The Weber-Seifert dodecahedral space

Answering a computational challenge

Benjamin Burton

The University of Queensland

February 2010

Introduction

Several core problems in knot theory and low-dimensional topology are algorithmic. Examples:

- **Unknotting problem:** How do we tell whether a given knot is equivalent (isotopic) to the unknot?
- **Homeomorphism problem:** How do we tell whether two given 3-manifolds are equivalent (homeomorphic)?
- **Identification problem:** Given some representation of a 3-manifold (such as a triangulation), find its name/structure.

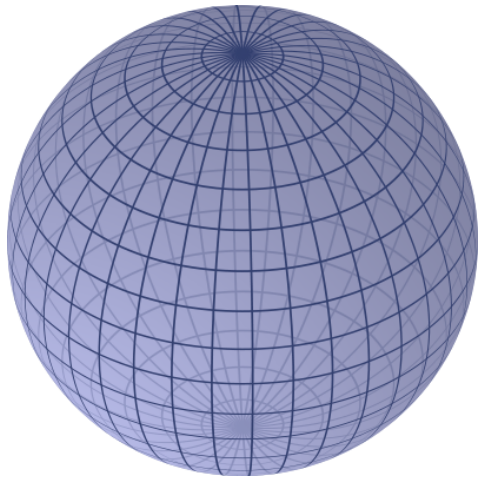
There are algorithms, and there are *practical* algorithms.

- Algorithms should be **fast** enough to run
- Algorithms should be **simple** enough to code

Introduction (ctd.)

Consider 3-sphere recognition:

- **First algorithm:** Almost normal surfaces (Rubinstein, 1992)
- Simplify “almost normal” (Thompson, 1994)
- Replace cutting with crushing (Jaco & Rubinstein, 2003)
- **First implemented:** Regina (B.B., 2004)
- Fast enumeration of normal surfaces (Letscher, 1997; B.B., 2008)
- Quadrilateral-octagon coordinates (Tollefson, 1998; B.B., 2009)
- **First application:** Computation with 4-manifold triangulations (Budney & B.B., work in progress)

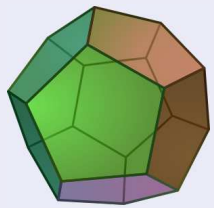


Our aim today

To resolve a conjecture made by Thurston around 1980:

Theorem (B.B., Rubinstein, Tillmann, 2009)

The Weber-Seifert dodecahedral space (one of the first known hyperbolic 3-manifolds) does not contain a two-sided incompressible surface.



Since 1984 there has been an algorithm to determine whether a given 3-manifold contains a two-sided incompressible surface.

$\longrightarrow$ This is a **computer proof**.

The main difficulty lies in making the algorithm **simple** and **fast** enough to code and run.

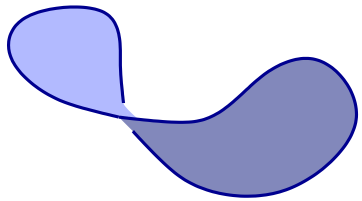
Outline

1. Theory
   - Normal surface theory
   - Haken manifolds and the Jaco-Oertel algorithm
2. Algorithms
   - The enumeration of vertex normal surfaces
   - Optimising Jaco-Oertel
3. Computation
   - The Weber-Seifert dodecahedral space

Normal surface theory

Normal surface theory gives us an algorithmic mechanism for finding interesting **surfaces** within a 3-manifold.

It appears in many important high-level algorithms:



- Unknot recognition (Haken, 1961)
- 3-sphere recognition (Rubinstein, 1992)
- Connected sum and JSJ decomposition (Jaco & Tollefson, 1995)
- Homeomorphism problem for Haken manifolds (Haken, Jaco & Oertel, Hemion, 1962–1984)
- Plus many more. . .

