Using Regina to experiment and compute with 3-manifold triangulations and normal surfaces

Benjamin A. Burton*

1 Introduction

Three-dimensional topology is a fertile ground for algorithmic problems. Prominent amongst these are *decision problems* (e.g., recognising the unknot, or testing whether two triangulated 3-manifolds are homeomorphic); *decomposition problems* (e.g., decomposing a triangulated 3-manifold into a connected sum of prime 3-manifolds); and *recognition problems* (e.g., "naming" the 3-manifold described by a given triangulation).

In three dimensions, such problems typically have highly complex and inefficient solutions—running times are often exponential or super-exponential, and implementations are often major endeavours developed over many years (if they exist at all). This is in contrast to dimension two, where many such problems are easily solved in small polynomial time, and dimensions ≥ 4, where such problems can become undecidable [8, 19].

Regina [2, 6] is a software package for 3-manifold topologists and knot theorists. It aims to provide powerful algorithms and heuristics to assist with decision, decomposition and recognition problems; more broadly, it includes a range of facilities for the study and manipulation of 3-manifold triangulations. It offers rich support for normal surface theory, a major algorithmic framework that recurs throughout 3-manifold topology.

Regina is now 13 years old, with over 175 000 lines of source code. It is released under the GNU General Public License, and contributions from the research community are welcome. It adheres to the following broad development principles, in order of precedence:

1. *Correctness:* Having correct output is critical, particularly since one of Regina's key applications is in computer proofs. For example, it uses arbitrary precision integer arithmetic where it cannot be proven unnecessary, and the API documentation makes thorough use of preconditions and postconditions.

2. *Generality:* Algorithms operate in the broadest possible scenarios (within reason), and do not require preconditions that cannot be easily tested. For instance, unknot recognition runs correctly for both bounded and ideal triangulations, and even when the input triangulation is not known to be a knot complement.

3. *Speed:* Because many of its algorithms run in exponential or super-exponential time, speed is crucial. Regina makes use of sophisticated algorithms and heuristics that, whilst adhering to the constraints of correctness and generality, make it practical for real topological problems.

Regina has featured in a number of topological applications. One recent example is the resolution of Thurston's 30-year old question of whether the Weber-Seifert dodecahedral space is non-Haken [7], using a computer proof that is the cumulation of several decades of theoretical and algorithmic developments by many different authors.

2 Overview of Regina

Regina is multi-platform, and offers a drag-and-drop installer for MacOS, an MSI-based installer for Windows, and ready-made packages for several GNU/Linux distributions. It is thoroughly documented, and stores its data files in a compressed XML format. Regina provides three levels of user interface:

- a full graphical user interface, based on the Qt framework [20];
- a scripting interface based on Python, which can interact with the graphical interface or be used a stand-alone Python module;
- a programmers' interface offering native access to Regina's mathematical core through a C++ shared library.

*School of Mathematics and Physics, The University of Queensland, bab@maths.uq.edu.au

There are facilities to help new users learn their way around, including an illustrated users' handbook, context-sensitive "what's this?" help, and sample data files that can be opened through the *File* ⟶ *Open Example* menu.

Regina's core strengths are in working with triangulations, normal surfaces and angle structures. It only offers basic support for hyperbolic geometries, for which the software packages SnapPea [27] and its successor SnapPy [9] are more suitable. Regina includes implementations of high-level decision and decomposition algorithms, including the only known full implementations of 3-sphere recognition and connected sum decomposition.

For the remainder of this paper we give a short outline of Regina's core features. For a more comprehensive list, see `http://regina.sourceforge.net/docs/featureset.html`.

3 Triangulations

The most basic object in Regina is a *3-manifold triangulation.* Regina does not restrict itself to simplicial complexes; instead it uses *generalised triangulations*, a more general notion that can represent a rich array of 3-manifolds using very few tetrahedra.

Specifically, a triangulation is formed from n tetrahedra by affinely identifying (or "gluing") some or all of their $4n$ faces in pairs. A face is allowed to be identified with another face of the same tetrahedron. It is possible that several edges of a single tetrahedron may be identified together as a consequence of the face gluings, and likewise for vertices. It is common to work with *one-vertex triangulations*, in which all vertices of all tetrahedra become identified as a single point.

