Computing invariants of knotted graphs given by sequences of points in 3-dimensional space

Vitaliy Kurlin

Abstract We design a fast algorithm for computing the fundamental group of the complement to any knotted polygonal graph in 3-space. A polygonal graph consists of straight segments and is given by sequences of vertices along edge-paths. This polygonal model is motivated by protein backbones described in the Protein Data Bank by 3D positions of atoms. Our KGG algorithm simplifies a knotted graph and computes a short presentation of the Knotted Graph Group containing powerful invariants for classifying graphs up to isotopy. We use only a reduced plane diagram without building a large complex representing the complement of a graph in 3-space.

1 Introduction: our motivations, key concepts and problems

This research is on the interface between knot theory, algebraic topology, homological algebra and computational geometry. Our main motivation is the application of topological and algebraic methods to recognizing knotted structures in 3-dimensional geometric graphs of long molecules such as protein backbones.

Backbones of proteins are polygonal curves in 3-space. A *protein* is a large molecule containing a big number of amino acid residues. The primary structure or the *backbone* of a protein is the linear sequence of its amino acids. More than 100K proteins have been tabulated in the Protein Data Bank http://www.rcsb.org/pdb, which is a large database of pdb files. The *pdb file* of a single protein contains noisy coordinates (x,y,z) of all atoms that are linearly ordered in the backbone.

A natural way to model a protein is to assume that each atom is a point in 3-dimensional Euclidean space $\mathbb{R}^3$, while every chemical bond between atoms is a straight line segment between corresponding points. In general, a *polygonal* curve

Microsoft Research Cambridge, 21 Station Road, Cambridge CB1 2FB, United Kingdom and Durham University, Durham DH1 3LE, UK. e-mail: vitaliy.kurlin@gmail.com, http://kurlin.org

with vertices $p_1,\dots,p_m \in \mathbb{R}^3$ is the union of line segments connecting each point p_{i-1} with p_i for $i = 2,\dots,m$. In addition, if $p_0 = p_m$, we get a closed curve in $\mathbb{R}^3$.

Definition 1 *A* polygonal knotted *graph is any embedded graph $K \subset \mathbb{R}^3$ consisting of finitely many straight line segments with pairwisely disjoint interiors. The number n of line segments in K is called the* length *of the polygonal graph $K \subset \mathbb{R}^3$.*

The *degree* of a vertex v in a graph K is the number $\deg v$ of edges attached to v, and a loop is counted twice. Vertices with $\deg \neq 2$ are *essential*. An *edge-path* of K is a polygonal chain with essential vertices at 2 endpoints and only *non-essential* vertices of degree 2 between them. The open trefoil in Fig. 1 is the edge-path with 4 non-essential vertices v_1, v_2, v_3, v_4 between 2 essential vertices v_0, v_5 of degree 1. In practice, a polygonal graph K in 3-space is represented in a computer memory as

- an unordered list of points (x,y,z) corresponding to all essential vertices of K;
- a sequence of points (x,y,z) at non-essential vertices along every edge-path of K.

The edge-path of the trefoil K in Fig. 1 is represented by the sequence v_1, v_2, v_3, v_4.

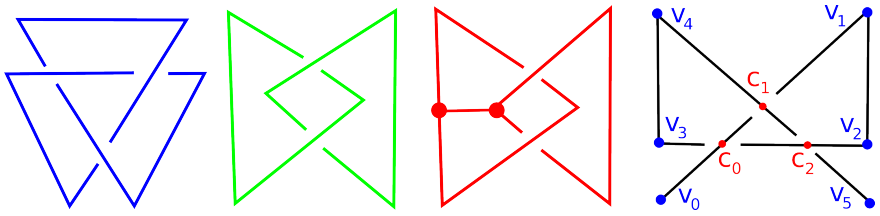

Fig. 1 Knotted polygonal graphs (trefoil, Hopf link, Hopf graph) and open trefoil with vertices $v_0 = (-2,-2,1)$, $v_1 = (2,2,-1)$, $v_2 = (2,-1,0)$, $v_3 = (-2,-1,0)$, $v_4 = (-2,2,2)$, $v_5 = (2,-2,-1)$ in $\mathbb{R}^3$ and crossings $c_0 = (-1,-1)$, $c_1 = (0,0)$, $c_2 = (1,-1)$ in the (x,y)-plane $\mathbb{R}^2$.

If the graph K is a circle, then $K \subset \mathbb{R}^3$ is a *knot*. If K is a disjoint union of several circles, then $K \subset \mathbb{R}^3$ is a *link*. Knotted graphs are usually studied up to *isotopy* that is a continuous deformation of $\mathbb{R}^3$ moving one graph to another, see Definition 3.

