On the Complexity of Immersed Normal Surfaces

Benjamin A. Burton[*]   Éric Colin de Verdière[†]   Arnaud de Mesmay[‡]

December 17, 2014

Abstract

Normal surface theory, a tool to represent surfaces in a triangulated 3-manifold combinatorially, is ubiquitous in computational 3-manifold theory. In this paper, we investigate a relaxed notion of normal surfaces where we remove the quadrilateral conditions. This yields normal surfaces that are no longer embedded. We prove that it is NP-hard to decide whether such a surface is immersed. Our proof uses a reduction from Boolean constraint satisfaction problems where every variable appears in at most two clauses, using a classification theorem of Feder. We also investigate variants, and provide a polynomial-time algorithm to test for a local version of this problem.

1   Introduction

The field of computational topology aims at providing computational and efficient tools to deal with topological problems. In this theory, the dimension of the problems considered has a very direct impact on the complexity of the algorithms designed to solve them. Fundamental problems tend to have polynomial-time solutions for surfaces [6], while in dimensions four and larger, things easily become undecidable [23, Chapter 9]. In the intermediate case, most of the problems encountered in 3-dimensional topology are decidable but typically solved with (at least) exponential-time algorithms. Probably the most famous of these is unknot recognition, whose complexity is in NP [13], and in co-NP assuming the Generalized Riemann Hypothesis [15], but for which no polynomial-time algorithm nor hardness proof is known.

Normal surfaces.   A standard way to study 3-manifolds is to investigate which surfaces can be embedded in them. Normal surfaces, used in a wealth of algorithms, are perhaps the most ubiquitous tool for this purpose. First brought to the algorithmic light by Haken [12], normal surfaces provide a compact and structured way to analyze and enumerate the most interesting surfaces embedded in a 3-manifold. Starting with a triangulation T of a 3-manifold M with t tetrahedra, a *normal surface* is a (possibly disconnected) surface in M whose intersection with each tetrahedron is a disjoint union of disks of seven simple possible types (see Figure 1 and Section 2.1 below). This allows a normal surface to be entirely described by a vector in $\mathbb{Z}_+^{7t}$, its *normal coordinates*. Many interesting surfaces, such as for example a Seifert disk of an unknotted closed curve, are witnessed by a normal surface having coordinates at most exponential in t; see Hass, Lagarias, and Pippenger [13]. This is the starting point of many

algorithms based on the enumeration of normal surfaces, which naturally have an exponential complexity.

In addition to providing a succinct representation of embedded surfaces, normal surfaces also possess an additional algebraic structure. Indeed, the natural addition and scalar multiplication of vectors translate to operations on normal surfaces, and the space of normal surfaces in $\mathbb{R}^{7t}$ is characterized by a set of equations: the *matching equations* and the *quadrilateral conditions*. The former are linear equations specifying the way to glue normal surfaces locally, while the latter are non-linear and ensure that the resulting surface is embedded. Spaces defined by linear constraints can be studied by the means of linear programming [17], which provides a very powerful framework to deal with decision and optimization problems. This motivates the study of a notion of relaxed normal surfaces, where we remove the quadrilateral conditions to obtain a simpler, polyhedral structure on the space of normal surfaces.

Our results. As we shall see later, removing the quadrilateral conditions amounts to removing the embeddedness of normal surfaces. Therefore, it amounts to dealing with *singular normal surfaces*. Among these, the *immersed normal surfaces* are well behaved, in the sense that while they can self intersect, they are still 2-manifolds locally. Moreover, their Euler characteristic depends linearly on their normal coordinates—this fact is crucial in algorithms that work with embedded normal surfaces [13], but does not hold for all singular normal surfaces. By coupling singular normal surface theory with an algorithm that efficiently separates immersed normal surfaces from the others, we would have powerful tools at our disposal: this could lead to efficient algorithms to find immersed low genus surfaces in 3-manifolds. Furthermore, through classical topological results like Dehn's lemma or the loop theorem [14] we would obtain embedded surfaces, which are the key behind the unknot problem and many others.

In this article, we show some inherent limitations of this method by proving in Theorem 6 that it is NP-hard to detect whether a given set of normal coordinates can represent an immersed normal surface. In other words, there exists no polynomial-time algorithm for this problem unless P=NP.

Immersed normal surfaces have been studied from a mathematical point of view by Letscher [16] and, in the particular case of the figure-eight knot complement, from a computational perspective by Aitchison, Matsumoto, and Rubinstein [2], Rannard [20], and Matsumoto and Rannard [18]. In the latter papers, the authors devise and implement an algorithm to decide whether a given set of normal coordinates can represent an immersed normal surface, and find an such an immersed surface if there is one. While the complexity of this algorithm is not explicitly computed, it is at least doubly exponential in the input size. Our main result shows that the problem is inherently hard and that no polynomial-time solution is to be expected.

The complexity reduction used in the proof of this theorem works by reducing the problem to a satisfiability problem, which at first sight appears to follow a standard technique. However, the flavor of this reduction is that in this problem, it turns out to be very hard to obtain more than two copies of every variable. Our proof thus relies on relatively intricate classification theorems on the complexity of Boolean constraint satisfaction problems where every variable occurs at most twice [8, 9]. This approach is thus, to some extent, original, and might prove useful to obtain other hardness proofs that could be hard to achieve by other traditional means.

Hardness results are scarce in 3-dimensional computational topology, and to our knowledge all the other difficulty results are deduced from the Agol, Hass, and Thurston construction [1], except for the recent hardness results on computing taut angle structures [4] and optimal Morse matchings [5]. Our result displays a different intractability aspect of this theory. In contrast to the aforementioned result by Agol, Hass, and Thurston [1], we also prove that this problem is NP-hard even when the input triangulation is a sub-manifold of $\mathbb{R}^3$, which is for example the case for the very important class of knot complements.

On the upper bound side, it is a natural question to wonder whether this problem can be

solved in polynomial time when the size of the triangulation is fixed. As a partial evidence for this, the aforementioned work of Matsumoto and Rannard [18] suggests that the problem may be solved in polynomial time in the specific case of the figure-eight knot complement. Although we make no progress on this question in full generality, we show that if the triangulation just consists of tetrahedra all sharing a single edge, the problem has a polynomial-time solution as it can solved by computing a maximum flow. Another view on this algorithm is that it can certify *local* immersibility, where the locality means that it can only check whether there is an obstruction to being immersed around every edge, but it is not global since ensuring immersibility as some point might force a branch point at some other point for the triangulation.