Figure 1 illustrates a two-tetrahedron triangulation of the real projective space $\mathbb{R}P^3$. The two tetrahedra are labelled 0 and 1, and the four vertices of each tetrahedron are labelled 0, 1, 2 and 3. Faces 012 and 013 of tetrahedron 0 are joined directly to faces 012 and 013 of tetrahedron 1, creating a solid ball; then faces 023 and 123 of tetrahedron 0 are joined to faces 132 and 032 of tetrahedron 1, effectively gluing the top of the ball to the bottom of the ball with a 180° twist.

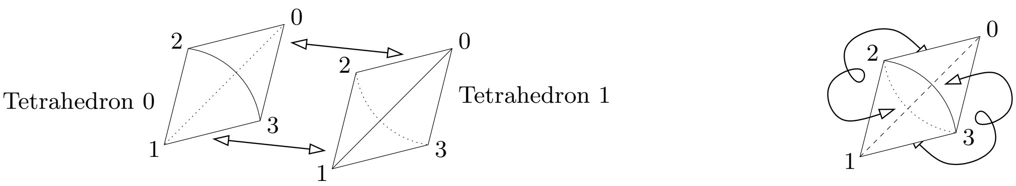

Figure 1: A triangulation of the real projective space $\mathbb{R}P^3$

All of this information can be encoded in a table of face gluings, which is how Regina represents triangulations:

Tetrahedron	Face 012	Face 013	Face 023	Face 123
0	1 (012)	1 (013)	1 (132)	1 (032)
1	0 (012)	0 (013)	0 (132)	0 (032)

Consider the cell in the row for tetrahedron t and the column for face abc. If this cell contains u (xyz), this indicates that face abc of tetrahedron t is identified with face xyz of tetrahedron u, using the affine gluing that maps vertices a, b and c of tetrahedron t to vertices x, y and z of tetrahedron u respectively.

For any vertex V of a triangulation, the *link* of V is defined as the frontier of a small regular neighbourhood of V. It is often useful to think of vertex links as triangulated 2-dimensional surfaces, formed by inserting a small triangle into each corner of each tetrahedron, and then joining together triangles from adjacent tetrahedra along their edges. This mirrors the traditional concept of a link in a simplicial complex, but is modified to support the generalised triangulations that we use in Regina.

The triangulation above of $\mathbb{R}P^3$ is a *closed* triangulation, because it represents a closed 3-manifold; in a closed triangulation, every vertex link is a 2-sphere. Regina can also work with *bounded* triangulations, where one or more tetrahedron faces are not glued to anything; these unglued faces together form the boundary of the resulting 3-manifold. In a bounded triangulation, every vertex link is either a 2-sphere or a disc. The following table describes a one-tetrahedron triangulation of the solid torus $B^2 \times S^1$, whose boundary consists of two triangles (faces 023 and 123 of tetrahedron 0) that together form a 2-dimensional torus:

Tetrahedron	Face 012	Face 013	Face 023	Face 123
0	0 (301)	0 (120)		

Regina can also work with *ideal* triangulations. These are triangulations in which vertex links can be higher-genus closed surfaces (such as tori or Klein bottles). Ideal triangulations can represent non-compact 3-manifolds (by deleting the vertices), or compact bounded 3-manifolds (by truncating the vertices). The following table shows Thurston's famous ideal triangulation of the figure eight knot complement [24]; this triangulation has one vertex, whose link is the torus surrounding the figure eight knot in S^3.

Tetrahedron	Face 012	Face 013	Face 023	Face 123
0	1 (210)	1 (031)	1 (231)	1 (302)
1	0 (210)	0 (031)	0 (231)	0 (302)

There are some triangulations that Regina cannot work with. It will not allow vertex links to be *bounded* surfaces other than discs (so annulus or punctured torus links are bad, for instance), and it will not allow an edge to be identified with itself in reverse. Any triangulation with such a feature will be marked as *invalid*.

Users may enter tetrahedron gluings directly; however, Regina can also create triangulations in other ways, such as importing from SnapPea [27] or other file formats, building "pre-packaged" constructions such as layered lens spaces or Seifert fibred spaces, or accessing large censuses that hold tens of thousands of triangulations of various types.

