A New Approach to Crushing 3-Manifold Triangulations*

Benjamin A. Burton
School of Mathematics
and Physics
The University of Queensland
Brisbane QLD 4072, Australia
bab@maths.uq.edu.au

ABSTRACT

The crushing operation of Jaco and Rubinstein is a powerful technique in algorithmic 3-manifold topology: it enabled the first practical implementations of 3-sphere recognition and prime decomposition of orientable manifolds, and it plays a prominent role in state-of-the-art algorithms for unknot recognition and testing for essential surfaces. Although the crushing operation will always reduce the size of a triangulation, it might alter its topology, and so it requires a careful theoretical analysis for the settings in which it is used.

The aim of this paper is to make the crushing operation more accessible to practitioners, and easier to generalise to new settings. When the crushing operation was first introduced, the analysis was powerful but extremely complex. Here we give a new treatment that reduces the crushing process to a sequential combination of three "atomic" operations on a cell decomposition, all of which are simple to analyse. As an application, we generalise the crushing operation to the setting of non-orientable 3-manifolds, where we obtain a new practical and robust algorithm for non-orientable prime decomposition.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Geometrical problems and computations*

Keywords

Computational topology, 3-manifold triangulations, normal surfaces, 0-efficiency

1. INTRODUCTION

Algorithms in computational 3-manifold topology often exhibit an enormous gap between theory and practice. Theoretical solutions are now known for a large number of difficult topological problems in three dimensions, ranging from smaller problems such as recognising the unknot [12] or recognising the 3-sphere [20] through to decomposition into geometric pieces [17] and of course the full homeomorphism (or "topological equivalence") problem [13, 14, 19]. Although these results are of high significance to the mathematical community, many remain algorithms in theory only—such algorithms are often far too intricate to implement, and far too slow to run.

In the last decade, however, there has been strong progress in the realm of practical, usable algorithms on 3-manifolds. For instance, there are now practical implementations of unknot recognition, 3-sphere recognition and orientable prime decomposition [1, 2], and recently more complex algorithms such as testing for closed essential surfaces have become viable [6, 8].

A key component in many of these practical algorithms is the *crushing procedure* of Jaco and Rubinstein [15]. This procedure was developed as part of their theory of *0-efficiency*, and operates in the context of normal surface theory, a common algorithmic toolkit for 3-manifold topologists. In essence, the crushing process modifies a triangulation to eliminate "unwanted" normal spheres and discs, whereupon the resulting triangulation is called *0-efficient* (we give a more precise definition shortly). This brings both theoretical and practical advantages: 0-efficient triangulations are typically smaller and easier to study, and algorithms upon them are easier to formulate—often significantly so [15, 16, 18, 21].

Although the full process of obtaining a 0-efficient triangulation requires worst-case exponential time, recent techniques based on combinatorial optimisation have made this extremely fast in a range of experimental settings [6, 7]. A notable application of crushing has been in 3-sphere recognition: here the introduction of Jaco and Rubinstein's 0-efficiency techniques was a major turning point that made 3-sphere recognition practical to implement for the first time [4, 15].

In summary: normal surface theory makes difficult 3-manifold problems *decidable*, whereas crushing and 0-efficiency often play a key role in making the resulting algorithms *practical*. It is therefore important for practitioners in computational 3-manifold topology to understand crushing and 0-efficiency, and to be able to apply them to new settings. The aims of this paper are (i) to make the crushing operation more accessible to the wider computational topology community, (ii) to simplify its analysis so that the techniques are easier to use and generalise, and (iii) to apply this simplified analysis to the non-orientable setting, yield-

*A full version of this paper is available at `arXiv:1212.1441`.

Author's self-archived version
Available from http://www.maths.uq.edu.au/~bab/papers/

The original version of this paper was published in SCG '13: Proceedings of the 29th Annual Symposium on Computational Geometry, ACM, 2013, pp. 415–424.

ing a new practical and robust algorithm for non-orientable prime decomposition.

In detail: the crushing procedure eliminates unwanted normal spheres and discs from a triangulation by cutting the manifold open along them, collapsing the resulting spheres or discs on the boundary to points, and then further "flattening" the resulting cell decomposition until we once again obtain a (different) triangulation. Importantly, this crushing procedure (i) is simple to implement, and (ii) always simplifies the triangulation by reducing the number of tetrahedra (neither of which are true for the related operation of cutting along a normal surface and retriangulating). The downside is that crushing could change the topology of the underlying 3-manifold in unintended ways, and so this crushing process is "destructive"; however, the possible changes are often both simple and detectable.

One difficulty with Jaco and Rubinstein's original paper is that, although their techniques are extremely powerful, the accompanying analysis is extremely complex: they study the potential effects of the crushing procedure through a series of detailed arguments as they collapse chains of truncated prisms and product regions throughout the triangulation. A second difficulty is that their analysis is restricted to orientable 3-manifolds only.

In Section 2 of this paper we both simplify and generalise these arguments. The key result is Lemma 2 (the *crushing lemma*), which shows that—after the initial act of cutting along and collapsing the original normal surface—the entire Jaco-Rubinstein crushing procedure can be expressed as a sequential combination of three local atomic operations on a cell decomposition: flattening a triangular or bigonal pillow to a face, and flattening a bigon face to an edge. Therefore, to analyse the "destructive" consequences of crushing in any given setting, we merely need to examine what can happen independently under each of these atomic operations. All three operations are simple to analyse: Lemma 3 lists the possible consequences of each operation, and Corollary 4 packages these together to describe the overall effects of the full crushing process.