This paper is organized as follows. We start by introducing the main concepts of 3-manifold topology, normal surface theory, and Boolean constraint satisfaction in Section 2. In Section 3, we describe our reduction and prove the main theorem. Section 4 explores some variants of the immersibility problem that remain NP-hard. Finally, in Section 5, we provide an algorithm to test immersibility in the restricted case of tetrahedra glued around a single edge.

2 Preliminaries

2.1 Background on normal surface theory

For completeness, we first give a (rather standard) background on normal surface theory; see e.g., Hass, Lagarias, and Pippenger [13] or Gordon [11] for more details. A *3-manifold with boundary* is a compact topological space such that every point is locally homeomorphic to $\mathbb{R}^3$ or to the closed half space $\{x, y, z \mid x, y, z \in \mathbb{R}, z \geq 0\}$. We will describe 3-manifolds using *triangulations*: A triangulation T is a topological space obtained from a disjoint set of t tetrahedra $T = (T_1, \ldots, T_t)$ by (combinatorially) gluing some pairs of two-dimensional faces of these tetrahedra; a gluing between two faces is specified by a bijection from the vertex set of the first face to the vertex set of the second face. As a result of these gluing, edges and vertices of tetrahedra are also identified; it is also allowed to glue two zero-, one-, or two-dimensional faces of the same tetrahedron. A *face* of a triangulation T is a two-dimensional simplex, incident to one or two tetrahedra in T.

The following lemma is standard [19]; we include it for further reference:

Lemma 1. *A triangulation is a 3-manifold (possibly with boundary) if and only if:*

1. each vertex has a neighborhood homeomorphic to $\mathbb{R}^3$ or to the closed half-space;

2. after the gluings, no edge is identified to itself in the reverse orientation.

(These two conditions are obviously necessary, and turn out to be sufficient to ensure that every point of the triangulation has a manifold neighborhood.)

Henceforth, T denotes a triangulation of a 3-manifold M. A *normal isotopy* is an ambient isotopy of M that is fixed on the 2-skeleton of T. A *normal surface* in T is a properly embedded surface in T that meets each tetrahedron in a (possibly empty) disjoint collection of *normal disks*, each of which being either a *triangle* (separating one vertex of the tetrahedron from the other three) or a *quadrilateral* (separating two vertices from the other two). In each tetrahedron, there are 4 possible types of triangles and 3 possible types of quadrilaterals, pictured in Figure 1. The intersection of an embedded normal surface with a face of the triangulation gives rise to a *normal arc*. There are 3 possible types of normal arcs within each face: the type of a normal arc is defined according to which vertex of the face it separates from the other two.

Each embedded normal surface has associated *normal coordinates*: a vector in $(\mathbb{Z}_+)^{7t}$, where t is the number of tetrahedra in T, listing the number of triangles and quadrilaterals of each type in each tetrahedron. The normal coordinates provide a very compact and elegant description of that surface, and satisfy two types of conditions:
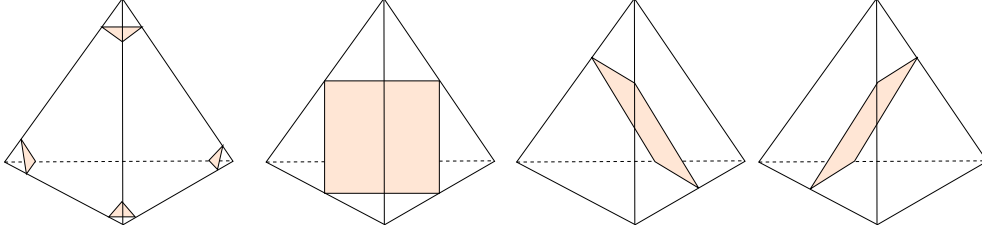
Figure 1: The seven types of normal disks within a given tetrahedron: Four triangles and three quadrilaterals.

- The first type of conditions is the *matching equations*. Consider a normal arc type in a given non-boundary face f of T. This normal arc type corresponds to exactly one triangle normal coordinate, $v_{t,1}$, and one quadrilateral normal coordinate, $v_{q,1}$, in a tetrahedron incident with f. Similarly, let $v_{t,2}$ and $v_{q,2}$ be the triangle and quadrilateral normal coordinates corresponding to that arc type in the opposite tetrahedron. The matching equation for that arc type is, by definition, $v_{t,1} + v_{q,1} = v_{t,2} + v_{q,2}$.

- The second type of conditions, the *quadrilateral conditions*, stipulates that, within any tetrahedron, at most one of the three quadrilateral coordinates must be non-zero. Indeed, two quadrilaterals of different types within the same tetrahedron must cross, and therefore this condition is needed to ensure that the surface does not self-intersect.

Conversely, if T is a triangulation of size t and $v \in (\mathbb{Z}_+)^{7t}$. Then v corresponds to an embedded normal surface if and only if the matching equations and the quadrilateral conditions are fulfilled. The reconstruction process can be described as follows:

- In each tetrahedron, by the quadrilateral conditions, there is at most one non-zero quadrilateral coordinate. One places as many parallel copies of this quadrilateral as needed in the tetrahedron, and then place each triangle close to the vertex of the tetrahedron that it separates from the other three. It is straightforward to do so without having any intersection between triangles and quadrilaterals.

- One glues the faces on the triangulation together, and in the process, one needs to glue normal arcs, i.e., triangles or quadrilaterals on the one side to triangles and quadrilaterals on the other side. By the matching equations, the numbers fit, and the gluing is imposed by the order in which the normal disks are placed in the tetrahedra.

Therefore, an embedded normal surface is represented up to a normal isotopy by a vector in $(\mathbb{Z}_+)^{7t}$ satisfying the matching equations and the quadrilateral conditions. Moreover, given a triangulation and normal coordinates, checking that the matching equations or the quadrilateral conditions hold can trivially be done in linear time.

From this construction, one sees moreover that every set of normal coordinates corresponds to a unique normal surface, up to a normal isotopy.

2.2 Singular and immersed normal surfaces

Consider a vector $v \in (\mathbb{Z}_+)^{7t}$ of normal coordinates satisfying the matching equations, but not necessarily the quadrilateral conditions. For each $i \in \{1, \ldots, 7t\}$, build v_i normal disks of the corresponding type in the corresponding tetrahedron, in general position, in a way that, on each non-boundary face f of T, the images of the normal arcs arising from both sides of f agree. The matching equations imply that such a construction is always possible. Note that, with this gluing, we do not forbid intersections between normal disks in a single tetrahedron, and that

such intersections are actually necessary if two different quadrilateral coordinates within the same tetrahedron are non-zero.

