

A Mathematical Webservice for Recognizing the Unknot

M. V. Andreeva ^{*}; I. A. Dynnikov [†] and K. Polthier [‡]

October 30, 2002

Abstract

In this paper we present the implementation of a partial knot recognition algorithm as a mathematical web service on the internet. Knots may interactively be loaded and edited, and then checked for being unknotted. We use the approach by Dynnikov, which is based on a combinatorial representation of knots in a three-page book, and we use JavaView as visualization environment.

1 Introduction

Among the most interesting algorithmic problems of topology is recognition of the unknot. Colloquially speaking this means to decide if a given knot is really knotted. Being interesting by their own, a strong mathematical application of knots is their usage in different representations of three-manifolds. For instance, any compact three-manifold without boundary can be represented as a covering of the three-sphere branched along some knot. In recent years knot theory has found interesting applications outside mathematics, for example, in biochemistry where the topological shape of knotted molecules sometimes contributes to their chemical properties.

In this paper we present the implementation of a knot recognition algorithm as a mathematical web service on the internet. The overall purpose of this activity is motivated by the following two goals:

1. We would like to use an experimental laboratory for the recognition of knots which is maintained on a central server but has world-wide access through the internet. This is essential for our own experiments since our co-authors happen to travel around and would hardly be able to synchronize their local computer or notebook with the latest version of the laboratory.

^{*}Institute of Information Systems, 630090 Novosibirsk, Russia, e-mail: mary@iis.nsk.su

[†]Mechanics and Mathematics Department, Moscow State University, Vorob'evy Gory, 119899 Moscow, Russia, e-mail: dynnikov@mech.math.msu.su

[‡]Institut für Mathematik, Technische Universität Berlin, Strasse des 17. Juni 136, 10623 Berlin, Germany, e-mail: polthier@math.tu-berlin.de

2. We would like to offer other researchers a simple access to perform own experiments without the need to run through software installation, or even worse, to repeated software installations when the laboratory is upgraded.

In section 2 we recall the principles behind the recognition algorithm, in section 3 we describe the basics of the web service and its components, and in section 4 we discuss in detail the functionality of the user interface and how to perform practical experiments.

2 Review of the Recognition Algorithm

Here a *knot* is a simple polygonal line in the Euclidean three-space \mathbb{R}^3 . Two knots Γ_1 and Γ_2 are called *isotopic* if there exists an orientation preserving piecewise linear homeomorphism of \mathbb{R}^3 to itself that sends Γ_1 to Γ_2 . In topology, knots are considered up to isotopy. The union of a few non-intersecting knots in the three-space is called a *link*. The notion of isotopy naturally extends to links.

A knot is said to be an *unknot* or a *trivial knot* if it is isotopic to a knot that lies in a plane. Otherwise we say that a knot is non-trivial. The simplest example of a non-trivial knot is the trefoil knot shown in Figure 1.

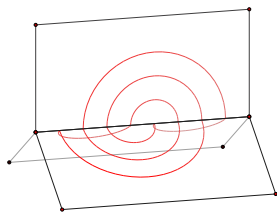


Figure 1: The trefoil knot

The problem of recognizing the unknot was theoretically solved by W. Haken who introduced an algorithm that has at least exponential complexity [8]. It was used by J. Birman and H. Hilden for recognizing the three-sphere among three-manifolds represented by genus two Heegaard splittings or, more general, by two-sheeted coverings of the three-sphere branched along links [3]. Some other topological problems need an effective algorithm for recognizing the unknot. Up to now no algorithm of polynomial complexity for solving this problem is known. The question of the existence of such an algorithm is an important and well-known unsolved problem of algorithmic topology.

One of the possible approaches to the problem consists in finding a *partial* algorithm for recognizing the unknot with polynomial running time and then see if one can improve it to be a real algorithm. ‘Partial’ formally means that the algorithm checks some sufficient condition for knot triviality. If the condition

is satisfied, the positive answer is given. If not, the algorithm ends up with an uncertain answer.

One of the most natural ways to construct a partial algorithm is by using some simplification of the knot presentation. The most naive approach, using simplification by Reidemeister moves, does not work already for knots with twenty crossings.