We emphasise that these results are general. We never assume orientability, and all results apply to compact manifolds both with or without boundary. Moreover, the key crushing lemma applies equally well to *ideal triangulations*, which triangulate non-compact manifolds by allowing vertices whose links are higher genus surfaces. The analysis of atomic operations in Lemma 3 and Corollary 4 is also straightforward in the ideal case, but the consequences of crushing become more numerous, and so in this short paper we restrict this latter analysis to triangulations of compact manifolds only.

In Section 3 we apply our results to develop the first practical algorithm for computing the prime decomposition of 3-manifolds to encompass both the orientable and non-orientable cases.[1]

For orientable manifolds, a modern implementation of the prime decomposition algorithm works by repeatedly crushing away normal spheres using the Jaco-Rubinstein procedure, and then "reading off" prime summands from the resulting collection of disconnected triangulations (some summands will have disappeared but we can restore these using homology). It is simple to discard trivial (3-sphere) summands, since the efficiency-based 3-sphere recognition algorithm dovetails into this procedure naturally. The blueprint for this algorithm is laid out in the original 0-efficiency paper [15]; see [4] for a modern "ready to implement" version.

For non-orientable manifolds, however, the current situation is much worse: the only available algorithm is the older Jaco-Tollefson method [17], where we must build a collection of disjoint embedded 2-spheres within the input triangulation using a complex series of cut-and-paste operations, and then cut along these 2-spheres and retriangulate to obtain the individual summands (an expensive operation that could vastly increase the number of tetrahedra). Detecting trivial summands is also significantly more complex to implement in this setting.

In Section 3 we bring the non-orientable algorithm in line with its simpler orientable cousin: using the new generalised results of Section 2, we show that one can crush away normal spheres and then "read off" the summands (again restoring missing summands via homology). There is an important complication: if the input manifold contains an embedded two-sided projective plane then the Jaco-Rubinstein crushing process could fail. We show that even in this setting, we can still run the algorithm: it might still succeed, and if it does fail due to a two-sided projective plane then we obtain a simple certificate alerting us to this fact (this is the sense in which the algorithm is "robust").

Beyond its theoretical contributions, the results of this paper are important for practitioners. In particular, the non-orientable prime decomposition algorithm of Section 3 will soon appear in the software package *Regina* [5]. In the full version of this paper, we also give another application of these techniques, in which we use 0-efficiency and crushing to study the combinatorial properties of non-orientable minimal triangulations.

For the special case of closed orientable manifolds, Fowler describes a different approach to simplifying 0-efficiency arguments using *spines*, elaborating on an earlier argument of Casson [11]. See also Matveev's book [19], which includes a more general (but also more complex) discussion of cutting along normal surfaces in the setting of special and almost special spines.

1.1 Preliminaries

As is common in computational 3-manifold topology, we work not with simplicial complexes but smaller and more flexible structures. A *generalised triangulation* $\mathcal{T}$ is defined to be a collection of n abstract tetrahedra, some or all of whose $4n$ faces are affinely identified (or "glued together") in pairs. The underlying topological space is often (but not always) a 3-manifold $\mathcal{M}$, in which case we say that $\mathcal{T}$ *triangulates* $\mathcal{M}$.

In a generalised triangulation, we allow two faces of the same tetrahedron to be identified. Moreover, as a consequence of the face identifications, we might find that several edges of a tetrahedron become identified, and likewise with vertices. The *link* of a vertex V in a triangulation is the surface obtained as the frontier of a small regular neighbourhood of V.

For convenience, we define several useful (and mutually

[1] Recall that *prime decomposition* asks us to decompose a given 3-manifold into a connected sum of prime 3-manifolds. The *connected sum* $M \# N$ of two manifolds M and N is formed by removing a small ball from each summand and gluing the summands together along the resulting sphere boundaries.

exclusive) subclasses of generalised triangulations. A *closed triangulation* is one that triangulates a closed 3-manifold (here every tetrahedron face must be glued to some partner, and every vertex link must be a sphere). A *bounded triangulation* is one that triangulates a compact 3-manifold with non-empty boundary (here one or more tetrahedron faces are left unglued, and all vertex links must be spheres or discs). An *ideal triangulation* is one in which every tetrahedron face is glued to some partner, but some vertices have links that are not spheres (e.g., tori, Klein bottles, or other higher-genus surfaces). Ideal triangulations are used to represent non-compact 3-manifolds by removing the tetrahedron vertices; a famous example is Thurston's 2-tetrahedron ideal triangulation of the figure eight knot complement [22].

In this paper we modify triangulations to obtain more general *cell decompositions*. Informally, these are natural extensions of generalised triangulations that allow 3-cells other than tetrahedra. A cell decomposition begins with a collection of abstract 3-cells, which are topological 3-balls whose boundaries are decomposed into curvilinear polygonal faces (in particular, we allow small 2-faces such as bigons, and we allow small 3-cells that are "pillows" bounded by a pair of opposite 2-faces). Moreover, we endow each edge on the boundary of each 3-cell with an affine structure (i.e., a homeomorphism from the edge to the interval $[0, 1]$). We explicitly list the possible types of 3-cell as we encounter them in this paper, and so we do not go into further detail here.

To form a cell decomposition, we identify (or "glue" together) some or all of the 2-faces of these 3-cells in pairs, using homeomorphisms that map edges to edges and vertices to vertices, and that restrict to affine maps on the edges. This generalises the affine maps between 2-faces that we use for triangulations. As with triangulations, we allow two 2-faces of the same 3-cell to be identified. For a concrete example of a cell decomposition, see Definition 1 below.