More precisely, consider a given normal arc type in a given non-boundary face f of T, corresponding (as above) to two normal coordinates $v_{t,1}$ and $v_{q,1}$ in a tetrahedron incident to f, and also to two normal coordinates $v_{t,2}$ and $v_{q,2}$ in the adjacent tetrahedron. Recall that the matching equations imply that $v_{t,1} + v_{q,1}$ and $v_{t,2} + v_{q,2}$ are equal. The data of a bijection between these $v_{t,1} + v_{q,1}$ normal disks in the first tetrahedron with these $v_{t,2} + v_{q,2}$ normal disks in the second tetrahedron is called the *local gluing* of that arc type. The aggregated information of all the local gluings is called the *global gluing*.
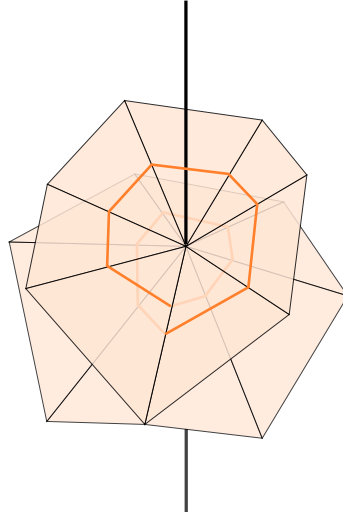

Figure 2: A branch point of a singular normal surface.

The union of these normal disks glued according to such rules is the image of a surface under a continuous map, since abstractly gluing triangles and quadrilaterals by pairwise identifications of edges always results in a surface (whose actual geometric realization in T may self-cross). This is called a *singular normal surface*. The continuous map may either be locally one-to-one, in which case it is called an *immersed normal surface*, or have *branch points*, as pictured in Figure 2. Since normal disks are embedded within each tetrahedron, any branch point of a singular surface is necessarily on an edge of the triangulation; it corresponds to a small closed curve on the surface "winding more than once" around the edge of the triangulation.

2.3   The immersibility problem

The main problem studied in this paper is the following. Consider now some normal coordinates satisfying the matching equations, but not necessarily the quadrilateral conditions. As described above, there are many ways to glue the corresponding normal disks to obtain a singular normal surface. Then, depending on the choice of the gluings, some of the resulting surfaces may be immersed while some other may have branch points. If there exists a global gluing whose corresponding singular surface is immersed, we say that the normal coordinates are *immersible*. In this chapter, we study the computational complexity of the following problem:

Problem 2 (IMMERSIBILITY).
Input: *A triangulation T and normal coordinates N.*
Output: *Are the normal coordinates N immersible?*

Two difficulties lie at the heart of this problem: Not only do we need to guess a "good" gluing, but this gluing may have an exponential complexity in the input, since the normal coordinates are naturally compressed by the bit representation. Therefore, the naive algorithm
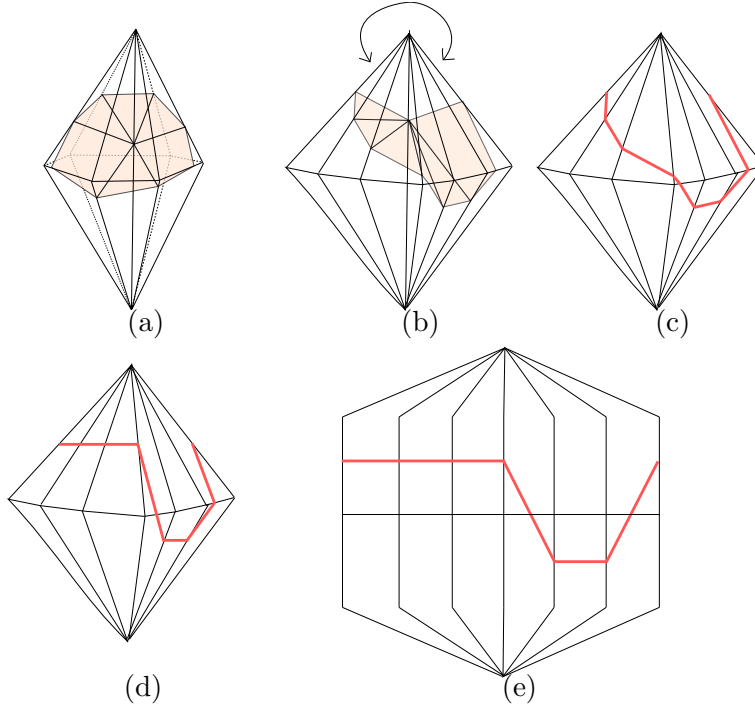
Figure 3: A *block*, namely, a set of tetrahedra sharing a common non-boundary edge, depicted with a part of a singular normal surface inside it, and its abstract representation.

(implemented by Matsumoto and Rannard [18]) is doubly exponential, which can be seen just at the level of local gluings: If there are k arcs of a given arc type, there are $k!$ possible local gluings, while the normal coordinates involved are $O(k)$ and can thus be encoded in $O(\log k)$ bits.

2.4   Representation of singular normal surfaces

To describe the singular normal surfaces more accurately, we now introduce a schematic representation of singular normal surfaces.

A *block* is a family of tetrahedra that all have one edge e in common; see Figure 3(a); we always assume that e is not on the boundary of T. In what follows, we consider the normal disks that intersect e. Since we want to picture cleanly what happens on the back of this block, we will unfold it as in Figure 3(b), with the implicit convention that the rightmost face is glued to the leftmost face. Although normal disks can be drawn inside this block, the pictures easily become congested when there are several of them. Instead, we will forget the edge e in the representation and represent the normal disks intersecting e by their normal arcs, i.e., by their intersection with the front faces (Figure 3(c)). These normal arcs are glued together and form possibly self-intersecting closed curves, called *block curves*. Abstracting a bit more, horizontal lines will represent triangles, while diagonal ones will stand for quadrilaterals (Figure 3(d)). Finally, to make these pictures even more readable, we will draw the edges between the tetrahedra vertically, only linking them at the extreme top and bottom parts of the figures (Figure 3(e)).

We will use the following convention in the figures (see Figure 4): Whenever we want to represent normal coordinates, without a specific gluing, the normal arcs are drawn so that they connect the midpoints of the corresponding edges of the triangulation. Whenever we want to represent normal coordinates with a particular gluing, we perturb these normal arcs to emphasize the crossings.

The singular normal surface has a branch point at e if and only if some block curve "winds more than once" around e or, equivalently, self-intersects, like in Figure 5. With this in mind,
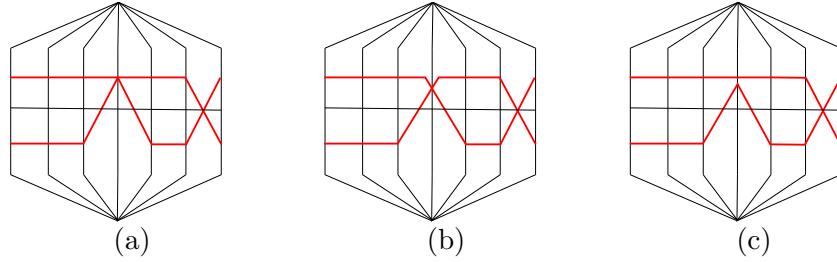
Figure 4: Normal coordinates drawn (a) without specifying gluings, (b) with a specified gluing, (c) with the opposite gluing from (b).