Recognition problem for protein backbones and knotted graphs in 3-space. To distinguish different knots or graphs $K \subset \mathbb{R}^3$ up to isotopy, mathematicians construct *knot invariants* that should take the same value on all knots isotopic to each other. If such an invariant has different values on two knots, these knots are different.

The simplest non-trivial invariant is the number of connected components of a graph $K \subset \mathbb{R}^3$, which is preserved under any continuous deformation of $\mathbb{R}^3$. Hence a knot is not equivalent to a link consisting of at least 2 circles. However, this simple invariant can not distinguish any knots, so more powerful invariants are needed. A knot invariant can be called *complete* if it distinguishes all knots up to isotopy.

The complement $\mathbb{R}^3 - K$ of a knotted graph is 3-dimensional and contains more information about the isotopy class of K in the ambient space $\mathbb{R}^3$ than the 1-dimensional graph K itself. The oldest invariant of a knot $K \subset \mathbb{R}^3$ is the *fundamental*

group of the knot complement $\mathbb{R}^3 - K$. Briefly, this group describes algebraic properties of closed loops that go around K in $\mathbb{R}^3$ and can be continuously deformed without intersecting K, see Definition 6. The *Alexander polynomial* of K is a simpler invariant that can be extracted from the fundamental group [2]. We highlight the advantages of the fundamental group over combinatorial invariants of knots.

- The group $\pi_1(\mathbb{R}^3 - K)$ is defined for any graph $K \subset \mathbb{R}^3$, not only for knots, and is an almost complete invariant of the isotopy class of K, see Theorems 7, 8, 9.
- Many invariants of knots $K \subset \mathbb{R}^3$ are introduced in terms of a *plane diagram*, which is a projection of K to $\mathbb{R}^2$ with only *double crossings*. These invariants are often computed in time exponential with respect to the number of crossings.
- Despite the group $\pi_1(\mathbb{R}^3 - K)$ is non-abelian, it leads to numerous abelian invariants that distinguish all prime knots with up to 11 crossings, see Theorem 12, using practically efficient algorithms from the HAP package of GAP [3].

Contributions of the current work to recognizing knotted graphs. Our input is any knotted polygonal graph K, which is motivated by real-life knotted structures. Our preferred invariant is the fundamental group $\pi_1(\mathbb{R}^3 - K)$ and is justified above. Our main result (Theorem 2 below) is a robust algorithm for a guaranteed fast computation of this almost complete invariant for arbitrary knotted graphs $K \subset \mathbb{R}^3$.

Theorem 2 *Given any polygonal graph $K \subset \mathbb{R}^3$ of a length n, our KGG algorithm first simplifies K to a small diagram with c crossings in time $O(n^2)$ and then writes a short presentation of the Knotted Graph Group $\pi_1(\mathbb{R}^3 - K)$ in time $O(c)$.*

The KGG algorithm and a proof of Theorem 2 are presented in Section 4. We highlight the improvements over the related past work, see more details in Section 3.

- We work with a Gauss code of a knotted graph $K \subset \mathbb{R}^3$ without modelling the complement $\mathbb{R}^3 - K$ by a cubical complex at a fixed resolution as in [1] and speed up the running time from seconds to milliseconds on a similar laptop, see Table 3.
- The fundamental group $\pi_1(\mathbb{R}^3 - K)$ is more powerful than the Alexander polynomial, which was used for recognising knotted proteins in the KnotProt [6].
- We substantially extend the KMT algorithm [9, 18], which smooths polygonal curves, to a simplification of any polygonal graph $K \subset \mathbb{R}^3$. Our implementation handles round-off errors much better than the state-of-the-art version in [8].

The KGG algorithm can fit well in a future version of the Homological Algebra Programming package (HAP) of GAP: Groups, Algorithms, Programming [3]. Moreover, the KGG algorithm can be used for connecting the Rosetta software (predicting protein structures as geometric graphs in 3-space) with the state-of-the-art recognition algorithm of trivial knots at http://www.javaview.de/services/knots.

This knot recognition is based on 3-page embeddings whose full theory was already extended to knotted graphs in $\mathbb{R}^3$ [13]. Gauss codes of knotted proteins produced by the KGG algorithm in this paper can be the input for the linear time algorithm [11] drawing 3-page embeddings of graphs. Hence we can visualize knotted proteins in a *3-page book* (a union of 3 half-planes with the same boundary line).

2 Background on topological invariants of knotted graphs

Knot theory: equivalences and plane diagrams of knotted graphs A *homeomorphism* is a bijection $f : X \to Y$ such that both f, f^{-1} are continuous. It is convenient to consider knots and graphs in the compact sphere S^3, which is obtained from $\mathbb{R}^3$ by adding a point at infinity, so $S^3 - \{\text{any point}\} \approx \mathbb{R}^3$ are homeomorphic.