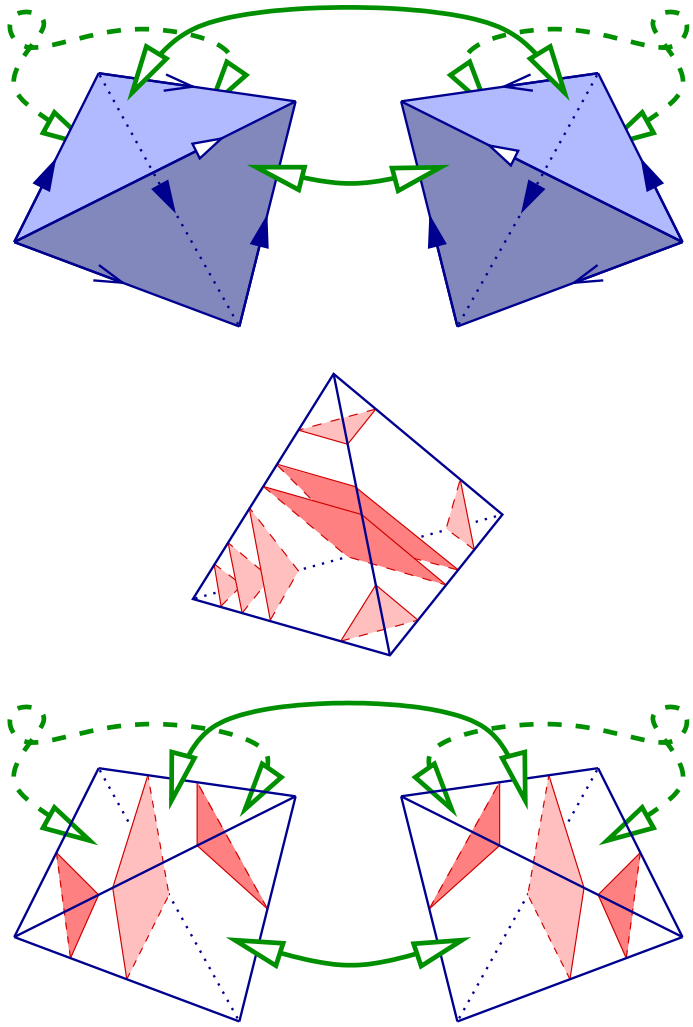
Essentially, normal surface theory allows us to search for surfaces in a **finite** and **discrete** manner.

Normal surface theory (ctd.)

We work with a 3-manifold triangulation, formed by identifying the faces of n tetrahedra in pairs.

A normal surface slices through each tetrahedron in normal discs (triangles and quadrilaterals).

Here we concentrate on embedded normal surfaces (no self-intersections).

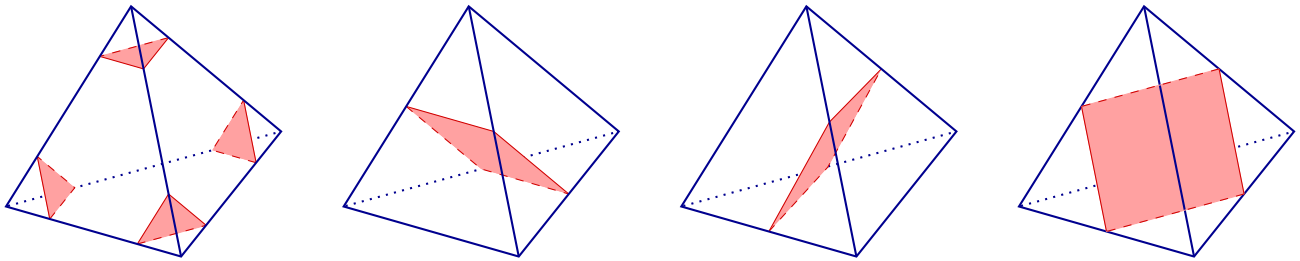


Normal surface theory: Vector representation

Normal surfaces can be described by a finite sequence of integers

$\longrightarrow$ Good for algorithms and computers!

Seven disc types for each tetrahedron:



The vector representation of a normal surface simply counts the number of discs of each type:

$$(t_{1,1}, t_{1,2}, t_{1,3}, t_{1,4}, q_{1,1}, q_{1,2}, q_{1,3}, t_{2,1}, t_{2,2}, \ldots, q_{n,3}) \in \mathbb{Z}^{7n},$$

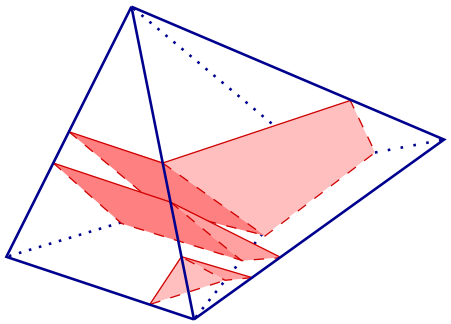
where n is the total number of tetrahedra.

$\longrightarrow$ Uniquely identifies the normal surface (up to normal isotopy).