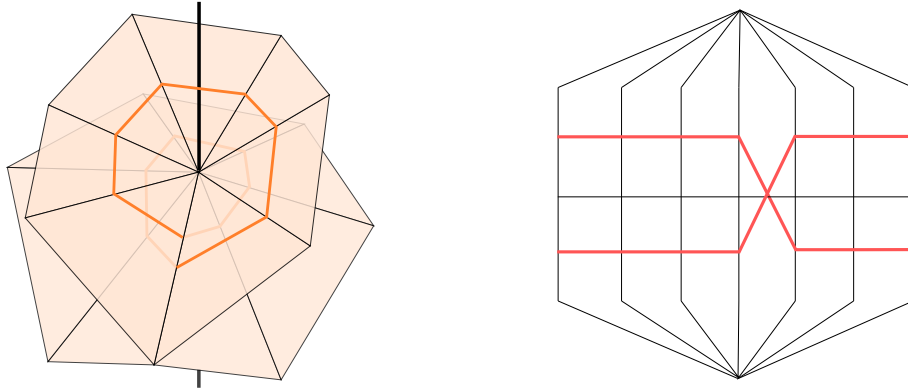it is easy to see that a branch point can only occur on a *non-boundary* edge of T.

Figure 5: A branch point (also depicted in Figure 2) and its representation by a block curve winding twice around an edge.

Therefore, given the data of normal coordinates satisfying the matching equations, together with a global gluing, we can easily determine whether the corresponding singular normal surface is immersed or not, in time linear in the size of the input (namely, sum of all the normal coordinates, of the complexity of the triangulation T and the complexity of the global gluing). However, our goal is to prove that, given only the normal coordinates, it is NP-hard to decide whether some global gluing leads to an immersed surface.

2.5 Complexity classes

We now very briefly introduce some standard notions of complexity theory; see, e.g., Arora and Barak [3] for more details. P denotes the class of yes/no problems solvable in polynomial time. NP denotes the class of yes/no problems whose "yes" instances admit a polynomial-sized *certificate*: some additional data such that, when given the instance and the certificate, it can indeed be verified in polynomial time that the instance is a "yes" instance. Obviously, P is included in NP, and probably the most outstanding open question in computer science is to determine whether P is equal or different from NP; it is widely believed that they are different.

A problem A *reduces* to a problem B (roughly) if, when given an oracle to solve problem B, one can solve problem A in polynomial time. A problem A is *NP-hard* if it is "at least as hard" as any problem in NP, or more precisely if every problem in NP reduces to it. If P and NP are different, then no NP-hard problem can be solved in polynomial time. Finally, a problem is *NP-complete* if it is both NP-hard and in NP.

2.6 Boolean constraint satisfaction problems

In this section, we recall a few basic results about Boolean constraint satisfaction problems; our presentation is inspired from Dalmau and Ford [8]. For a more detailed account of this tremendous body of research, we refer to the book of Creignou, Khanna, and Sudan [7].

We start by introducing the *generalized satisfiability problem* SAT(R), which is a variant of the usual SAT problem. An r-ary relation R is any nonempty subset of $\{0,1\}^r$. A CNF(R)-formula is a finite conjunction of clauses $C_1 \wedge \ldots \wedge C_n$ such that each clause, C_i, is an *atomic formula* of the form $R(v_1, \ldots, v_r)$ where $v_1 \ldots v_r$ are Boolean variables. An atomic formula $R(v_1, \ldots v_r)$ is *satisfied* by a variable assignment $f : V \to \{0,1\}$ if and only if $(f(v_1) \ldots f(v_r)) \in R$, and a CNF($R$) formula is *satisfiable* if and only if there exists an assignment satisfying all its clauses simultaneously. Each relation R gives rise to the *generalized satisfiability problem* SAT(R): given a CNF(R)-formula, is it satisfiable?

It is sometimes convenient to assume that constants can appear in CNF(R)-formulas: Each clause is an atomic formula of the form $R(v_1, \ldots, v_r)$ where each v_i is a Boolean variable or a constant (0 or 1). We call any formula obtained this way a CNF$_C$(R)-formula. Similarly, the problem *generalized satisfiability problem with constants*, SAT$_C$(R), is defined with CNF$_C$(R)-formulas.

The computational complexity of the generalized satisfiability problem with constants has been completely classified by Schaefer in a celebrated paper [21]. We introduce the following definitions in order to state this classification. Here, the symbols $\wedge$, $\vee$, and $\oplus$: $\{0,1\}^r \times \{0,1\}^r \to \{0,1\}$ denote the usual logical operations AND, OR and XOR, applied bit-wise.

A relation R is

- *Horn* if $x, y \in R \to x \wedge y \in R$,

- *dual-Horn* if $x, y \in R \to x \vee y \in R$,

- *bijunctive* if $x, y, z \in R \to (x \wedge y) \vee (x \wedge z) \vee (y \wedge z) \in R$,

- *affine* if $x, y, z \in R \to x \oplus y \oplus z \in R$.

A relation R is *Schaefer* if it is Horn, dual-Horn, bijunctive, or affine.

Theorem 3 ([21]). *Let R be a relation. If R is Schaefer, then $SAT_C(R)$ is in P, otherwise it is NP-hard.*

For our reduction, we will restrict ourselves to constraint satisfaction problems where the number of occurrences of every variable is at most 2. We denote by SAT($2, R$) the instances of SAT(R) in which every variable occurs at most twice. Similarly, we denote by SAT$_C$($2, R$) the instances of SAT$_C$(R) in which every variable occurs at most twice. The following definition is key to the classification of these problems.

Let $R \subseteq \{0,1\}^r$ be a relation. Let $x, y, x' \in \{0,1\}^r$, then x' is a *step* from x to y if $d(x, x') = 1$ and $d(x, x') + d(x', y) = d(x, y)$, where d is the Hamming distance. R is a Δ-*matroid (relation)* if it satisfies the following *two-step axiom*:

> For all $x, y \in R$ and for all x' a step from x to y, either $x' \in R$ or there exists $x'' \in R$ which is a step from x' to y.