Regina can also reconstruct triangulations from *isomorphism signatures* [5]. These are short pieces of text that completely encode a triangulation; for instance, the example triangulation of $\mathbb{R}P^3$ above is described by the isomorphism signature `cPcbbbahh`. A feature of isomorphism signatures is that two triangulations are combinatorially isomorphic (i.e., related by a relabelling of tetrahedra and/or their vertices) if and only if their isomorphism signatures are the same. Note that, as a consequence, reconstructing a triangulation from its isomorphism signature may yield a differently-labelled (but isomorphic) copy of the original.

There are many ways to study a 3-manifold triangulation using Regina. At a high level, Regina offers decomposition and recognition routines, including: exact algorithms for 3-sphere recognition, 3-ball recognition and connected sum decomposition (these are always conclusive and correct); heuristic combinatorial algorithms for recognising much larger families of manifolds such as Seifert fibred spaces, surface bundles and graph manifolds (these "recognise" the structure of the triangulation, and will often be inconclusive); and routines from the SnapPea kernel [27] for computing volumes of hyperbolic manifolds. At a lower level, Regina can compute invariants of the underlying 3-manifold (such as homology, fundamental group and Turaev-Viro invariants), as well as combinatorial properties of the specific triangulation (such as computing the 0, 1 and 2-skeleta, or searching for common combinatorial "building blocks" within a triangulation).

Regina can also modify triangulations. Operations include local moves such as bistellar flips and edge collapses, and global operations such as barycentric subdivision, boundary coning and vertex truncation. A frequently-used operation is simplification, in which Regina uses a range of heuristic techniques to retriangulate the given 3-manifold using as few tetrahedra as it can.

4 Normal surfaces

One of Regina's core strengths is its ability to enumerate and work with normal surfaces. A *normal surface* in a 3-manifold triangulation $\mathcal{T}$ is a properly embedded surface in $\mathcal{T}$ that meets each tetrahedron of $\mathcal{T}$ in a (possibly empty) collection of disjoint curvilinear triangles and quadrilaterals, as illustrated in Figure 2.

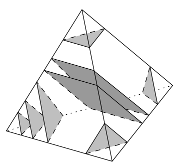

Figure 2: Normal triangles and quadrilaterals in a tetrahedron

Normal surfaces were introduced by Kneser [16] and developed by Haken for use in algorithms [11]. One of their key benefits is that they give significant insights into the structure of the underlying 3-manifold. For instance, any triangulation of $\mathbb{R}P^3$ (as in Figure 1) will contain a normal one-sided projective plane, and any triangulation of the solid torus (i.e., the unknot complement) will contain a normal non-separating compressing disc.

Properties such as this make normal surfaces a powerful tool for high-level recognition and decomposition routines in 3-manifold topology and knot theory. For example, they are central to algorithms for unknot recognition (where one searches for a normal disc that the unknot bounds) [11], connected sum decomposition (where one searches for normal 2-spheres that separates prime factors) [14], 3-sphere recognition [22], Hakenness testing [13], and many more.

Let $\mathcal{T}$ be a 3-manifold triangulation with n tetrahedra. Any normal surface in $\mathcal{T}$ can be described as an integer vector in $\mathbb{R}^{7n}$, whose elements count the number of triangles and quadrilaterals of each type in each tetrahedron. Specifically, for each tetrahedron Δ there are four *triangle coordinates* that count how many triangles sit within each of the four corners of Δ, and three *quadrilateral coordinates* that count how many quadrilaterals pass through Δ in each of the three possible directions. Figure 3 illustrates all seven types of triangle and quadrilateral in Δ.