Definition 3 *Two knotted graphs $K, K' \subset S^3$ are called* equivalent *if there is a homeomorphism $f : S^3 \to S^3$ taking K to K', so $f(K) = K'$. The graphs $K, K' \subset S^3$ are* ambiently isotopic *if the above homeomorphism also preserves an orientation of S^3 or, equivalently, there is an* ambient isotopy *that is a continuous family of homeomorphisms $f_t : S^3 \to S^3$, $t \in [0,1]$, such that $f_0 = \mathrm{id}$ on S^3 and $f_1(K) = K'$.*

The two mirror images of a trefoil are equivalent, but not isotopic, see a short proof in [4]. A knot $K \subset S^3$ is *trivial* (or the *unknot*) if K is isotopic to a round circle. The main problem in knot theory is to classify knots and more general knotted graphs up to equivalence or ambient isotopy from Definition 3. A *plane diagram* of a knotted graph $K \subset S^3$ is the image of K under a projection to a horizontal plane $\mathbb{R}^2$ in a general position having only transversal intersections (*double crossings*).

At each crossing we specify a short arc that crosses over another arc, see Fig. 1. The natural visual complexity of the isotopy class of a knotted graph $K \subset \mathbb{R}^3$ is the minimum number of crossings over all plane diagrams representing the graph K.

Knot recognition: Reidemeister moves and Gauss codes of graphs. For the KGG algorithm in Section 4, we use the Reidemeister move R1 from generalized Reidemeister's Theorem 4 below saying that any isotopy of knotted graphs in $\mathbb{R}^3$ can be realized by a finite sequence of moves on plane diagrams in Fig. 2.

Theorem 4 *[7] Two plane diagrams represent isotopic knotted graphs in 3-space $\mathbb{R}^3$ if and only if the diagrams can be obtained from each other by an isotopy in (a continuous deformation of) $\mathbb{R}^2$ and finitely many Reidemeister moves in Figure 2. (The move* R5 *is only for rigid graphs, the move* R5′ *is only for non-rigid graphs.)*

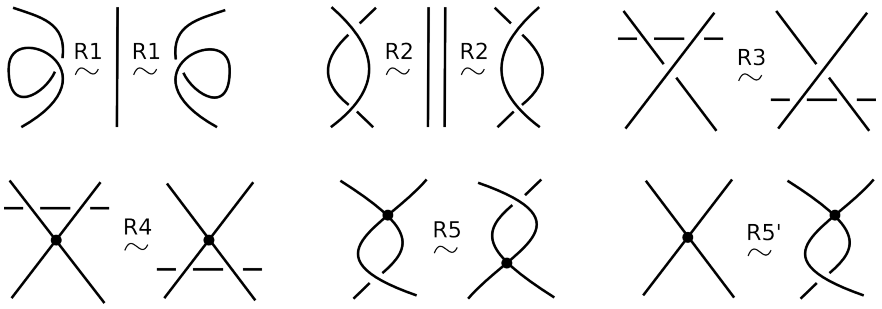

Fig. 2 Reidemeister moves on plane diagrams of knotted graphs, see Theorem 4.

The move R4 in Fig. 2 is for a vertex of degree 4 and similarly works for other degrees. The move R5 turns a small neighborhood of a vertex upside down. So a cyclic order of edges at vertices is preserved by R5. The move R5′ can reorder all edges at a vertex. Theorem 4 includes all symmetric images of moves in Fig. 2.

As described in [12], we shall encode a diagram of a knotted graph $K \subset \mathbb{R}^3$ by a simple Gauss code, which will be later converted into a presentation of the Knotted Graph Group $\pi_1(\mathbb{R}^3 - K)$. If a graph K contains a circle S^1 that is disjoint with the rest of the graph, then one of degree 2 vertices on S^1 will be *essential* so that the circle can be formally considered as an edge-path from this vertex to itself.

Definition 5 *Let $D \subset \mathbb{R}^2$ be a plane diagram of a knotted graph K with only double crossings and essential vertices $A, B, C, \ldots$ of degree not equal to 2. We fix directions of all edge-paths in K and arbitrarily label all crossings of D by $1, 2, \ldots, l$. The* Gauss code *of D consists of all words W_{AB} associated to directed edge-paths AB of K from one essential vertex A to another essential vertex B as follows, see Fig. 3:*

- *W_{AB} starts with A, finishes with B and has the labels of all crossings in AB;*
- *if the edge-path AB goes under another edge-path of the graph K at a double crossing i, then we add the negative sign in front of the label i in the word W_{AB}.*

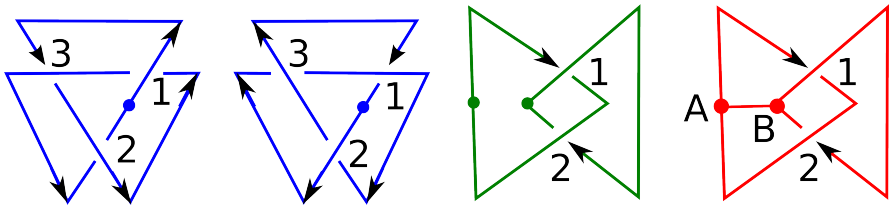

Fig. 3 Plane diagrams with directed edge-paths and labeled crossings illustrating Definition 5.