We now come to the classification theorem for SAT$_C$($2, R$):

Theorem 4 (Feder [9]). *Let R be a relation that is not a Δ-matroid relation. Then $SAT_C(2, R)$ is polynomially equivalent to $SAT_C(R)$.*

Theorems 3 and 4 immediately imply the complexity result that we will use:

Corollary 5. *Let R be a relation that is not Schaefer (that is, not Horn, dual Horn, bijunctive, or affine) and not a Δ-matroid. Then $SAT_C(2, R)$ is NP-hard.*

3   NP-hardness of detecting immersibility

In this section, we prove the following theorem.

Theorem 6. *The problem* IMMERSIBILITY *is NP-hard.*

The proof of Theorem 6 will proceed by a reduction of $\text{SAT}_C(2, R)$ to the problem IMMERSIBILITY for a relation R that is neither Schaefer nor a Δ-matroid, which implies by Corollary 5 that $\text{SAT}_C(2, R)$ and hence IMMERSIBILITY are NP-hard.

3.1   Gadgets

We now show how to reduce $\text{SAT}_C(2, R)$ to the problem IMMERSIBILITY. To this end, we use a 6-ary relation R which we will describe later. We start with a formula Φ that is, by definition, a conjunction of clauses of the form $R(x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4}, x_{i_5}, x_{i_6})$, where x_i is either a variable or a constant, and every variable appears at most twice in Φ.

The gadgets that we use for our reduction are of three types. Each clause is represented by a *clause gadget*, a block of six tetrahedra glued together around an edge. For each variable occurring exactly twice in Φ, we connect these two occurrences in the clauses using *tubes*, which are also blocks of six tetrahedra. Finally, the *constant gadgets* are used to represent the constants 0 or 1 appearing in the clauses. The idea for the proof is that a clause is satisfiable if and only if the normal coordinates in the clause gadget are immersible; the tubes then enforce consistency between the clauses. Therefore, the whole formula will be satisfiable if and only if the associated normal coordinates are immersible.

The clause gadget.   Consider the gadget G pictured in Figure 6. It consists of six tetrahedra that all have an edge in common, and contain each three normal disks: two triangles and one quadrilateral. For every clause in Φ, we create a copy of the clause gadget G; these copies will be connected using tubes, described below.
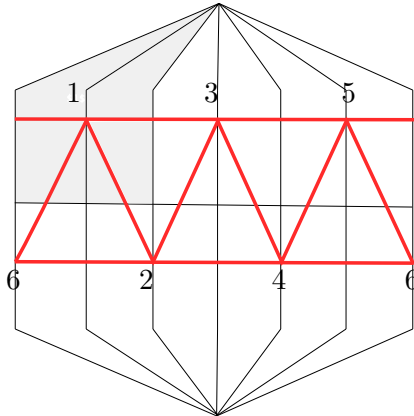

Figure 6: The clause gadget. The couple of faces corresponding to the variable x_1 is shaded.

The rationale behind this gadget is the following. At the interface of two adjacent tetrahedra, exactly two gluings can be done (see Figure 7). This choice of gluing can be described by a variable $x_i \in \{0, 1\}$, where 0 corresponds to the gluing a. and 1 to the gluing b. Equivalently, a value of 1 corresponds to the fact that the two block curves at the specified position cross. Therefore, each variable in a clause has a pair of associated faces on the boundary of the clause gadget; for example, the two shaded triangles in Figure 6 are the pair of faces associated to variable x_1.

This way, a global gluing of the singular normal surface in G is described by an element $x \in \{0, 1\}^6$. The order on the six variables is pictured in Figure 6.
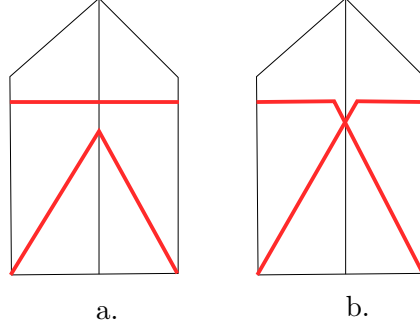
Figure 7: a. Local gluing corresponding to $x_i = 0$. b. Local gluing corresponding to $x_i = 1$.

We define the following relation R on 6 variables:

$$R \;=\; \{ \quad (0,0,0,0,0,0); (0,0,0,1,0,1); (0,0,1,0,1,0); (0,1,0,0,0,1); (0,1,0,1,0,0);$$
$$(0,1,1,0,1,1); (1,0,0,0,1,0); (1,0,1,0,0,0); (1,0,1,1,0,1); (1,1,0,1,1,0);$$
$$(1,1,1,1,1,1) \quad \}$$

This allows us to get to the following lemma.

Lemma 7. *The singular normal surface in the gadget G specified by the gluing $x \in \{0,1\}^6$ is immersed if and only if $x \in R$.*

Proof of Lemma 7. The proof is done by exhaustive checking, i.e., checking for every possible 6-tuple whether there is a branch point around the central edge or not. As an example, Figure 8 pictures the singular normal surfaces obtained with the global gluings $(1,0,1,1,0,1)$ and $(1,0,1,1,1,1)$, yielding in one case an immersed normal surface and in the other a singular normal surface with a branch point. $\qquad\square$
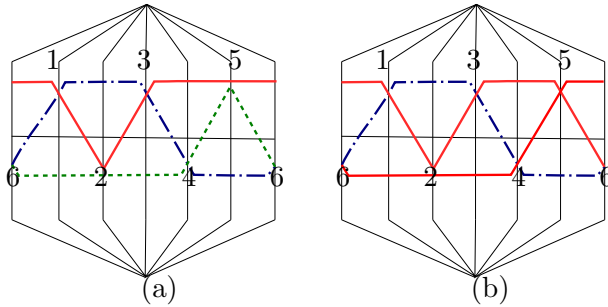

Figure 8: (a) Immersed normal surface corresponding to coordinates (1,0,1,1,0,1), yielding three block curves. (b) Singular normal surface corresponding to coordinates (1,0,1,1,1,1), yielding two block curves. The one winding twice around the central edge witnesses a branch point.

Moreover:

Proposition 8. $SAT_C(2, R)$ *is NP-hard.*

Proof. By Corollary 5, it suffices to prove that the relation R is not Schaefer, that is, neither (i) Horn, (ii) dual Horn, (iii) bijunctive, nor (iv) affine; and that, furthermore, it is (v) not a Δ-matroid. The proofs are somewhat tedious but straightforward.

(i) $(1,0,1,0,0,0)$ and $(1,1,0,1,1,0)$ are in R, but their conjunction $(1,0,0,0,0,0)$ is not.

10

(ii) $(1,0,1,0,0,0)$ and $(1,1,0,1,1,0)$ are in R, but their disjunction $(1,1,1,1,1,0)$ is not.