Normal surface theory: Admissible vectors

What are the necessary and sufficient conditions for $\mathbf{v} \in \mathbb{Z}^{7n}$ to represent an embedded normal surface?

- Non-negativity: All coordinates are non-negative.
- Matching equations: For each pair of adjacent tetrahedra, the discs in each tetrahedron can be matched up:



$$t_{i,a} + q_{i,b} = t_{j,c} + q_{j,d}.$$

- Quadrilateral constraints: ≤ 1 quadrilateral type per tetrahedron. That is, for each i, at most one of $q_{i,1}, q_{i,2}, q_{i,3}$ is non-zero.

These conditions define an admissible vector.

Normal surface theory: The projective solution space

Theorem (Haken)

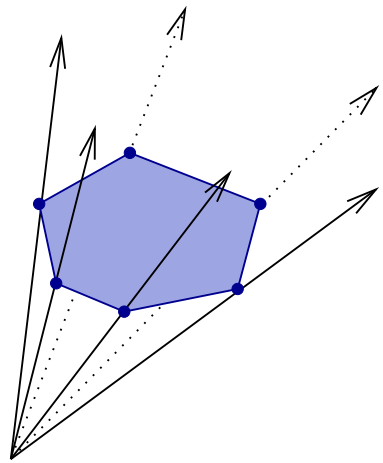
A non-zero vector in $\mathbb{Z}^{7n}$ represents an embedded normal surface if and only if it is admissible.

$\rightarrow$ To find normal surfaces, search for admissible vectors!

To make our search finite, we introduce the projective solution space. This is a **bounded** convex polytope, formed by intersecting:

- The non-negative orthant $O \in \mathbb{R}^{7n}$;
- The hyperplanes $H_1, \ldots, H_g$ defined by the matching equations;
- The projective hyperplane J defined by

$$J = \{\mathbf{x} \in \mathbb{R}^{7n} \;\Big|\; \sum x_i = 1\}.$$



Normal surface theory: The magic

Typical theorem

If a 3-manifold contains an "interesting" surface, then it contains an interesting **normal** surface, and moreover one that scales down to a **vertex** of the projective solution space.

True for two-sided incompressible surfaces (Jaco & Oertel), essential discs and spheres (Jaco & Tollefson), and more.

A typical topological algorithm contains the following key steps:

- Enumerate the vertices of the projective solution space;
- If a vertex satisfies the quadrilateral constraints, reconstruct the surface and test whether it is interesting.

If a connected orientable surface scales down to a vertex of the projective solution space, we call it a vertex normal surface.

Haken manifolds

A compact orientable irreducible 3-manifold is Haken if it contains a two-sided incompressible surface.

Haken manifolds are interesting because they admit hierarchies, which decompose the manifold by cutting along incompressible surfaces.

Such hierarchies offer natural frameworks for inductions and algorithms, e.g.:

- Haken manifolds are characterised by their fundamental group (Waldhausen, 1968).
- Geometrisation was initially proven for Haken manifolds (W. Thurston, 1979).
- The homeomorphism problem can be solved for Haken manifolds by constructing and comparing hierarchies (Haken, Hemion, Jaco & Oertel, 1962–1984).

The Jaco-Oertel algorithm

To determine whether a compact orientable irreducible 3-manifold M is Haken:

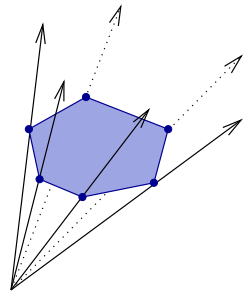
- Find a triangulation T of M.
- Construct all vertex normal surfaces in T.
  Each is a *potential* incompressible surface.
- For each vertex normal surface S:
  - Cut T along the surface S and retriangulate as T'.
  - Search for a compressing disc in the triangulation T', i.e., a properly embedded disc whose boundary is non-trivial in $\partial T'$.

    If there are compressing discs, one must appear as a fundamental normal surface in T' (a generalisation of vertex normal surfaces).
  - If no compressing disc is found, S is incompressible and M is Haken.
- If no incompressible surface is found, M is non-Haken.

The enumeration of vertex normal surfaces

The difficult part is now enumerating the vertices of the projective solution space, where we intersect:

- The non-negative orthant O;
- The projective hyperplane J where $\sum x_i = 1$;
- The matching hyperplanes $H_1, \ldots, H_g$.