The neighbors (vertices or crossings) of each vertex A are clockwisely ordered in $\mathbb{R}^2$, so the Gauss code specifies a cyclic order of all words starting or finishing at A.

The trefoils in Fig. 3 have codes $(1,-3,2,-1,3,-2)$ and $(2,-3,1,-2,3,-1)$, which are defined up to cyclic permutations. The Hopf link has the Gauss code consisting of 2 words $(1,-2)$ and $(-1,2)$. The Hopf graph has the Gauss code consisting of 3 words corresponding to the 3 edges: (A,B), $(A,-1,2,A)$, $(B,1,-2,B)$.

The fundamental group and abelian invariants of a graph complement in S^3.

Definition 6 *Let $X \subset \mathbb{R}^3$ be a path-connected subset, so any two points in X can be connected by a continuous path within X. A closed* loop *at a base point $p \in X$ is a continuous map $f : [0,1] \to X$ with $f(0) = p = f(1)$. Two such loops $f_0, f_1 : [0,1] \to X$ are* path-homotopic *if they can be connected by a continuous family of loops $f_t : [0,1] \to X$, $t \in [0,1]$, always passing through the base point $p = f_t(0) = f_t(1)$ for $t \in [0,1]$. The* fundamental group *$\pi_1(X,p)$ is the group of all path-homotopy classes of closed loops in X. The product of two loops is obtained by going along the first loop (starting and finishing at the base point p), then along the second loop.*

A *connected sum* $K\#K'$ of knots K,K' is obtained by removing 2 short open arcs $a \subset K$, $a' \subset K'$ and by joining the resulting 4 endpoints to form a larger knot $(K-a)\cup(K'-a')$, see Fig. 4. The isotopy class of $K\#K'$ depends only on the isotopy classes of K,K', not on a choice of a,a'. A knot not isotopic to a connected sum of non-trivial knots is called *prime*. Any knot uniquely decomposes into a connected sum of prime knots (up to permutations), hence only prime knots are classified.

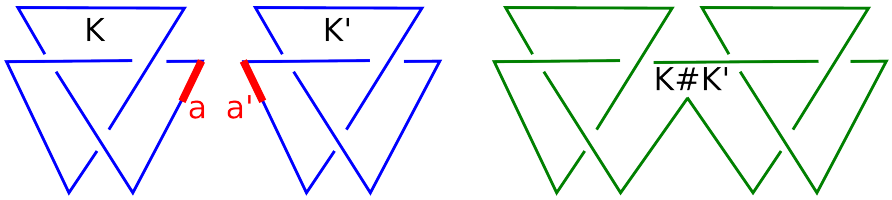

Fig. 4 A connected sum $K\#K'$ of 2 trefoils K and K' is well-defined up to ambient isotopy in $\mathbb{R}^3$.

Table 1 Exact numbers of prime non-trivial knots from http://www.indiana.edu/~knotinfo

Number of crossings	≤ 6	7	8	9	10	11	12	≤ 12
Knot isotopy classes	7	7	21	49	165	552	2176	2977

Theorems 7, 8 imply that $\pi_1(S^3-K)$ is a complete invariant for all prime knots. So $\pi_1(S^3-K)$ and its abelian invariants can be used for recognizing knots. For a knotted graph $K \subset S^3$, let $N(K)$ be a small open neighborhood of the graph K. For instance, this neighbourhood can be the open ε-*offset* $K^\varepsilon = \cup_{p\in K} B(p;\varepsilon)$ consisting of open balls with a small radius $\varepsilon > 0$ and centers at all points $p \in K$. The complement $S^3 - N(K)$ is a compact 3-manifold whose boundary is $\partial N(K)$.