(iii) If we take $x = (1,0,1,0,0,0)$, $y = (1,1,0,1,1,0)$, and $z = (0,0,0,0,0,0)$, $(x \wedge y) \vee (x \wedge z) \vee (y \wedge z) = (1,0,0,0,0,0)$, which is not in R.

(iv) If we take $x = (1,0,1,0,0,0)$, $y = (1,1,0,1,1,0)$, and $z = (0,0,0,0,0,0)$, we have $x \oplus y \oplus z = (0,1,1,1,1,0)$, which is not in R.

(v) We take $x = (1,1,1,1,1,1)$, $y = (1,0,0,0,1,0)$, and $x' = (1,0,1,1,1,1)$, x and y are in R and x' is a step from x to y, but x' is not in R and there does not exist any $x'' \in R$ which is a step from x' to y. $\qquad\square$

The tubes. A *tube* is the block pictured in Figure 9; it is comprised of six tetrahedra that all have one edge in common, and contain each two normal disks: either a pair of one triangle and one quadrilateral, or a pair of triangles of different types. As in Figure 9, we denote by A_1 and A_2 the two pairs of faces that are crossed by two adjacent quadrilaterals.

Similarly as for the clause gadget, the gluings in a tube gadget at A_1 and A_2 can be specified by two variables, again with the convention of Figure 7: 0 and 1 respectively for non-crossing and crossing block curves.

Lemma 9. *The singular normal surface in a tube specified by the gluing is immersed if and only if both variables of the tube are equal.*

Proof. This is immediate by checking the four possible assignments of the variables. $\qquad\square$
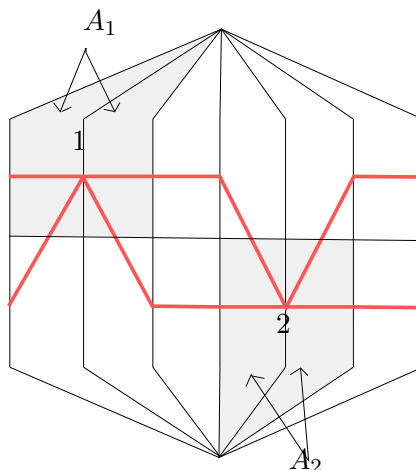

Figure 9: A tube, where the shaded regions denote the locations where the tube will be connected to clauses.

Consider a variable v appearing exactly twice in Φ, in clauses C_1 and C_2 (it may be that $C_1 = C_2$). Let G_1 and G_2 be the copies of the clause gadget corresponding to C_1 and C_2, respectively; let B_1 be the pair of faces in G_1 corresponding to the occurrence of the variable v in G_1, and similarly let B_2 be the other pair of faces in G_2 corresponding to the occurrence of the variable v in G_2. We create a tube T_v, and we glue the pair of faces A_1 to B_1 and A_2 to B_2.

The constants. A constant gadget is one of the blocks pictured in Figure 10 (since each gadget only consists of one tetrahedron, we did not adopt the schematic representation in this drawing). The gadget CG_0 for the constant 0 consists of one tetrahedron containing two triangles, while the gadget CG_1 for the constant 1 consists of one tetrahedron containing two crossing quadrilaterals. In both cases, the pair of faces A, at which the constant will be connected to a clause gadget, is made of the two front faces in Figure 10.
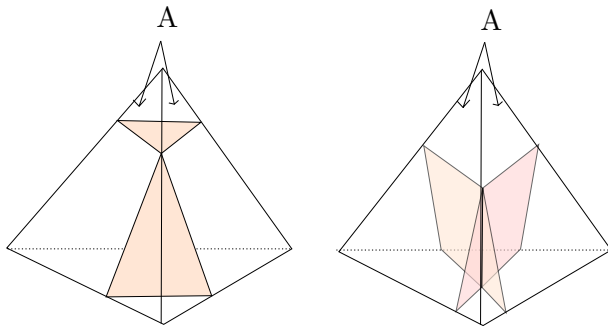
Figure 10: Left: The gadget CG_0. Right: The gadget CG_1. In both cases, the two front faces constitute the pair A. The trace of the singular normal surface on A is the same in both cases.

Whenever a constant 0 appears in Φ, we create a copy of the constant gadget CG_0, and attach the corresponding pair of faces in the clause gadget to the pair A in the constant gadget. The same holds for the constant 1, with CG_1 instead of CG_0.

3.2   Proof of the reduction

We now have all the tools to prove Theorem 6. Starting with a formula Φ, we build a triangulation T and normal coordinates N with the clause gadgets, the tubes, and the constant gadgets. We first prove that this triangulation forms a 3-manifold with boundary.

Proposition 10. *The triangulation T corresponding to a formula Φ is a 3-manifold with boundary.*

Proof of Proposition 10. First, we show that every vertex v of T has a neighborhood homeomorphic to the closed half-space. The vertex v is adjacent to a clause gadget C as well as between zero and three tubes or constant gadgets. We focus on the case with three tubes T_1, T_2 and T_3, the other ones being handled similarly. Since gadgets are glued along discs with disjoint interiors, and the tubes are locally disjoint except at their intersection with C, the neighborhood around v is the one pictured in Figure 11. One readily sees that this neighborhood is homeomorphic to a half-space.
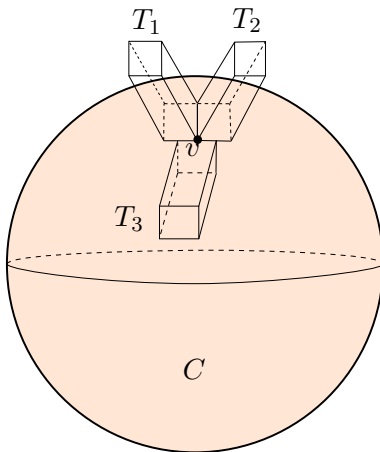

Figure 11: The local picture around a vertex v of the triangulation T.

Moreover, by construction, no edge is identified to itself in reverse. Lemma 1 concludes the proof. $\square$

Proposition 11. *The normal coordinates N correspond to an immersible surface if and only if the formula Φ is satisfiable.*

12

Proof of Proposition 11. We will use the following observation. Consider an edge e between the two faces forming a pair of faces in a gadget. The block around e is made of three or four tetrahedra, depending on whether a constant gadget or a tube is glued to the pair on the clause gadget. Then the singular surface has no branch point on e if and only if the values of the variables on the two pair of faces are identical, and, in case one of the gadget is a constant gadget, the variable takes the value specified by that gadget.