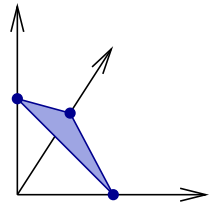
Use the double description method (Motzkin et al., 1953), a technique from linear programming and operations research.

This is simple to code, but can become **exponentially slow** in the number of tetrahedra.

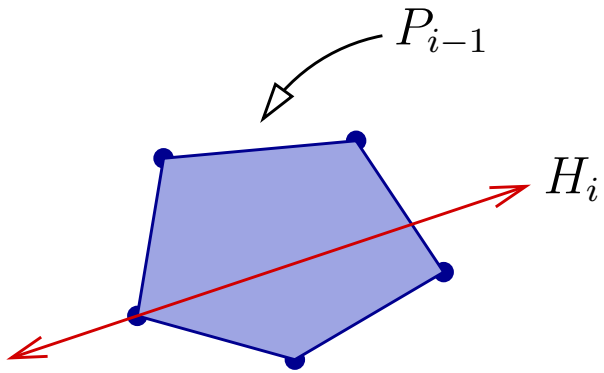
The algorithm is not at fault here—the underlying vertex enumeration problem is known to be **NP-hard** (Dyer, Khachiyan et al.).

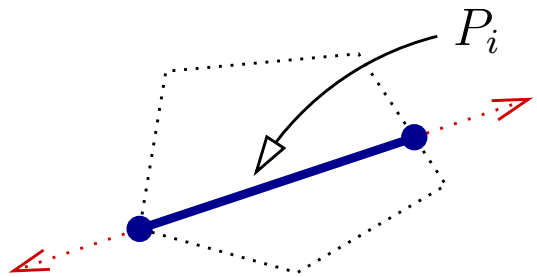
Enumeration: The double description method

The key steps:



- Begin with a starting polytope $P_0 = O \cap J$, which is the unit simplex in $\mathbb{R}^{7n}$.
- Inductively compute the polytope $P_i = P_{i-1} \cap H_i$ for each $i = 1, \ldots, g$.





- The solution is the final polytope P_g.

Each polytope is stored both as an *intersection of hyperplanes and half-spaces*, and as a *convex hull of vertices*.

Enumeration: The double description method (ctd.)

The problem with this algorithm is that the intermediate polytopes P_i can grow extremely large (exponentially many vertices).

This "combinatorial explosion" causes serious problems for both running time and memory usage.

Keeping the problem under control is something of a black art.

We can go a long way with a little extra knowledge:

- We only want solutions that satisfy the **quadrilateral constraints**.
- The matching equations are **sparse**.

Improvement #1: Vertex filtering (Letscher)

Aim to exploit the quadrilateral constraints.

Key idea

At each stage of the double description method, **throw away** every vertex that does not satisfy the quadrilateral constraints.

It is easy to prove correctness using induction on the intermediate polytopes.

Performance:

- Dramatically reduces the size of the intermediate vertex sets.
- Running time and memory are both reduced by orders of magnitude.

Improvement #2: Hyperplane sorting (B.B.)

Aim to exploit the sparseness of the matching equations.

The performance of the double description method is known to be highly sensitive to the **order** in which hyperplanes $H_1, \ldots, H_g$ are used.

Key idea

Sort the matching equations so that, in early stages of the algorithm, the equations involve as few tetrahedra as possible.

By doing this, we hope to force more non-zero coordinates and filter out more vertices as a result.

Performance:

- Consistently outperforms classical orderings from the literature.
- Running time and memory again reduced by orders of magnitude.

Improvement #3: Inner product representation (B.B.)

Aims to reduce memory consumption.

Key idea

Store only "essential information" for vertices:

- the inner products with hyperplanes that we have not yet used;
- a bitmask that indicates which coordinates are set to zero.

This works because each vertex can be reconstructed from the **bitmask alone**. The inner products let us avoid this reconstruction during intermediate stages of the algorithm.

Performance:

- Memory consumption: $\sim 6/7$ of original down to almost zero!
- All intermediate stages run faster.
- The final reconstruction is a negligible one-off cost.

Optimising Jaco-Oertel

(Joint work with J. Hyam Rubinstein and Stephan Tillmann)

Jaco-Oertel is, unfortunately, still infeasibly slow.

- Current algorithms allow us to enumerate vertex normal surfaces of the original triangulation.
  But...
- When we slice along a surface and look for compressing discs, the resulting triangulations can become extremely large, and vertex enumeration is still out of our reach.