Theorem 7 *[5, Theorem 1] Two knots $K,K' \subset \mathbb{R}^3$ are equivalent* if and only if *there is a homeomorphism between their complements $S^3 - N(K) \approx S^3 - N(K')$. Two knots $K,L \subset \mathbb{R}^3$ are ambiently isotopic if and only if there is an orientation-preserving homeomorphism between their complements $S^3 - N(K) \approx S^3 - N(K')$.*

Theorem 8 *[20] If prime knots $K,K' \subset S^3$ have isomorphic groups $\pi_1(S^3-K) \cong \pi_1(S^3-K')$, then their complements are homeomorphic: $S^3 - K \approx S^3 - K'$.*

The Knotted Graph Group $\pi_1(S^3-K)$ is almost a complete invariant in the sense that a *peripheral structure* of $\pi_1(S^3-K)$ should be also preserved under a group isomorphism. Peripheral structures are completely characterised for links in [14].

We will assume that a knotted graph $K \subset S^3$ (if disconnected) is not *splittable*, namely K is not equivalent to a graph whose components are located in disjoint balls in S^3. The complement of any splittable graph K contains a sphere $S^2 \subset S^3 - N(K)$

separating components of K, so $S^3-N(K)$ can be simplified by cutting S^2. The complement of any non-splittable graph can not be simplified in this way. In this case any $S^2 \subset S^3-N(K)$ is called *incompressible* and $S^3-N(K)$ is called *irreducible*.

A cycle $C \subset K$ in a knotted graph $K \subset S^3$ is *trivial* if the knot C in $C \cup (S^3-K)$ is trivial, namely C bounds a topological disk D^2 in S^3-K. If a knotted graph K has a trivial cycle C, we can compress the complement $S^3-N(K)$ along the disk D^2 spanning C, so $S^3-N(K)$ can be simplified by cutting D^2. The complement of K without trivial cycles can not be simplified in this way. In this case $\partial(S^3-N(K))$ is called *incompressible* and $S^3-N(K)$ is called *boundary-irreducible*.

Theorem 9 *[19, Corollary 6.5] For two non-splittable knotted graphs $K,K' \subset S^3$ without trivial cycles, let $\phi : \pi_1(S^3-N(K)) \to \pi_1(S^3-N(K'))$ be an isomorphism that descends to an isomorphism $\pi_1(\partial(S^3-N(K))) \to \pi_1(\partial(S^3-N(K')))$. Then there is a homeomorphism $S^3-N(K) \approx S^3-N(K')$ inducing the isomorphism ϕ.*

Theorems 7, 9 imply that K,K' are equivalent. So the Knotted Graph Group $\pi_1(S^3-K)$ is an almost complete invariant (complete with a peripheral structure).

Theorem 10 *Any finitely generated abelian group Z is isomorphic to a direct sum of cyclic groups $\mathbb{Z}^r \oplus \mathbb{Z}_{q_1} \oplus \cdots \oplus \mathbb{Z}_{q_l}$, where $r \geq 0$ is the* rank *and $q_1,\dots,q_l$ are powers of primes. The numbers $r,q_1,\dots,q_l$ are called the* abelian invariants *of the group Z and are uniquely determined by Z up to a permutation of indices $q_1,\dots,q_l$.*

The above classification theorem says that any finitely generated abelian group can be completely described by its abelian invariants (a set of integers) and leads to numerous abelian invariants below that can be extracted from a non-abelian group G and efficiently computed by GAP if G has a short enough presentation [3].

Definition 11 *The* index *of a subgroup H in a group G is the number of disjoint cosets $gH = \{gh \mid g \in G,\ h \in H\}$ that fill the group G. The* abelianization *of H is the quotient $H/[H,H]$ over the* commutator subgroup *$[H,H]$ generated by all $[a,b] = aba^{-1}b^{-1}$, $a,b \in H$. The* abelian invariants *of a non-abelian group G are the abelian invariants of $H/[H,H]$ over all subgroups $H \subset G$ up to a certain index.*

3 Past work on computing invariants of knotted proteins

Standard KMT algorithm for shortening a knotted protein backbone. The KMT algorithm is named after Koniaris and Muthukumar [9] and Taylor [18], though their methods are different. Taylor [18] actually suggested how to smooth a protein backbone. Namely, each vertex B with two neighbours A,C is iteratively replaced by the center of the triangle $\triangle ABC$, which visually smooths an original polygonal curve K. The standard KMT algorithm simply shortens K replacing the chain ABC by the single edge AC when the isotopy class of K is preserved.

We discuss the implementation of the KMT algorithm [8] used in the Rosetta program predicting structures of proteins at https://www.rosettacommons.org. One

orders all degree 2 vertices $v_1,\dots,v_l$ according to the distance between their only neighbors A,C. Then the triangle $\triangle ABC$ based on a shortest segment AC is likely to be small and will probably not intersect any edge of K, see Fig. 5.

To check a potential intersection of $\triangle ABC$ with another edge DE, the plane ABC is intersected with the infinite line through DE. Finding an exact intersection point P requires divisions and leads to floating point errors, especially when DE is almost parallel to the plane ABC. Then three angles $\angle APB$, $\angle APC$, $\angle BPC$ are computed by using the arccos function, which also quickly accumulates computational errors.