Assume first that the normal coordinates N correspond to an immersible surface. The gluings corresponding to variables appearing exactly once in Φ define a partial assignment of the variables. For the ones appearing twice, Lemma 9 implies that the two variables in the same tube are equal. Since they are glued to the pairs of faces representing their two occurrences in the clause gadgets, and by the observation, this defines a consistent assignment of the variables; moreover, again by the observation, the variables on the pair of faces on the clause gadgets corresponding to the constants take the appropriate values. Finally, since no branch point arises in a clause gadget, Lemma 7 implies that each clause is satisfied by the assignment of variables.

Conversely, assume that Φ is satisfiable. The satisfying assignment naturally defines the values of the variables on each pair of faces. We prove that these gluing rules do not create branch points. Branch points only occur on non-boundary edges of the triangulation; each such edge is either an edge in a clause gadget, an edge in a tube gadget, or an edge at the interface between two pairs of faces. No such edge contains contains a branch point, by Lemmas 7 and 9 and the observation, respectively. This concludes the proof. □

4 Variants

We now establish a few corollaries, settling the complexity of a few variants of the IMMERSIBIL-ITY problem.

The first one settles the complexity of the problem k-BOUNDED-IMMERSIBILITY, which is the one of testing immersibility when all the normal coordinates are bounded by $k \geq 1$.

Corollary 12. *The problem k-BOUNDED-IMMERSIBILITY is NP-complete.*

(Recall that a problem is *NP-complete* if it is both in NP and NP-hard.)

Proof of Corollary 12. The NP-hardness reduction for Theorem 6 only involves normal coordinates bounded by 1, so k-BOUNDED-IMMERSIBILITY is NP-hard as well.

The certificate we use to prove the membership in NP is the global gluing for an immersed surface. At an interface, the local gluing can be described by a permutation in S_k, which is of finite size since k is fixed. Since the number of interfaces is bounded by the size of the triangulation, the certificate has polynomial size. As mentioned in the preliminaries, when one is provided with normal coordinates and a global gluing, one can test in linear time whether the corresponding singular normal surface is immersed. □

The gadget we use to show the NP-hardness of IMMERSIBILITY only outputs a 3-manifold with boundary, but can be easily tweaked to handle the problem with boundaryless manifolds, which we name BOUNDARYLESS-IMMERSIBILITY.

Corollary 13. *The problem BOUNDARYLESS-IMMERSIBILITY is NP-hard.*

Proof of Corollary 13. The triangulation T_Φ with normal coordinates N obtained by the reduction in Theorem 6 can become boundaryless by *doubling* it, i.e., by taking $T_\Phi \bigsqcup T_\Phi$ and gluing one onto the other with the identity homeomorphism on their boundaries. It is straightforward to check that the resulting space is a 3-manifold, and that N is immersible in T_Φ if and only if $N \bigsqcup N$ is immersible in $T_\Phi \bigsqcup T_\Phi$. □
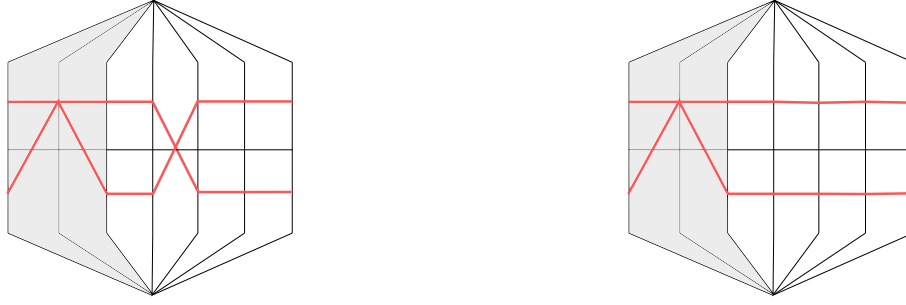
Figure 12: When these two blocks are made into a single triangulation by gluing them along the shaded regions, the normal coordinates are locally immersible around every edge, but not globally immersible.

As we mentioned in the preliminaries, the triangulations we consider in this chapter may not be simplicial complexes, and indeed the gadgets we use in our reduction may display some slightly pathological behavior. For example, if a tube links the variables 1 and 5 of a clause gadget, the central edge of the tube has its endpoints identified. This can be avoided by replacing each tube gadget by two copies of itself, glued to each other along the pairs A_1 and A_2. These "double tubes" are glued to the clause gadgets as usual ones. Then it can be checked that the resulting triangulation is a simplicial complex, and this shows that IMMERSIBILITY is also NP-hard when the triangulation is a simplicial complex.

As a last remark, we note that since the triangulation obtained by the reduction in Theorem 6 consists of balls linked by tubes, it can be embedded in $\mathbb{R}^3$. This shows that IMMERSIBILITY is NP-hard even when restricted to submanifolds of $\mathbb{R}^3$, which is for example the case of knot complements.

5 Testing local immersibility

In this section, we provide a polynomial-time algorithm to test whether normal coordinates are immersible when the triangulation is just a block, i.e., a collection of tetrahedra glued around a single edge. When applied to every edge of a more complicated triangulation, this provides a *local* test for immersibility, in that it detects local obstructions but not more involved ones, as pictured in Figure 12.

For the rest of this section, we denote by T a triangulation comprised of a single block, and by N some normal coordinates for this triangulation. The algorithm works by reducing the problem to a maximum flow computation in a directed graph with integer capacities. In this context, given a directed graph G where each edge e bears a nonnegative integer *capacity* $c(e)$, and two vertices s and t, a flow can be seen as a set of paths in G from s to t such that each edge e is used at most $c(e)$ times by the paths; a maximum flow is a flow with a maximum number of paths. It is well-known that maximum flows in directed graphs can be computed in small polynomial time [22, Ch. 10].

In our problem, the construction of the directed graph G corresponding to a block and to given normal coordinates is pictured in Figure 13: Every triangle type or quadrilateral type is associated to a directed edge, and the corresponding normal coordinate is carried over as the capacity of that edge. We denote by s_1 and s_2 the extremal vertices of this graph on one side of the block, and t_1 and t_2 the vertices on the other side. Note that in the triangulation, both sides of the block are identified, and therefore for $i = 1, 2$, s_i corresponds to the same vertex as t_i, which we denote by x_1 and x_2. The algorithm then simply computes the maximum flow between s_1 and t_1, and checks whether it is equal to the sum of the capacities of the edges going out of s_1.
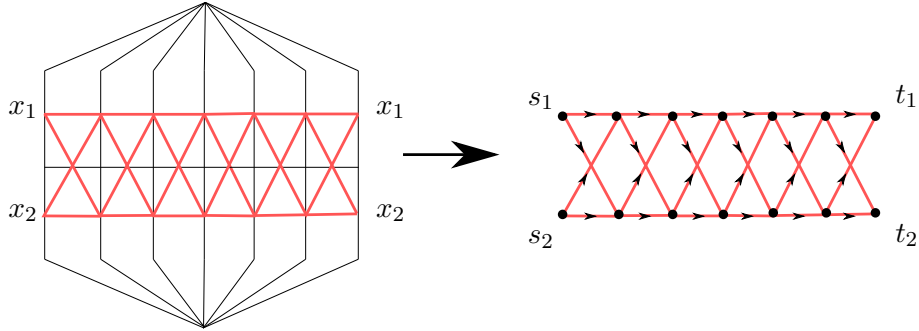
Figure 13: The directed weighted graph G is obtained by choosing an orientation around the block, adding a single directed edge for every normal disk type, and setting the capacity of an edge to be equal to the corresponding normal coordinate.