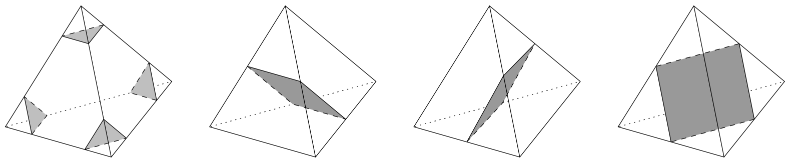

Figure 3: The seven triangle and quadrilateral types within a tetrahedron

An integer vector $\mathbf{x} \in \mathbb{R}^{7n}$ represents a normal surface in $\mathcal{T}$ if and only if: (i) $\mathbf{x} \geq 0$; (ii) $A\mathbf{x} = 0$ for a sparse matrix A of *matching equations* derived from $\mathcal{T}$; and (iii) $\mathbf{x}$ satisfies the *quadrilateral constraints*, which require that at most one quadrilateral coordinate within each tetrahedron can be non-zero. Conditions (i) and (ii) map out a polyhedral cone in $\mathbb{R}^{7n}$, called the *normal surface solution cone*; condition (iii) then maps out a (typically non-convex) union of faces of this cone.

A normal surface whose vector lies on an extreme ray of the normal surface solution cone is called a *vertex normal surface*, and a normal surface whose vector lies in the Hilbert basis of this cone is called a *fundamental normal surface*. For many high-level algorithms (including all of those mentioned earlier), it can be shown that important surfaces—if they exist—can be found as vertex normal surfaces (or for some more difficult algorithms, fundamental normal surfaces). For instance, in any triangulation of a non-prime 3-manifold $\mathcal{M}$, there is some *vertex* normal 2-sphere that separates $\mathcal{M}$ into a connected sum of non-trivial factors. The basic procedure then for a typical high-level algorithm is to enumerate all vertex normal surfaces (or for some problems, all fundamental normal surfaces), and then run some problem-specific test or procedure over each.

Regina comes with heavily optimised algorithms for enumerating all vertex normal surfaces [3] or fundamental normal surfaces [4] in a triangulation. It can enumerate and/or view surfaces in a number of coordinate systems, including *standard coordinates* in $\mathbb{R}^{7n}$ (as outlined above), *quadrilateral coordinates* in $\mathbb{R}^{3n}$ [26] (where we consider only the quadrilaterals in each tetrahedron), and *edge weight space* in $\mathbb{R}^{e}$ (where we count the intersections with each of the e edges of the triangulation). It can also work with octagonal *almost normal surfaces* [23] (used in 3-sphere recognition, where we allow a single octagonal piece), and *spun normal surfaces* [25] (used with ideal triangulations, where we allow infinitely many triangles spinning in towards a vertex).

Regina offers several ways to analyse normal surfaces, both "at a glance" and in detail. It also supports the key operations of cutting a triangulation open along a normal surface and retriangulating, or crushing a surface to a point (in the Jaco-Rubinstein sense [14], where there may be additional changes in topology but which can be controlled and detected).

5 Angle structures

In addition to normal surfaces, Regina can also enumerate and analyse angle structures on a triangulation. An *angle structure* on a 3-manifold triangulation $\mathcal{T}$ assigns non-negative internal dihedral angles to each edge of each tetrahedron of $\mathcal{T}$, so that (i) opposite edges of a tetrahedron are assigned the same angle; (ii) all angles in a tetrahedron sum to 2π; and (iii) all angles around any internal edge of $\mathcal{T}$ likewise sum to 2π (see Figure 4). Such structures are often called *semi-angle structures* [15], to distinguish them from *strict angle structures* in which all angles are strictly positive.

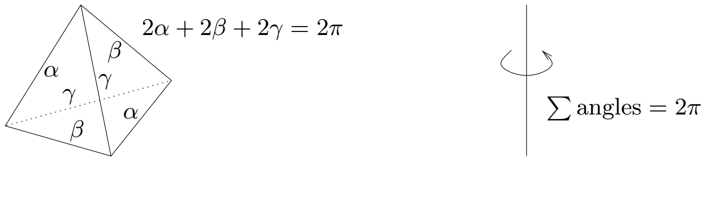

Figure 4: The conditions for an angle structure on a triangulation

Angle structures were introduced by Casson and further developed by Lackenby and Rivin [17, 21], and are a simpler (but weaker) combinatorial analogue of a complete hyperbolic structure. Some angle structures are of particular interest: these include *taut angle structures* in which every angle is precisely 0 or π (representing "flattened" tetrahedra) [12, 18], and *veering structures* which are taut angle structures with powerful combinatorial constraints [1, 12]. Veering structures, and in some settings taut angle structures, can yield strict angle structures [12, 15], which in turn can point the way towards building a complete hyperbolic structure on $\mathcal{T}$ [10].