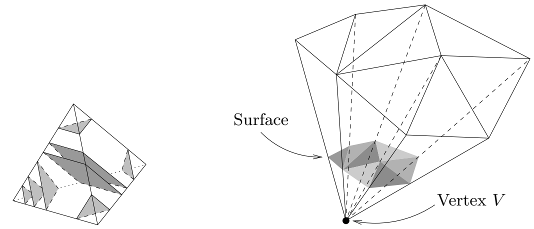
We define an *invalid edge* of a generalised triangulation or cell decomposition to be one that (as a consequence of the 2-face gluings) becomes identified with itself in reverse. A triangulation or cell decomposition is called *valid* if it does not contain an invalid edge. The underlying topological space of an invalid triangulation or cell decomposition cannot be a 3-manifold, since a small regular neighbourhood of the midpoint of an invalid edge will be bounded by $\mathbb{R}P^2$.

A *normal surface* in a generalised triangulation $\mathcal{T}$ is a properly embedded surface[2] in $\mathcal{T}$ that meets each tetrahedron in a (possibly empty) collection of curvilinear triangles and quadrilaterals, as illustrated in Figure 1(a). A *vertex linking surface* (also called a *trivial surface*) is a connected normal surface formed entirely from triangles; any such surface must surround some vertex V of the triangulation as illustrated in Figure 1(b), and effectively triangulates the link of V.

The concept of *0-efficiency* is defined as follows [15]. If $\mathcal{T}$ is a closed or ideal triangulation, then we call $\mathcal{T}$ 0-efficient if and only if it contains no non-trivial normal spheres. If $\mathcal{T}$ is a bounded triangulation, then we call $\mathcal{T}$ 0-efficient if and only if it contains no non-trivial normal discs.[3]

[2] A surface S is *properly embedded* in a triangulation $\mathcal{T}$ if S has no self-intersections, and the boundary of S is precisely where S meets the boundary of $\mathcal{T}$.

[3] Jaco and Rubinstein show that, under appropriate assumptions, if $\mathcal{T}$ has no non-trivial normal discs then $\mathcal{T}$ must have no non-trivial normal spheres also [15, Proposition 5.15].



(a) Triangles and quadrilaterals (b) A vertex linking surface

Figure 1: Normal surfaces within a triangulation

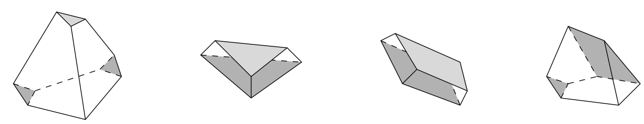


Figure 2: Examples of cells obtained after cutting open along S

Jaco and Rubinstein describe a "destructive" crushing procedure which they use for many purposes, such as creating 0-efficient triangulations of manifolds, and decomposing orientable manifolds into connected sums. This procedure is the main focus of this paper, and we describe it now in detail.

Definition 1. Let S be a normal surface in some generalised triangulation $\mathcal{T}$. The *Jaco-Rubinstein crushing procedure* operates on S as follows:

1. We cut $\mathcal{T}$ open along the normal surface S. This converts the triangulation into a cell decomposition with a large variety of possible cell types (such as truncated tetrahedra, triangular or quadrilateral prisms, and truncated triangular prisms, some of which are illustrated in Figure 2). If S is two-sided in $\mathcal{T}$ then we obtain two new copies of S on the boundary of this cell decomposition, and if S is one-sided in $\mathcal{T}$ then we obtain one new copy of the double cover of S on the boundary.

2. We then collapse (or "shrink") each copy of S on the boundary to a point (using the quotient topology), as illustrated in Figure 3. Specifically, if S was two-sided in $\mathcal{T}$ then we collapse the two copies of S on the boundary to two points, and if S was one-sided in $\mathcal{T}$ then we collapse the double cover of S on the boundary to one point. This converts the triangulation into a cell decomposition $\mathcal{C}$, with cells of the following types:

    - *3-sided footballs*, illustrated in Figure 4(a), which we obtain from regions of $\mathcal{T}$ between two parallel triangles of S, or between a triangle of S and a tetrahedron vertex;

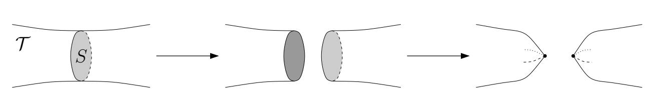
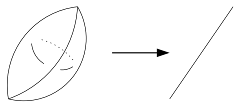
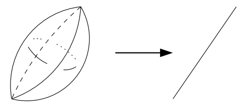


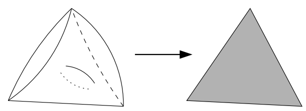
Figure 3: Collapsing copies of S on the boundary to points



(a) A 3-sided football



(b) A 4-sided football



(c) A triangular purse

Figure 4: Destructively flattening non-tetrahedron cells

- *4-sided footballs*, illustrated in Figure 4(b), which we obtain from regions of $\mathcal{T}$ between two parallel quadrilaterals of S;
- *triangular purses*, illustrated in Figure 4(c), which we obtain from regions of $\mathcal{T}$ between a quadrilateral of S and nearby triangles or tetrahedron vertices;
- *tetrahedra*, which we obtain from the central regions of tetrahedra in $\mathcal{T}$ that do not contain any quadrilaterals of S.

3. We next eliminate any non-tetrahedron cells, as illustrated in Figure 4, by simultaneously flattening all footballs to edges, and flattening all triangular purses to triangular faces (again using the quotient topology). Note that all remaining tetrahedron cells are preserved in this step (the flattening operations only affect cells with bigon faces).