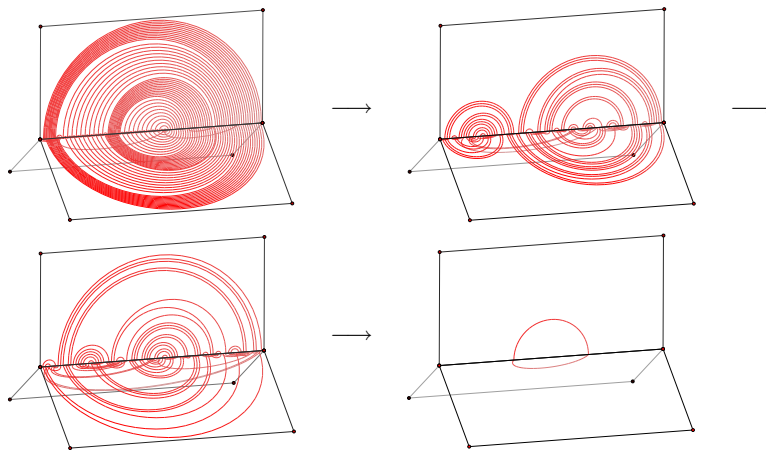
An partial algorithm based on simplification of a combinatorial presentation of the knot complement was proposed by S. V. Matveev.

Recently one of us (I.D.) proposed another partial algorithm for recognizing the unknot. It is based on a simplification of the knot with respect to a complexity different from the number of crossings. An advantage is the small running time for the algorithm, $O(\text{complexity}^3)$, which allows to run it for monster knots (having, say, 1000 crossings on a planar diagram). The result of the algorithm is a knot isotopic to the initial one that cannot be simplified any further by certain type transformations.

The estimation to what complexity the algorithm will distinguish between trivial and non-trivial knots is a hard problem. We tried to run it for a few monster unknots (see [4, 5] for an example) with hundreds of crossings and they have been successfully untangled. However, some unknots are not simplified completely by the algorithm. So far, we do not have such an example with crossing number < 50 . One of the purposes of our project is to visualize the simplifying process and possibly find new methods to improve the algorithm.

See the example of such procedure applied to one of the Goeritz diagrams on the figure 2.

Figure 2: Untangling Goeritz 13-21



The subject of our talk is exposition of its computer realization [1] which may serve as a good tool for working topologists. This package is based on the

JavaView library developed by one of us (K.P.) [2].

The approach is based on a simple and very well known observation that every link in the three-space is topologically equivalent to a link that lies entirely in a three-page book, i.e. the union of three half-planes with common boundary. Though being not convenient for human perception, this way of presenting links seems to be very efficient for handling knots by computer. It provides a very quick way from a combinatorial description of a link to its three-dimensional presentation. A three-page link admits a lot of transformations that preserve its isotopy class and can be easily found. This fact is used in the knot simplification tool which we are developing now. For algebraic aspects of the three-page representation of links we refer to [6, 7].

3 The Web Service

The web service at [1] consists of a Java applet embedded in a HTML web page. The Java applet is based on the Javaview software library plus an additional library which contains the knot recognition and editing functionality.

In contrast to other mathematical web services the software runs on the local computer of each user, and not on our server. This distinguishes our Java based approach from web services which are, for example, built with webMathematica. The advantage for us is a reduced requirement on the computation power of our web server, especially, if multiple users are accessing web services. The user has the benefit that he can choose what ever computational resources he needs by selecting an appropriate computer in his environment. But usually there is no big need to care about this issue on the user side. For smaller knots the simplification algorithms runs fast even on laptops, that means, basically the service is available everywhere on world.

4 Components of the User Interface

The user interface of the web services at [1] shows a 3D geometry viewer for interactive modifications on the currently investigated knot and a control panel to enable different functionalities:

1. view knots available in the database of the server
2. create your own knot by entering a 3D curve or a Gauss code
3. simplify or perform partial transformation a knot
4. interactively edit a knot.

We shall now explain them very briefly.

1. View a knot.

To download a knot from the server click ‘Choose knot’ button. At the present time, we have a collection of knots presented in jkb format, as 3D curve

(* .txt files), and as a Gauss code (* .gc files). Files 31.jkb, 141.jkb, 322.jkb, and 539 have been generated automatically from hand-made (I.D.) planar diagrams of unknots having 31, 141, 322, and 539 crossings, respectively. These unknots were used to test the simplifier.

The unknot with 539 crossings is a satellite of this unknot (see [4] for a picture). The source of the knots in txt format is the Knot-Plot web site, which is created by Robert Scharein [9] and contains a collection of knots and links with small crossing number (less than or equal to 10). The files with gc extension contain Gauss codes of the same links. These codes were generated automatically from the *.txt files by a program written by S.E.Koval. For converting a planar diagram to a three-page diagram the algorithm described in [4] is used.

2. Input a knot.

To input your own knot click ‘Input knot’ button and choose a method of input. Now available methods are using a 3D presentation and the Gauss code.