Two new optimisations:

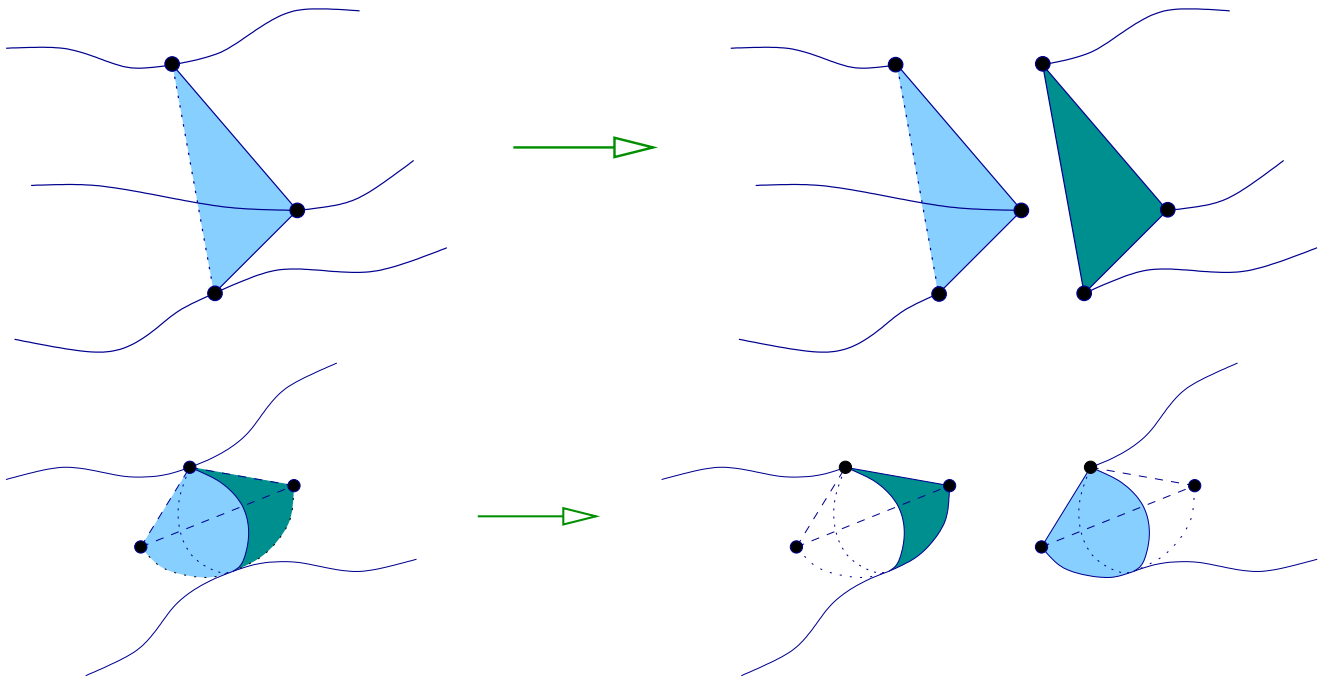
- Heuristic pruning (look for "simple" compressing discs);
- Compatible pairs (incompressible surfaces must come in pairs).

Heuristic pruning

Key idea

Search for compressing discs with simple structures:

- a single face with three boundary edges;
- a disc surrounding an internal edge of degree one.



Requires good algorithms for **simplifying** our triangulations.

Compatible pairs

A 0-efficient triangulation of a closed 3-manifold has no non-trivial normal 2-spheres (Jaco & Rubinstein, 2003).

Theorem (B.B., Rubinstein, Tillmann, 2009)

Let T be a 0-efficient triangulation of a closed, orientable, irreducible Haken manifold.

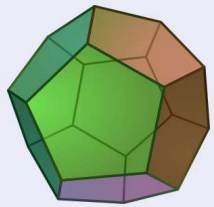
*Then (with the exception of a handful of cases that are easy to test), T must contain **two** distinct incompressible vertex normal surfaces, and these surfaces must be **compatible** (no intersecting quads).*

We prove this using a finger move.

The Weber-Seifert dodecahedral space

Definition (Weber & Seifert, 1933)

The Weber-Seifert dodecahedral space is formed by identifying opposite faces of a dodecahedron with a $3/10$ twist.