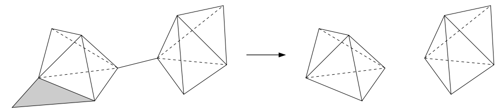
   We can now "read off" a resulting generalised triangulation $\mathcal{T}_{\mathrm{JR}}$, which is defined *only* by the surviving tetrahedra and the resulting identifications between their 2-dimensional faces. In particular:

   - Any triangles, edges and/or vertices that do not belong to a tetrahedron are removed entirely, as illustrated in Figure 5(a). We might even lose entire connected components in this way.
   - If different pieces of a triangulation are connected along pinched edges or vertices then these pieces will "fall apart", as illustrated in Figure 5(b) (since there are no 2-dimensional faces holding them together).

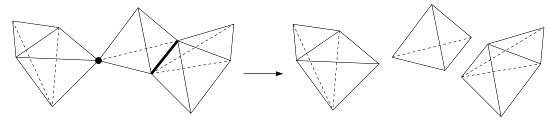
The final result of the crushing procedure is this generalised triangulation $\mathcal{T}_{\mathrm{JR}}$. Note that this might be disconnected, empty or invalid, and might not even represent a 3-manifold.

For convenience, we refer to steps 1 and 2 as *non-destructive crushing* (yielding the cell complex $\mathcal{C}$), and all three steps 1–3 as *destructive crushing* (yielding the generalised triangulation $\mathcal{T}_{\mathrm{JR}}$). Unless otherwise specified, "crushing" always refers to the full destructive operation.

We observe that each original tetrahedron Δ of $\mathcal{T}$ can only give rise to at most one tetrahedron of $\mathcal{T}_{\mathrm{JR}}$ (and only if Δ contains no quadrilaterals of S). It follows that $\mathcal{T}_{\mathrm{JR}}$ has strictly fewer tetrahedra than $\mathcal{T}$, unless S is a union of vertex



(a) Edges or faces not in a tetrahedron will disappear



(b) Tetrahedra joined along pinched edges or vertices will fall apart

Figure 5: Triangulations are defined by face gluings only

linking surfaces, in which case $\mathcal{T}_{\mathrm{JR}}$ and $\mathcal{T}$ are isomorphic (i.e., the crushing procedure has no effect).

Note that the intermediate cell decomposition $\mathcal{C}$ and the final triangulation $\mathcal{T}_{\mathrm{JR}}$ are well-defined. In particular, the flattening operations above can never result in any 2-face of a cell being identified with more than one partner 2-face, and the final triangulation $\mathcal{T}_{\mathrm{JR}}$ is independent of any "order" in which we flatten the footballs and/or purses of $\mathcal{C}$.

A key property of this procedure (which we generalise in Corollary 4) is that, for orientable manifolds, any "destructive" changes are both limited and detectable:

THEOREM 1 (JACO AND RUBINSTEIN [15]). *Let $\mathcal{T}$ be a generalised triangulation of a compact orientable 3-manifold $\mathcal{M}$ (with or without boundary), and let S be a normal sphere or disc in $\mathcal{T}$. Then, if we destructively crush S using the Jaco-Rubinstein procedure, we obtain a valid generalised triangulation $\mathcal{T}_{\mathrm{JR}}$ whose underlying 3-manifold $\mathcal{M}_{\mathrm{JR}}$ is obtained from $\mathcal{M}$ by zero or more of the following operations:*

- *undoing connected sums, i.e., replacing some intermediate manifold $\mathcal{M}'$ with the disjoint union $\mathcal{M}'_1 \cup \mathcal{M}'_2$, where $\mathcal{M}' = \mathcal{M}'_1 \# \mathcal{M}'_2$;*
- *cutting open along properly embedded discs;*
- *filling boundary spheres with 3-balls;*
- *deleting 3-ball, 3-sphere, $\mathbb{R}P^3$, $L_{3,1}$ or $S^2 \times S^1$ components.*[4]

For reference, Jaco and Rubinstein do not present Theorem 1 in this unified form—the full theorem statement above collects the results of several detailed arguments from throughout their original paper [15].

We emphasise again that all generalised triangulations and cell decompositions in this paper are defined entirely by their 3-cells and the pairwise identifications between the 2-faces of these 3-cells. In particular, if we modify a cell decomposition so that some edge or 2-face does not belong to a 3-cell then that edge or 2-face will disappear (as in Figure 5(a)), and if different pieces of the cell decomposition become connected along pinched edges or vertices then those pieces will fall apart (as in Figure 5(b)).

[4]Regarding notation: $\mathbb{R}P^3$ denotes real projective space, $L_{3,1}$ is a lens space, and $S^2 \times S^1$ is the product space of the 2-sphere and the circle.

2. THE CRUSHING LEMMA

In this section we present our "atomic" formulation of the Jaco-Rubinstein crushing procedure. We begin with the crushing lemma (Lemma 2), which establishes the sufficiency of our three atomic operations, and shows that they can be performed sequentially (as opposed to simultaneously). Lemma 3 then analyses the precise behaviour of each operation on a compact manifold, and Corollary 4 uses this to prove a generalisation of Theorem 1 that covers both orientable and non-orientable manifolds.

We emphasise that the crushing lemma is completely general: the triangulation may be non-orientable, or ideal, or even invalid. In this sense, the crushing lemma is intended as a launching point for generalising crushing and 0-efficiency technology to a wide range of settings (such as ideal triangulations, which we do not pursue in detail in this short paper).