The correctness of this algorithm is warranted by the following lemma.

Lemma 14. *The normal coordinates N are immersible if and only if the maximum flow between s_1 and t_1 equals the sum of the capacities of the edges leaving s_1.*

Proof of Lemma 14. For the first implication, if the normal coordinates are immersible, there exists a gluing inducing no branch point. The associated normal surface is therefore a union of disks winding exactly once around the central edge. Every such disk corresponds in G to a path going from s_1 to t_1 or from s_2 to t_2. Therefore, the maximum flow going from s_1 to t_1 is the sum of the capacities going out of s_1.

In the other direction, such a maximum flow naturally defines a gluing corresponding to an immersed surface. Indeed, the flow can be seen as a set of paths in G from s_1 to t_1, which in turn correspond to cycles when identifying back s_1 and t_1. This gives a partial gluing of the normal disks, which yields a (not necessarily disjoint) union of embedded disks passing through x_1. It is also easy to see that, if we decrease the capacities of each edge of G by the number of times it is used by the flow, then, in this resulting *residual graph*, there exists a maximum flow from s_2 to t_2 that uses all residual capacities. (This could also be checked using the same algorithm.) This flow similarly corresponds to normal disks. All in all, we obtain an immersed surface whose normal coordinates are precisely the capacities of the original graph, which completes the proof. □

It is an intriguing problem to extend this algorithm to deal with more complicated triangulations. With some work, we can for example obtain a polynomial-time algorithm to test immersibility for the triangulation pictured in Figure 12. Finding a general polynomial-time algorithm is ruled out by our main Theorem 6 though. To conclude, it would be interesting to determine whether the immersibility problem is *fixed-parameter tractable* [10] in the size of the triangulation T; or, in other words, whether there exists an algorithm with running time $f(|T|).\text{poly}(n)$ for this purpose, where $f(|T|)$ is an arbitrary function depending on the number of simplices of T and $\text{poly}(n)$ is an arbitrary polynomial.

References

[1] I. AGOL, J. HASS, AND W. THURSTON, *The computational complexity of knot genus and spanning area*, Transactions of the American Mathematical Society, 358 (2006), pp. 3821–3850.

[2] I. R. AITCHISON, S. MATSUMOTO, AND J. H. RUBINSTEIN, *Surfaces in the figure-8 knot complement*, Journal of Knot Theory and Its Ramifications, 7 (1998), pp. 1005–1025.

[3] S. ARORA AND B. BARAK, *Computational complexity: a modern approach.* Draft of a book, 2006.09.21. See `http://www.cs.princeton.edu/theory/complexity/`, 2006.

[4] B. BURTON AND J. SPREER, *The complexity of detecting taut angle structures on triangulations*, in Proceedings of the Twenty-Fourth Annual Symposium on Discrete Algorithms (SODA), 2013, pp. 168–183.

[5] B. A. BURTON, T. LEWINER, J. A. PAIXÃO, AND J. SPREER, *Parameterized complexity of discrete morse theory*, in Proceedings of the 29th Annual Symposium on Computational Geometry (SOCG), 2013, pp. 127–136.

[6] É. COLIN DE VERDIÈRE, *Topological algorithms for graphs on surfaces*, habilitation thesis, École normale supérieure, 2012. available at `http://www.di.ens.fr/~colin/`.

[7] N. CREIGNOU, S. KHANNA, AND M. SUDAN, *Complexity classifications of Boolean constraint satisfaction problems*, SIAM Monographs on Discrete Mathematics and Applications, SIAM, 2001.

[8] V. DALMAU AND D. FORD, *Generalized satisfiability with limited occurrences per variable: a study through delta-matroid parity*, in Mathematical Foundations of Computer Science (MFCS), vol. 2747 of Lecture Notes in Computer Science, 2003, pp. 358–367.

[9] T. FEDER, *Fanout limitations on constraint systems*, Theoretical Computer Science, 255 (2001), pp. 281–293.

[10] M. R. FELLOWS, *Parameterized complexity: New developments and research frontiers*, in Aspects of Complexity, De Gruyter, 2001, pp. 51–72.

[11] C. GORDON, *The theory of normal surfaces*, 2001. Lecture notes, typeset by Richard Kent.

[12] W. HAKEN, *Theorie der Normalflächen, ein Isotopiekriterium für den Kreisnoten*, Acta Mathematica, 105 (1961), pp. 245–375.

[13] J. HASS, J. C. LAGARIAS, AND N. PIPPENGER, *The computational complexity of knot and link problems*, Journal of the ACM, 46 (1999), pp. 185–211.

[14] A. HATCHER, *Notes on basic 3-manifold topology.* Notes available on the author's webpage, 2007.

[15] G. KUPERBERG, *Knottedness is in NP, modulo GRH*, Adv. Math., (2014). To appear.

[16] D. M. LETSCHER, *Immersed normal surfaces and decision problems for 3-manifolds*, PhD thesis, University of Michigan, 1997.

[17] J. MATOUŠEK AND B. GÄRTNER, *Understanding and using linear programming*, Springer-Verlag, 2007.

[18] S. MATSUMOTO AND R. RANNARD, *The regular projective solution space of the figure-eight knot complement*, Experimental Mathematics, 9 (2000), pp. 221–234.

[19] E. E. MOISE, *Affine structures in 3-manifolds. V. The triangulation theorem and Hauptvermutung*, Annals of Mathematics. Second Series, 56 (1952), pp. 96–114.

[20] R. RANNARD, *Computing immersed normal surfaces in the figure-eight knot complement*, Experimental Mathematics, 8 (1999), pp. 73–84.

[21] T. J. SCHAEFER, *The complexity of satisfiability problems*, in Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC), 1978, pp. 216–226.

[22] A. Schrijver, *Combinatorial optimization. Polyhedra and efficiency*, vol. 24 of Algorithms and Combinatorics, Springer-Verlag, Berlin, 2003.

[23] J. Stillwell, *Classical topology and combinatorial group theory*, Springer-Verlag, New York, second ed., 1993.