Now the point P is inside the triangle $\triangle ABC$ if and only if the sum of 3 angles is $2\pi = \angle APB + \angle APC + \angle BPC$. In practice, for points P inside $\triangle ABC$, the above sum is only close to 2π, so the width of $3 \cdot 10^{-4}$ is used to handle round off errors.

In Section 4 we extend KMT to the KGG (Knotted Graph Group) algorithm using only additions and multiplications without evaluations of complicated functions. We have checked that our algorithm correctly runs on similar protein backbones from the PDB database with the much smaller error of only 10^{-10}, see Section 5.

Alexander polynomial of knotted proteins in KnotProt [6]. The knot recognition of polygonal graphs $K \subset \mathbb{R}^3$ in the largest database KnotProt of knotted proteins is based on the Alexander polynomial [2, section 8.3], which is a polynomial invariant of the fundamental group $\pi_1(\mathbb{R}^3 - K)$. Historically, there were no efficient algorithms to compare non-abelian groups up to isomorphism, hence a cubic computational time for the Alexander polynomial was acceptable. Moreover, the Alexander polynomial indeed classifies all knots with up to 8 crossings.

However, the Alexander polynomial attains only 550 different values on 801 prime non-trivial knots (without mirror images) up to 11 crossings. So we feel that the time has come for more powerful invariants, especially due to the efficient algorithms in GAP [3]. The following experimental result by Brendel et al. [1] demonstrates the power of the fundamental group for a practical classification of knots.

Theorem 12 *[1, Theorem 2] The abelianizations of subgroups with an index up to 6 in the fundamental group $\pi_1(\mathbb{R}^3 - K)$ distinguish all 801 prime non-trivial knots (up to mirror image) with plane diagrams having up to 11 crossings.*

Best methods for enumerating knots are based on triangulations of knot complements with a hyperbolic metric, which is not adapted yet for knotted graphs.

Discrete Morse theory for computing the fundamental group of a complex. Brendel et al. [1] suggested a general algorithm for computing the fundamental group of any regular cell complex. The algorithm uses a discrete Morse theory and is practically fast, though the theoretical complexity was hard to determine.

A protein backbone was modelled by a cubical knot $K \subset \mathbb{R}^3$, which is a union of small cubes at a fixed manually chosen resolution. For instance, the complement of the protein backbone $1V2X$ with joined endpoints in $\mathbb{R}^3$ was represented as a cubical complex C with 5674743 cells. This 3-dimensional complex C is deformed through several stages to a regular 2-dimensional complex C''' with 30743 cells. The

time for computing the knot group of $1V2X$ is about 35 seconds [1, section 5], while our KGG algorithm takes 67 milliseconds on a similar laptop, see Table 2.

Here are the key differences between our new approach and past work [1, 6, 8].

- The KMT algorithm only shortens a linear chain, while our KGG algorithm simplifies any knotted graph K and computes $\pi_1(\mathbb{R}^3 - K)$, which is more powerful than the Alexander polynomial used for recognizing knotted proteins in [6].
- Our KGG algorithm avoids evaluations of complicated functions and better handles floating point errors than KMT, also using the Reidemeister move R1 for extra reductions in the overall size of a knotted polygonal graph $K \subset \mathbb{R}^3$.
- We compute a simple presentation of the fundamental group $\pi_1(\mathbb{R}^3 - K)$ by working with only a given knotted polygonal graph $K \subset \mathbb{R}^3$ without modelling the complement $\mathbb{R}^3 - K$ as a large cubical complex at a fixed resolution as in [1].

4 KGG algorithm for computing the Knotted Graph Group

The input is a knotted polygonal graph $K \subset \mathbb{R}^3$ given by sequences of vertices along edge-paths. The output is a presentation of the Knotted Graph Group $\pi_1(\mathbb{R}^3 - K)$ with generators and relations. Here is a high-level description of all the stages.

1. In a given graph $K \subset \mathbb{R}^3$, identify all non-essential vertices that can be removed keeping the isotopy class of K after computing only five 3-by-3 determinants.
2. For a simplified graph $K' \subset \mathbb{R}^3$, find all crossings in a plane diagram of K'. Going along K', compute the Gauss code using the found crossings in the plane diagram of K'. Apply the Reidemeister move R1 for a further reduction if possible.
3. Turn a Gauss code into a presentation of the fundamental group $\pi_1(\mathbb{R}^3 - K)$ whose abelian invariants can be calculated using efficient algorithms of GAP.