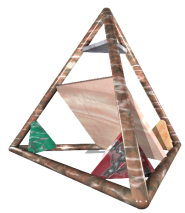


This was one of the first known hyperbolic 3-manifolds.

Theorem (B.B., Rubinstein, Tillmann)

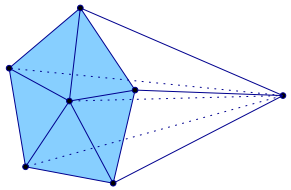
The Seifert-Weber dodecahedral space does not contain a two-sided incompressible surface.

All computations are performed using the software package Regina (`regina.sourceforge.net`).

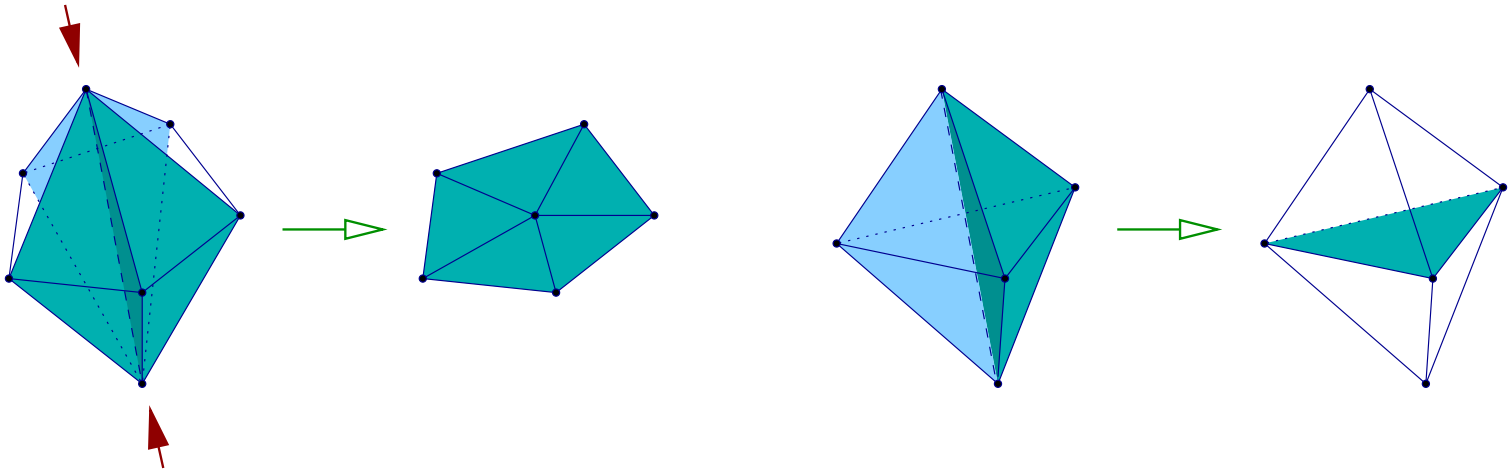


Triangulating the Weber-Seifert dodecahedral space

- Build a regular dodecahedron using twelve pentagonal cones;
- Triangulate each cone with five tetrahedra;
- Identify opposite faces with a 3/10 twist.



This gives 60 tetrahedra. After simplfiying we obtain a 23 tetrahedron, 0-efficient triangulation.



Running Jaco-Oertel

Enumerate vertex normal surfaces:

- 1751 surfaces in total.
- Computation takes $\sim$ 1 second on my laptop.

Ignore neighbourhoods of vertices / edges $\Rightarrow$ 1726 surfaces remain.

Run heuristic pruning:

- Eliminates 1710 surfaces, i.e., **all but 16**.
- Computation takes $\sim$ 40 minutes on my laptop.

Test for compatible pairs:

- No two of these surfaces are pairwise compatible!
- Comptuation is more or less instantaneous.

Therefore **the Weber-Seifert dodecahedral space is non-Haken**.

Further Reading

Theory:

- William Jaco and Ulrich Oertel, *An algorithm to decide if a 3-manifold is a Haken manifold*, Topology **23** (1984), no. 2, 195–209.

Algorithms:

- B.B., *Optimizing the double description method for normal surface enumeration*, Math. Comp. **79** (2010), no. 269, 453–484.

Computation:

- B.B., J. Hyam Rubinstein and Stephan Tillmann, *The Weber-Seifert dodecahedral space is non-Haken*, Preprint, `arXiv:0909.4625`, September 2009.

Software:

- `http://regina.sourceforge.net/`