The proof of the crushing lemma uses an algorithmic approach: we show how the full crushing procedure can be performed one atomic operation at a time. We note that this algorithm is intended to assist with the theoretical analysis, not the implementation; a practical implementation could simply flatten non-tetrahedron cells "in bulk".[5]

Lemma 2 (Crushing lemma). *Let $\mathcal{T}$ be a generalised triangulation containing a normal surface S. Let $\mathcal{C}$ be the cell decomposition obtained by non-destructively crushing S, as described in steps (1)–(2) of Definition 1, and let $\mathcal{T}_{\mathrm{JR}}$ be the final triangulation obtained at the end of the destructive crushing procedure, after flattening away all non-tetrahedron cells in step (3) of Definition 1. Then $\mathcal{T}_{\mathrm{JR}}$ can be obtained from $\mathcal{C}$ by a sequence of zero or more of the following atomic operations, one at a time, in some order:*

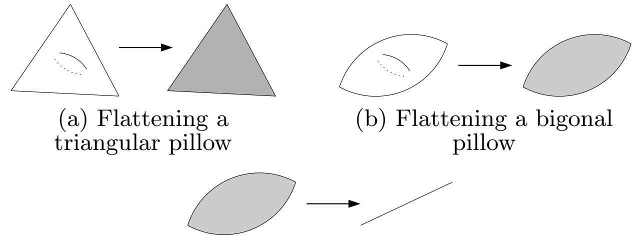
- *flattening a triangular pillow to a triangular face, as shown in Figure 6(a);*
- *flattening a bigonal pillow to a bigon face, as shown in Figure 6(b);*
- *flattening a bigon face to an edge, as shown in Figure 6(c).*

As in Definition 1, after each atomic operation we remove any "orphaned" 2-faces, edges or vertices that do not belong to a 3-cell, and we pull apart any pieces of the cell decomposition that are connected along pinched edges or vertices (as in Figure 5).

Note that each atomic operation might be performed several times, and that multiple instances of one operation might be interspersed with instances of the others.

Proof. We outline the main ideas here; see the full version of this paper for complete details of the proof.

The crushing process requires us to simultaneously flatten footballs to edges and purses to triangles; our task now is to find a good *order* in which to flatten these cells that allows us to express this process as a sequence of atomic operations, as described above. The key complication is that local moves on one cell might change the shapes of adjacent cells, and so we must choose our ordering carefully to avoid creating any unexpected new cell types.



(a) Flattening a triangular pillow (b) Flattening a bigonal pillow (c) Flattening a bigon face

Figure 6: Atomic moves for the Jaco-Rubinstein crushing procedure

To make our task easier, we introduce three intermediate cell types: bigonal pyramids, triangular pillows, and bigonal pillows. This allows us fine-grained control over the crushing process—for instance, to flatten a 3-sided football to an edge we could first (i) flatten one of its bigon faces to create a bigonal pillow, then (ii) flatten this bigonal pillow to yield a bigon face, and finally (iii) flatten this bigon face to an edge. Note that each of these steps is an atomic operation from the lemma statement.

In the detailed proof, we present an explicit algorithm that builds an ordering of atomic operations that only ever creates cells of the initial and intermediate types that we have explicitly described. Termination is guaranteed because each atomic operation strictly reduces the number of non-tetrahedron cells plus the number of bigon faces. Again, see the full version of this paper for details. □

One might observe that the crushing lemma simply replaces the three original moves of Figure 4 with the three atomic moves of Figure 6. Nevertheless, this brings important advantages:

- The new atomic moves operate on smaller subcomplexes (triangular pillows, bigonal pillows and bigon faces), which means fewer special cases or unusual behaviours to analyse.
- More importantly, our new atomic moves can be performed *sequentially*, and can therefore be studied individually as *local operations*. The original flattening moves of Figure 4 must be done *simultaneously* (otherwise we introduce many additional cell types each with their own moves and analyses), which means the original flattening moves must be studied as a complex *global operation* (as Jaco and Rubinstein do in their original paper).
- We extend our analysis to more general settings, such as non-orientable and arbitrary ideal triangulations.

From this point onwards we restrict our attention to compact manifolds (i.e., closed or bounded triangulations), and study the possible outcomes of each of our three atomic moves.

Lemma 3. *Let $\mathcal{C}$ be a cell decomposition of a compact 3-manifold $\mathcal{M}$ (with or without boundary) that contains no two-sided projective planes. Then applying one of the atomic*

[5]The reader is invited to peruse *Regina*'s source code [5] to see how this can be done; see the function `NNormalSurface::crush()`.

moves of Lemma 2 will yield a (valid) cell decomposition of a 3-manifold $\mathcal{M}'$, where either $\mathcal{M}' = \mathcal{M}$, or else $\mathcal{M}'$ is obtained from $\mathcal{M}$ by one of the following operations:

- *If we flattened a triangular pillow, then $\mathcal{M}'$ might remove a single connected 3-ball, 3-sphere or $L_{3,1}$ component from $\mathcal{M}$;*
- *If we flattened a bigonal pillow, then $\mathcal{M}'$ might remove a single connected 3-ball, 3-sphere or $\mathbb{R}P^3$ component from $\mathcal{M}$;*
- *If we flattened a bigon face, then (i) $\mathcal{M}'$ might be obtained by cutting $\mathcal{M}$ open along a properly embedded disc, (ii) $\mathcal{M}'$ might be obtained by filling one boundary sphere of $\mathcal{M}$ with a 3-ball; (iii) $\mathcal{M}'$ might be obtained by cutting $\mathcal{M}$ open along an embedded sphere and filling the two resulting boundary spheres with 3-balls; or (iv) we might have $\mathcal{M} = \mathcal{M}' \# \mathbb{R}P^3$; that is, $\mathcal{M}'$ might remove a single $\mathbb{R}P^3$ summand from the connected sum decomposition of $\mathcal{M}$.*

Proof. This is simply a matter of enumerating the possible consequences of each atomic move, and the details of the argument are laid out carefully in the full version of this paper. In summary:

- The pillow moves are easiest to handle: if they change the underlying topology then either both faces of the pillow are identified or both faces are boundary. Either way, the pillow forms its own separate connected component of the triangulation, and so the pillow move must delete one of the few types of connected component that can be constructed in this way.
- Flattening bigon faces is more delicate: here topological changes arise if both edges of the bigon are identified or both edges are boundary. There are several cases to consider, each of which corresponds to cutting along the bigon and/or eliminating bigon boundaries; together these yield the list of changes (i)–(iv) above. ☐

Now that we understand the possible behaviour of each atomic move, we can aggregate this information to understand the Jaco-Rubinstein crushing procedure as a whole. In the following result we do this for arbitrary compact manifolds, thereby generalising Theorem 1 to both orientable and non-orientable settings. As in the previous lemma, we exclude two-sided projective planes (which can lead to invalid edges); however, even this exclusion can be partially overcome as we see later in Section 3.

Corollary 4. *Let $\mathcal{T}$ be a generalised triangulation of a compact 3-manifold $\mathcal{M}$ (with or without boundary) that contains no two-sided projective planes, and let S be a normal sphere or disc in $\mathcal{T}$. Then, if we destructively crush S using the Jaco-Rubinstein procedure, we obtain a valid triangulation $\mathcal{T}_{\mathrm{JR}}$ whose underlying 3-manifold $\mathcal{M}_{\mathrm{JR}}$ is obtained from $\mathcal{M}$ by zero or more of the following operations:*

- *undoing connected sums, i.e., replacing some intermediate manifold $\mathcal{M}'$ with the disjoint union $\mathcal{M}'_1 \cup \mathcal{M}'_2$, where $\mathcal{M}' = \mathcal{M}'_1 \# \mathcal{M}'_2$;*
- *cutting open along properly embedded discs;*
- *filling boundary spheres with 3-balls;*
- *deleting 3-ball, 3-sphere, $\mathbb{R}P^3$, $L_{3,1}$, $S^2 \times S^1$ or twisted $S^2 \tilde{\times} S^1$ components.*

Proof. This follows immediately from Lemmata 2 and 3, and the fact that the *non-destructive* act of crushing S to a point (which precedes the sequence of atomic moves) has the effect of either undoing a connected sum (if S is a separating sphere), removing an $S^2 \times S^1$ or twisted $S^2 \tilde{\times} S^1$ summand from the connected sum decomposition (if S is a non-separating sphere), or cutting along a properly embedded disc (if S is a disc). ☐

3. NON-ORIENTABLE PRIME DECOMPOSITION

We finish this paper with an application of our results: a modern approach to prime decomposition of non-orientable manifolds based on the crushing process.

In 1995, Jaco and Tollefson described an algorithm that, given a closed 3-manifold triangulation $\mathcal{T}$, decomposes the underlying manifold into a connected sum of prime manifolds [17]. In essence, it involves the following steps:

1. Enumerate all *vertex normal spheres* in $\mathcal{T}$. These are normal spheres that are represented by extreme rays of a high-dimensional polyhedral cone derived from the triangulation $\mathcal{T}$; see [17] for a precise definition.
2. Convert these into a (possibly much larger) collection of pairwise disjoint embedded spheres in $\mathcal{T}$ using an intricate series of cut-and-paste operations.
3. Cut $\mathcal{T}$ open along these embedded spheres, retriangulate, and fill the boundaries with balls to obtain the final list of irreducible summands.

Despite its theoretical importance, the Jaco-Tollefson algorithm is both slow and complex. Step 1 requires us to enumerate *all* vertex normal spheres (of which there could be exponentially many), which prevents us from using highly effective optimisations based on linear programming [7]. Step 2 is extremely complex to implement, and could significantly increase the number of spheres under consideration (which is already exponential in the number of tetrahedra). Likewise, the cut-open-and-retriangulate operation of step 3 is highly intricate to implement, and could vastly increase the number of tetrahedra in the final collection of triangulated summands.

For orientable triangulations, the Jaco-Rubinstein theory of 0-efficiency from 2003 simplified this algorithm enormously [15]. In brief, the new procedure is:

1. Locate *any* non-trivial normal sphere in the triangulation, and destructively crush this to obtain a new (possibly disconnected) triangulation with strictly fewer tetrahedra. Repeat this step for as long as a non-trivial normal sphere can be found.
2. Once no non-trivial normal spheres exist (i.e., the remaining triangulation is 0-efficient), each connected component of the triangulation will represent a single prime summand. There may be additional "missing" summands that were lost, but these can be reconstructed by tracking changes in homology.

This 0-efficiency-based algorithm is much faster (though still exponential time), and is significantly cleaner to implement. Moreover, it becomes far simpler to detect and discard trivial 3-sphere summands, since 3-sphere recognition is significantly less demanding for 0-efficient triangulations than for general inputs [15]. Historically this algorithm was the turning point at which prime decomposition first became practical, and in 2004 it became the foundation for the first real software implementation [2].

For non-orientable triangulations, the state of the art remains the original Jaco-Tollefson algorithm, which has still never been implemented due to the speed and intricacy reasons outlined above. Here we now use the results of Section 2 to develop a fast and simple prime decomposition algorithm for non-orientable triangulations, based on 0-efficiency and the Jaco-Rubinstein crushing process.

In this setting there is a major complication: for non-orientable manifolds, the Jaco-Rubinstein crushing process might leave us with an *invalid triangulation*, where some edge is identified with itself in reverse. As noted in the detailed proof of Lemma 3, this can only occur if the triangulation contains an embedded two-sided projective plane.