Stage 1: robust algorithm for shortening a polygonal graph. Each degree 2 vertex B of a graph K has 2 neighbours, say A, C. We process all non-essential vertices B in the increasing order of $|AC|$. In comparison with the KMT algorithm, we much more robustly check if the interior of the triangle $\triangle ABC$ meets any edges of K.

For any edge DE with endpoints $D, E \notin \{A, B, C\}$, first we check if D, E are on different sides of the plane ABC. It is enough to compute the signed volumes of the tetrahedra $ABCD$ and $ABCE$, see Fig. 5. The volume V_{ABCD} is proportional (with factor $\frac{1}{6}$) to the 3-by-3 determinant whose columns are formed by the 3 coordinates of the 3 vectors $\overrightarrow{AB}, \overrightarrow{AC}, \overrightarrow{AD}$. The points D, E are on different sides of the plane ABC if and only if the signed volumes V_{ABCD} and V_{ABCE} have opposite signs.

If the edge DE intersects the plane ABC, we should check whether the intersection is inside the triangle $\triangle ABC$. Here is a simple geometric criterion: the edge DE meets the triangle $\triangle ABC$ if and only if the 3 tetrahedra $ABDE$, $ACDE$, $BCDE$ in Fig. 5 cover the union of the tetrahedra $ABCD$ and $ABCE$ *without any overlap*.

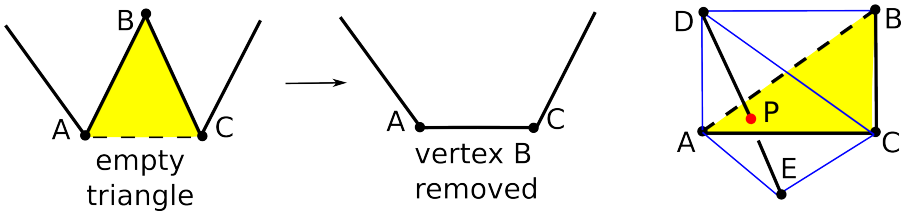

Fig. 5 Left: removing vertex B when $\triangle ABC$ is empty. Right: $ABCDE$ splits into 3 tetrahedra.

Such a geometric splitting is equivalent to the following algebraic identity between unsigned volumes: $|V_{ABCD}|+|V_{ABCE}| = |V_{ABDE}|+|V_{ACDE}|+|V_{BCDE}|$. Hence it remains to compute only 3 more 3-by-3 determinants for the quadruples $ABDE$, $ACDE$, $BCDE$. All computations involve only basic additions and multiplications.

Stage 2: computing a Gauss code of a reduced plane diagram of K. Let $K \subset \mathbb{R}^3$ be a simplified polygonal graph obtained by all possible shortenings at Stage 1 above. Now we simply project K' to the (x,y)-plane $\mathbb{R}^2$ finding all intersections between straight edges in the projection of K'. If the plane diagram of K' is not in a general position, we slightly perturb its vertices to guarantee that we have only double crossings, because we are interested only in the isotopy class of $K' \subset \mathbb{R}^3$.

This stage requires a quadratic time $O(m^2)$ in the length m of the simplified graph K', because we check all potential pairwise intersections of non-adjacent edges. In experiments on protein backbones in Section 5, the chain K' is much shorter than the original backbone K, hence this stage is fast enough in practice. For each edge $[v_i, v_{i+1}]$ of K', we build a list of crossings with other edges $[v_j, v_{j+1}]$. We keep this list of crossings in order from the vertex v_i to v_{i+1}. Apart from the actual coordinates (x,y) of a crossing, we also note the corresponding indices i, j and the heights z_i, z_j of the points in the intersecting edges above the crossing (x,y).

After completing these ordered lists over all edges, we can go along each edge-path of K' and assign a correct label to every crossing, because we can recognize if a crossing has been passed before. Since we kept actual heights z_i, z_j at each crossing, we can add negative signs to all undercrossings as needed by Definition 5.

Finally, if a Gauss code contains a consecutive pair of labels $(l,-l)$ or $(-l,l)$, the plane diagram contains a small loop that can be easily removed by the Reidemeister move R1 in Fig. 2. Assume that this crossing (x,y) in the move R1 is formed by (projections of) edges $[v_i, v_{i+1}]$ and $[v_j, v_{j+1}]$ for $i+1 < j$. Then we can shorten these edges by continuously moving the endpoints v_{i+1} and v_j towards the points (at the heights z_i, z_j, respectively) that project exactly to the crossing (x,y) in $\mathbb{R}^2$.

The chain of edges from v_{i+1} to v_j does not cross any other edges by our choice of the crossing in the move R1 and can be replaced by the single vertical edge from (x,y,z_i) to (x,y,z_j). This extra simplification can potentially make a few triangles on 3 consecutive vertices empty. Hence we can check if the simplifications from Stage 1 are possible for a few triangles related to the vertices $v_i, v_{i+1}, v_j, v_{j+1}$.