2.a. Input a knot as a 3D curve. To input a knot presented by a 3D polygon just type coordinates of all vertices in the order they follow on the knot. Coordinates must be separated by blank spaces or new lines. For example the trefoil knot is presented by a polygon with vertices (0, 0, 0), (3, 0, 1), (0, 2, 0), (1, 0, 1), (0, 4, 0), (4, 0, 1), (0, 1, 0), (2, 0, 1), (0, 3, 0), (0, 0, 1). To input it, type

0 0 0 3 0 1 0 2 0 1 0 1 0 4 0 4 0 1 0 1 0 2 0 1 0 3 0 0 0 1

and click ‘ok’ button. The Gauss code of the knot appears. Click ‘ok’ button again to see the three-page picture of the knot. If your link consists of more than one connected component, separate records for different components by a blank line. At the present state, for converting a 3D polygon to a planar diagram of a knot, projection along the z -axis is used. we would like to warn that if the projection of your knot along the z -axis is not generic (i.e. two vertices are projected to the same point, a vertex is projected to an internal point of an edge, or a triple crossing occurs), then the conversion may fail.

2.b. Input the Gauss code of a knot.

If you have planar diagram of a knot or link you may input it by typing its Gauss code. The diagram must be connected and have at least one crossing. We use an encoding a little different from what is usually called Gauss code, providing additional information that allows to distinguish knots from their mirror images. To input the link do the following. For each component, choose an orientation and a starting point. Assign different indices, which must be natural numbers, to all the crossings. For each connected component, go along it starting from the marked point in the chosen direction, and each time you come through a crossing, write down the index of the crossing preceded by two symbols: the first one must be + or – indicating whether you go over or under the crossing, and the second must be > or < sign indicating whether you see the arc that you intersect going from left to the right or from right to the left, respectively. Entries for different crossings must be separated by spaces. When you come back to the starting point of the chosen component, proceed

with the next one. Separate entries for different components by the keyword ‘component’.

3. Simplify a knot.

If you press the ‘Simplify’ button, the knot simplifier starts. It implements an algorithm described in [4]. The algorithm is based on a monotonic simplification of the three-page diagram by transformations that preserve the isotopy class of the knot (or link). Most frequently used transformations are of the following type. A connected component of the intersection of the link with a plane formed by two of the three pages is removed. Then the rest of the link is deformed within the three-page book so that the endpoints are kept fixed. Finally, the endpoints are connected by another curve in the same plane where the removed part of the link lied. At each step, we chose a transformation of this kind that leads to the simplest possible diagram, i.e. having the least number of points at the triple line.

At the present state, the simplifier does not search among all three-page diagrams that can be obtain from the current one by available transforms and does not make a “canonical” choice. So, if you click the ‘Simplify’ button again, the picture may change.

The running time for simplification of the knot of complexity n is bounded from above by $O(n^3)$.

4. Edit a knot.

The only available editing tool for now is a cyclic permutation of vertices on the binding line, which is achieved by pressing the ‘Vertex permutation’ button. Other editing tools are coming soon.

Of course, the developing of the Knot Simplifier is still in progress and more things have to be done such as:

1. introducing more editing tools;
2. converting to/from other formats;
3. comparing and recognizing some simple knots which are not unknots.

Acknowledgement 1. *This project was supported by Sonderforschungsbereich 288 “Differential Geometry and Quantum Physics” at Technische Universität in Berlin. We also would like to thank S. E. Koval and I. A. Taimanov who collaborated in this project at its different stages.*

References

- [1] The Knot Simplifier web service is available at <http://www.javaview.de/services/knots>.
- [2] K. Polthier, S. Khadem-Al-Charieh, E. Preuß, and U. Reitebuch. Publication of interactive visualizations with JavaView. In J. Borwein, M. H. Morales, K. Polthier, and J. F. Rodrigues, editors, *Multimedia Tools*

for *Communicating Mathematics*. Springer Verlag, 2002. <http://www.javaview.de>.

- [3] J. Birman, H. Hilden. Heegaard splittings of branched coverings of S^3 . *Transactions of the Amer. Math. Soc.* **213** (1975), 315–352.
- [4] I. A. Dynnikov. Three-page link presentation and an untangling algorithm. In: Proc. of the International Conference *Low-Dimensional Topology and Combinatorial Group Theory*, Chelyabinsk, July 31–August 7, 1999; Kiev, 2000, pp. 112–130.
- [5] I. A. Dynnikov. A new way to represent links. One-dimensional formalism and an untangling technology, *Acta Applicanda Mathematicae*, to appear.
- [6] I. A. Dynnikov. Three-page approach to knot theory. Encoding and local moves. *Functional Analysis and Its Appl.* **33:4** (1999), 260–269.
- [7] I. A. Dynnikov. Three-page approach to knot theory. Universal semigroup. *Functional Analysis and Its Appl.* **34:1** (2000), 24–32.
- [8] W. Haken. Theorie der Normalflächen. Ein Isotopiekriterium für der Kreisknoten. *Acta Math.* **105** (1961), 245–275.
- [9] R. Scharein. The Knot-Plot Homepage <http://www.pims.math.ca/knotplot>.