Regina can construct and analyse angle structures on a 3-manifold triangulation $\mathcal{T}$. Specifically, conditions (i)–(iii) map out a polytope in $\mathbb{R}^{3n}$, where n is the number of tetrahedra; the vertices of this polytope—called *vertex angle structures*—then generate all possible angle structures on $\mathcal{T}$. Regina can enumerate all vertex angle structures, or (optionally) only taut angle structures, and can detect veering structures when they are present.

6 Scripting in Python

Regina offers a powerful scripting facility, whereby most of the C++ classes and functions in its mathematical engine are made available through a dedicated Python module. Python is a popular scripting language that is easy to write and easy to read, and the Python module in Regina makes it easy to quickly prototype new algorithms, run tests over large bodies of census data, or perform complex tasks that would be cumbersome through a point-and-click interface.

Users can access Regina's Python module in two ways:

- by opening a Python console from within the graphical user interface, which allows users to directly study and/or modify data in the current working file;
- by starting the command-line program `regina-python`, which brings up a standalone Python prompt.

Users can also run their own Python scripts directly via `regina-python`, embed scripts within data files as *script packets*, or write their own libraries of frequently-used routines that will be loaded automatically each time a Regina Python session is started.

The following sample Python session constructs the triangulation of $\mathbb{R}P^3$ from Section 3, prints its first homology group, enumerates all vertex normal surfaces, and then locates and prints the coordinates of the vectors that represent vertex normal projective planes.

```
bab@rosemary:~$ regina-python
Regina 4.92
Software for 3-manifold topology and normal surface theory
Copyright (c) 1999-2012, The Regina development team
>>> tri = NTriangulation()
>>> t0 = tri.newTetrahedron()
>>> t1 = tri.newTetrahedron()
>>> t0.joinTo(0, t1, NPerm4(1,0,3,2))       # Glues 0 (123) -> 1 (032)
>>> t0.joinTo(1, t1, NPerm4(1,0,3,2))       # Glues 0 (023) -> 1 (132)
>>> t0.joinTo(2, t1, NPerm4(0,1,2,3))       # Glues 0 (013) -> 1 (013)
>>> t0.joinTo(3, t1, NPerm4(0,1,2,3))       # Glues 0 (012) -> 1 (012)
>>> print tri.getHomologyH1()
Z_2
>>> s = NNormalSurfaceList.enumerate(tri, NNormalSurfaceList.STANDARD, 1)
>>> print s
5 vertex normal surfaces (Standard normal (tri-quad))
>>> for i in range(s.getNumberOfSurfaces()):
...     if s.getSurface(i).getEulerCharacteristic() == 1:
...         print s.getSurface(i)
...
0 0 0 0 ; 0 1 0 || 0 0 0 0 ; 0 1 0
0 0 0 0 ; 0 0 1 || 0 0 0 0 ; 0 0 1
>>>
```

7 Future development

Regina continues to enjoy active development and regular releases. The developers are currently working towards a major version 5.0 release, which will also work with triangulated *4-manifolds* and normal hypersurfaces (much of this code is already running and well-tested in the development repository).

Users are encouraged to contribute code and offer feedback. For information on new releases, interested parties are welcome to subscribe to the low-traffic mailing list `regina-announce@lists.sourceforge.net`.

Acknowledgments. The author is grateful to many people and organisations for contributing to Regina over the years, and to Ryan Budney and William Pettersson in particular; see `http://regina.sourceforge.net/docs/credits.html` for details. The author is supported by the Australian Research Council under the Discovery Projects funding scheme (project DP1094516).

References

[1] I. Agol. Ideal triangulations of pseudo-Anosov mapping tori. In *Topology and Geometry in Dimension Three*, volume 560 of *Contemp. Math.*, pages 1–17. Amer. Math. Soc., Providence, RI, 2011.

[2] B. A. Burton. Introducing Regina, the 3-manifold topology software. *Experiment. Math.*, 13(3):267–272, 2004.