We employ a "permissive" strategy for dealing with this complication: we run the algorithm regardless of whether there might be problems, and after it finishes we test whether anything went wrong by looking for invalid edges (an easy test to perform). This permissive approach has two benefits:

- There are *no onerous preconditions* to test before we run the algorithm.[6] Instead we can start the algorithm immediately, oblivious to whether there is an embedded two-sided projective plane or not.

- The algorithm *might still succeed* even if there is an embedded two-sided projective plane: if we "get lucky" and do not create an invalid edge, we still guarantee correctness. If we are unlucky and we do create an invalid edge during some atomic move, we prove that this can be detected after the fact, once the crushing process is complete.

We begin this section with Lemma 5, a non-orientable extension to Corollary 4 that uses the crushing lemma to identify the possible consequences of crushing in the presence of two-sided projective planes. We follow this with the full prime decomposition algorithm, as detailed in Algorithm 6.

LEMMA 5. *Let $\mathcal{T}$ be a generalised triangulation of any closed compact 3-manifold $\mathcal{M}$, and let S be a normal sphere in $\mathcal{T}$. Then, if we destructively crush S using the Jaco-Rubinstein procedure, one of the following things happens:*

1. *we obtain an invalid triangulation, in which some edge is identified with itself in reverse;*

2. *we obtain a valid triangulation $\mathcal{T}_{\mathrm{JR}}$ whose underlying 3-manifold $\mathcal{M}_{\mathrm{JR}}$ is obtained from $\mathcal{M}$ by zero or more of the following operations:*

    - *undoing connected sums, i.e., replacing some intermediate manifold $\mathcal{M}'$ with the disjoint union $\mathcal{M}'_1 \cup \mathcal{M}'_2$, where $\mathcal{M}' = \mathcal{M}'_1 \# \mathcal{M}'_2$;*

    - *deleting 3-sphere, $\mathbb{R}P^3$, $L_{3,1}$, $S^2 \times S^1$ or twisted $S^2 \tilde{\times} S^1$ components.*

Moreover, (1) can only occur if $\mathcal{M}$ contains an embedded two-sided projective plane.

PROOF. Once more we give a brief sketch here, and give the complete details in the full version of this paper.

If $\mathcal{M}$ does not contain an embedded two-sided projective plane then this is a special case of Corollary 4, which immediately yields outcome (2) above.

If there are embedded two-sided projective planes, then we once again study the possible consequences of each atomic move from Lemma 2. Here we discover that flattening a bigon face might create a pair of invalid edges.

Our task then is to show that, if an atomic move ever creates invalid edges, then any subsequent atomic moves *preserve* the existence of invalid edges. For this we use a parity argument. Define an *odd or even vertex* of the triangulation to be one that is incident with an odd or even number of invalid edges respectively. We show that the first invalid edges we create introduce odd vertices, and although subsequent atomic moves might remove invalid edges they will never reduce the number of odd vertices. Therefore there will be an odd vertex after all operations are complete (and hence an invalid edge), and so we obtain outcome (1) above. □

We can now package the results of Lemma 5 into a general algorithm for computing the prime decomposition of a triangulated 3-manifold, either orientable or non-orientable. The structure of the algorithm follows the modern "ready to implement" framework presented in [4] for the orientable case. The process can be further improved by simplifying triangulations at key stages of the algorithm; we omit this here, but details can be found in [4].

ALGORITHM 6 (PRIME DECOMPOSITION). *Given an input triangulation $\mathcal{T}$ of any closed connected 3-manifold $\mathcal{M}$, the following algorithm will either decompose $\mathcal{M}$ into a connected sum of prime manifolds, or else prove that $\mathcal{M}$ contains an embedded two-sided projective plane.*

1. *Compute the first homology of $\mathcal{T}$, and let r, t_2 and t_3 denote the $\mathbb{Z}$ rank, $\mathbb{Z}_2$ rank and $\mathbb{Z}_3$ rank respectively.*

2. *Create an input list $\mathcal{L}$ of triangulations to process, initially containing just $\mathcal{T}$, and an output list $\mathcal{O}$ of prime summands, initially empty.*
   While $\mathcal{L}$ is non-empty:

    - *Let $\mathcal{N}$ be the next triangulation in the list $\mathcal{L}$. Remove $\mathcal{N}$ from $\mathcal{L}$, and test whether $\mathcal{N}$ has a non-trivial normal sphere F.*
        - *If there is such a normal sphere, then perform the Jaco-Rubinstein crushing procedure on F.*
            - *If the resulting triangulation has an invalid edge, then terminate with the statement that the input manifold contains an embedded two-sided projective plane.*
            - *If the resulting triangulation has no invalid edges, then add each connected component of the resulting triangulation back into the list $\mathcal{L}$.*
        - *If there is no such normal sphere, then append $\mathcal{N}$ to the output list $\mathcal{O}$.*

[6] The absence of two-sided projective planes is an "onerous" precondition, in the sense that there is no algorithm known at present that can test this in polynomial time.

3. *Compute the first homology of each triangulation in the output list $\mathcal{O}$, and let r', t_2' and t_3' denote the sums of the $\mathbb{Z}$ ranks, $\mathbb{Z}_2$ ranks and $\mathbb{Z}_3$ ranks respectively.*

4. *Append $(t_2 - t_2')$ copies of $\mathbb{R}P^3$ and $(t_3 - t_3')$ copies of $L_{3,1}$ to $\mathcal{O}$. If the input triangulation was orientable, append $(r - r')$ copies of $S^2 \times S^1$ to $\mathcal{O}$, and otherwise append $(r - r')$ copies of the twisted product $S^2 \tilde{\times} S^1$ to $\mathcal{O}$.*

If we did not terminate earlier due to an invalid edge, then the final output list $\mathcal{O}$ will contain a collection of triangulated prime manifolds $\mathcal{O}_1, \ldots, \mathcal{O}_k$ for which the original manifold $\mathcal{M}$ can be expressed as the connected sum $\mathcal{O}_1 \# \mathcal{O}_2 \# \ldots \# \mathcal{O}_k$.

