Stocking showed that (many) Heegaard surfaces can be made almost normal, but there are infinitely many candidates surfaces. [Lackenby 2008] When $M$ has cusps, can compute the genus by using the right triangulation.
Computability in practice.

- **Rank:** Occasionally. Can search for smaller generating sets via Todd-Coxeter coset enumeration. Lower bounds are hard to come by, except for the rank of $H_1(M; \mathbb{Z})$.

- **Genus:** Sometimes. Start with a presentation of $\pi_1(M)$ coming from a triangulation, then simplify via Tietze transformations. The result inevitably comes from a Heegaard splitting of $M$. Using randomization, can get a good idea of what the genus should be. Lower bounds, other than the rank, are few, e.g. quantum invariants.

Note: Quantum invariants can be used to reprove the examples of Boileau-Zieschang [Wong 2007].
So far, we don’t even have any candidate hyperbolic examples, even though our methods quickly find many of the known non-hyperbolic examples.

We think we’ve found a new non-hyperbolic example:

We know that $\text{rank}(M) = 3$ and strongly suspect that $\text{rank}(M) = 4$.
SnapPy

What is SnapPy?

SnapPy is a user interface to the SnapPea kernel which runs on Mac OS X, Linux, and Windows. SnapPy combines a link editor and 3D-graphics for Dirichlet domains and cusp neighborhoods with a powerful command-line interface based on the Python programming language. You can see it in action, learn how to install it, and read the tutorial.

Contents

- Screenshots: SnapPy in action
- Installing and running SnapPy
- Tutorial
- snappy: A Python interface for SnapPea
- Using SnapPy’s link editor
- To Do List
- Development Basics: OS X
- Development Basics: Windows XP

Credits

Written by Marc Culler and Nathan Dunfield. Uses the SnapPea kernel written by Jeff Weeks. Released under the terms of the GNU General Public License.