[3] B. A. Burton. Optimizing the double description method for normal surface enumeration. *Math. Comp.*, 79(269):453–484, 2010.

[4] B. A. Burton. Fundamental normal surfaces and the enumeration of Hilbert bases. Preprint, `arXiv:1111.7055`, Nov. 2011.

[5] B. A. Burton. Simplification paths in the Pachner graphs of closed orientable 3-manifold triangulations. Preprint, `arXiv:1110.6080`, Oct. 2011.

[6] B. A. Burton, R. Budney, W. Pettersson, et al. Regina: Software for 3-manifold topology and normal surface theory. `http://regina.sourceforge.net/`, 1999–2012.

[7] B. A. Burton, J. H. Rubinstein, and S. Tillmann. The Weber-Seifert dodecahedral space is non-Haken. *Trans. Amer. Math. Soc.*, 364(2):911–932, 2012.

[8] M. Chiodo. Finding non-trivial elements and splittings in groups. *J. Algebra*, 331:271–284, 2011.

[9] M. Culler, N. M. Dunfield, and J. R. Weeks. SnapPy, a computer program for studying the geometry and topology of 3-manifolds. `http://snappy.computop.org/`, 1991–2011.

[10] D. Futer and F. Guéritaud. From angled triangulations to hyperbolic structures. In *Interactions Between Hyperbolic Geometry, Quantum Topology and Number Theory*, volume 541 of *Contemp. Math.*, pages 159–182. Amer. Math. Soc., Providence, RI, 2011.

[11] W. Haken. Theorie der Normalflächen. *Acta Math.*, 105:245–375, 1961.

[12] C. D. Hodgson, J. H. Rubinstein, H. Segerman, and S. Tillmann. Veering triangulations admit strict angle structures. *Geom. Topol.*, 15(4):2073–2089, 2011.

[13] W. Jaco and U. Oertel. An algorithm to decide if a 3-manifold is a Haken manifold. *Topology*, 23(2):195–209, 1984.

[14] W. Jaco and J. H. Rubinstein. 0-efficient triangulations of 3-manifolds. *J. Differential Geom.*, 65(1):61–168, 2003.

[15] E. Kang and J. H. Rubinstein. Ideal triangulations of 3-manifolds II; Taut and angle structures. *Algebr. Geom. Topol.*, 5:1505–1533, 2005.

[16] H. Kneser. Geschlossene Flächen in dreidimensionalen Mannigfaltigkeiten. *Jahresbericht der Deut. Math. Verein.*, 38:248–260, 1929.

[17] M. Lackenby. Word hyperbolic Dehn surgery. *Invent. Math.*, 140(2):243–282, 2000.

[18] M. Lackenby. Taut ideal triangulations of 3-manifolds. *Geom. Topol.*, 4:369–395, 2000.

[19] A. A. Markov. Insolubility of the problem of homeomorphy. In *Proc. Internat. Congress Math. 1958*, pages 300–306. Cambridge Univ. Press, New York, 1960.

[20] Nokia Corporation. Qt – cross-platform application and UI framework. `http://qt.nokia.com/`, 1992–2012.

[21] I. Rivin. Combinatorial optimization in geometry. *Adv. in Appl. Math.*, 31(1):242–271, 2003.

[22] J. H. Rubinstein. An algorithm to recognize the 3-sphere. In *Proceedings of the International Congress of Mathematicians (Zürich, 1994)*, volume 1, pages 601–611. Birkhäuser, 1995.

[23] A. Thompson. Thin position and the recognition problem for S^3. *Math. Res. Lett.*, 1(5):613–630, 1994.

[24] W. P. Thurston. The geometry and topology of 3-manifolds. Lecture notes, Princeton University, 1978.

[25] S. Tillmann. Normal surfaces in topologically finite 3-manifolds. *Enseign. Math. (2)*, 54:329–380, 2008.

[26] J. L. Tollefson. Normal surface Q-theory. *Pacific J. Math.*, 183(2):359–374, 1998.

[27] J. R. Weeks. SnapPea: Hyperbolic 3-manifold software. `http://www.geometrygames.org/SnapPea/`, 1991–2000.