The correctness of this algorithm follows immediately from Lemma 5 (the changes in $\mathbb{Z}$, $\mathbb{Z}_2$ and $\mathbb{Z}_3$ ranks indicate the number of $S^2 \times S^1$ or $S^2 \tilde{\times} S^1$, $\mathbb{R}P^3$, and $L_{3,1}$ summands that were lost respectively). If the input triangulation contains n tetrahedra, it is clear that we terminate after crushing at most n normal spheres, since each crushing operation strictly reduces the total number of tetrahedra in the input list $\mathcal{L}$. Note that some of the output manifolds $\mathcal{O}_k$ might be trivial (i.e., redundant 3-sphere summands); however, such trivial summands are easy to detect, as outlined below.

As presented above, this algorithm gives the *summands* in the connected sum decomposition, but these may not uniquely define the original manifold (since there can be orientation-related decisions to make when performing the connected sum operation). This can be resolved by tracking orientations explicitly through the crushing process, a straightforward but slightly messy enhancement to the algorithm that we do not describe in detail here.

We finish with some implementation notes:

- There are well-known polynomial-time procedures for computing homology in step (1), based on Smith normal form; see [9, 10] for examples.

- In step (2) we must locate a non-trivial normal sphere, if one exists. Traditionally, one does this by enumerating all *quadrilateral vertex normal surfaces*; see [4] for details on what this means and why it works. A newer (and experimentally much faster) alternative is to make a targeted search for a normal sphere using branch-and-bound techniques from combinatorial optimisation; see [7] for details.

- It is easy to eliminate trivial 3-sphere summands from the output list. If an output triangulation $\mathcal{O}_i$ has non-trivial homology then it is a non-trivial summand; otherwise $\mathcal{O}_i$ must be 0-efficient, whereupon Jaco and Rubinstein show that $\mathcal{O}_i$ is trivial if and only if (i) it has more than one vertex, or (ii) it contains an embedded *almost normal sphere*. See [15] for details on almost normal spheres, and see [3, 7] for fast algorithms for detecting them.

4. ACKNOWLEDGEMENTS

The author is grateful to the anonymous referees for their detailed comments. This work is supported by the Australian Research Council under the Discovery Projects funding scheme (project DP1094516).

5. REFERENCES

[1] M. V. Andreeva, I. A. Dynnikov, and K. Polthier. A mathematical webservice for recognizing the unknot. In *Mathematical Software: Proceedings of the First International Congress of Mathematical Software*, pages 201–207. World Scientific, 2002.

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

[3] B. A. Burton. Quadrilateral-octagon coordinates for almost normal surfaces. *Experiment. Math.*, 19(3):285–315, 2010.

[4] B. A. Burton. Computational topology with Regina: Algorithms, heuristics and implementations. In C. D. Hodgson, W. H. Jaco, M. G. Scharlemann, and S. Tillmann, editors, *Geometry and Topology Down Under*, number 597 in Contemporary Mathematics. Amer. Math. Soc., Providence, RI, 2013.

[5] 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.

[6] B. A. Burton, A. Coward, and S. Tillmann. Computing closed essential surfaces in knot complements. In *SCG '13: Proceedings of the 29th Annual Symposium on Computational Geometry*, pages 405–414. ACM, 2013.

[7] B. A. Burton and M. Ozlen. A fast branching algorithm for unknot recognition with experimental polynomial-time behaviour. Preprint, `arXiv:1211.1079`, Nov. 2012.

[8] 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.

[9] B. R. Donald and D. R. Chang. On the complexity of computing the homology type of a triangulation. In *32nd Annual Symposium on Foundations of Computer Science (San Juan, PR, 1991)*, pages 650–661. IEEE Comput. Soc. Press, Los Alamitos, CA, 1991.

[10] J.-G. Dumas, F. Heckenbach, D. Saunders, and V. Welker. Computing simplicial homology based on efficient Smith normal form algorithms. In *Algebra, Geometry, and Software Systems*, pages 177–206. Springer, Berlin, 2003.

[11] J. Fowler. *Finding 0-efficient triangulations of 3-manifolds*. Senior honors thesis, Harvard University, 2003.

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

[13] W. Haken. Über das Homöomorphieproblem der 3-Mannigfaltigkeiten. I. *Math. Z.*, 80:89–120, 1962.

[14] W. Jaco. The homeomorphism problem: Classification of 3-manifolds. Lecture notes, Available from `http://www.math.okstate.edu/~jaco/pekinglectures.htm`, 2005.

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

[16] W. Jaco and E. Sedgwick. Decision problems in the space of Dehn fillings. *Topology*, 42(4):845–906, 2003.

[17] W. Jaco and J. L. Tollefson. Algorithms for the

complete decomposition of a closed 3-manifold. *Illinois J. Math.*, 39(3):358–406, 1995.

[18] T. Li. An algorithm to determine the Heegaard genus of a 3-manifold. *Geom. Topol.*, 15(2):1029–1106, 2011.

[19] S. Matveev. *Algorithmic Topology and Classification of 3-Manifolds*. Number 9 in Algorithms and Computation in Mathematics. Springer, Berlin, 2003.

[20] 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.

[21] J. H. Rubinstein. An algorithm to recognise small Seifert fiber spaces. *Turkish J. Math.*, 28(1):75–87, 2004.

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