Stage 3: writing a Wirtinger presentation for the fundamental group. We remind how to write down a presentation of the group $\pi_1(\mathbb{R}^3 - K)$ by using a plane diagram D of a knotted graph $K \subset \mathbb{R}^3$, see more details in [2, section 6.1].

We arbitrarily orient all edge-paths of K, though our choice will not affect $\pi_1(\mathbb{R}^3 - K)$. We fix a base point $p \in \mathbb{R}^3$ at infinity, say at the point $(0,0,z)$ for a large coordinate $z > 0$. If we cut all essential vertices (of degree at least 3) and crossings (in lower edges), the diagram D splits into several oriented arcs $a_1, \dots, a_m$. In the 3rd picture of Fig. 6 these arcs in D contain the following vertices and crossings:

$$a_1 = [v_0, c_1, c_2], \quad a_2 = [c_2, v_1, v_2, c_3, c_1], \quad a_3 = [c_1, v_3, v_4, c_2, c_3], \quad a_4 = [c_3, v_5].$$

We associate to every resulting arc a_i a generator $x_i \in \pi_1(\mathbb{R}^3 - K)$. Each generator x_i can be represented by a closed loop $\tilde{x}_i$ that goes from the base point p to a point near the arc a_i along a path γ_i, makes a loop around the oriented arc a_i and then goes back to the base point p along γ_i in the opposite direction. In the 2nd and 3rd pictures of Fig. 6 we show each long loop $\tilde{x}_i$ only by a short arrow under a_i. Each short arrow is labelled by the generator $x_i \in \pi_1(\mathbb{R}^3 - K)$ represented by the loop $\tilde{x}_i$.

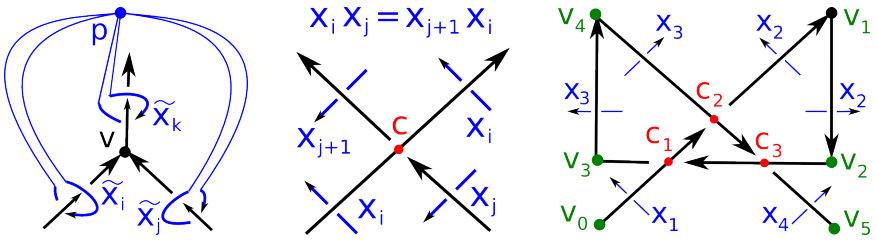

Fig. 6 Left: generators around a vertex and crossing. Right: 4 generators for 4 arcs in a diagram.

At each a crossing, two consecutive arcs a_j, a_{j+1} share the same endpoint c and another arc a_i crosses over c, see the 2nd picture of Fig. 6. To this crossing we associate the relation $x_i x_j x_i^{-1} = x_{j+1}$ saying that the loop $\tilde{x}_{j+1}$ around a_{j+1} can be obtained by going first along $\tilde{x}_i$, then along $\tilde{x}_j$ and along the reversed loop $\tilde{x}_i{}^{-1}$.

If we join two vertices of degree 1 away from the rest of the diagram, the initial and final generators are equal, e.g. $x_1 = x_4$ in the 3rd picture of Fig. 6. The crossings c_1, c_2, c_3 in the same picture have the associated relations $x_1 x_2 x_1^{-1} = x_3$ and $x_3^{-1} x_1 x_3 = x_2$ and $x_2 x_4 x_2^{-1} = x_3$, respectively. Together with $x_1 = x_4$, the 4 relations reduce to the short presentation $\langle x_1, x_2 \mid x_1 x_2 x_1 = x_2 x_1 x_2 \rangle$ of the trefoil group.

If a vertex v has attached arcs $a_1, \dots, a_l$, then write the relation $x_1^{\varepsilon_1} \dots x_l^{\varepsilon_l} = 1$, where $\varepsilon_i = +1$ for arcs a_i coming to v and $\varepsilon_i = -1$ for arcs a_i going out of v. The vertex v in the 1st picture of Fig. 6 has the associated relation $x_i x_j x_k^{-1} = 1$.

Any closed loop in the complement $\mathbb{R}^3 - K$ easily decomposes into a product of loops $\tilde{x}_i$ around arcs $a_1, \dots, a_m$. However, it is a non-trivial theorem that the simple relations above define the fundamental group $\pi_1(\mathbb{R}^3 - K)$, see [2, section 6.3].

We can convert a Gauss code of a plane diagram of K into a Wirtinger presentation of $\pi_1(\mathbb{R}^3 - K)$ as follows. The above arcs a_i between successive undercrossings in the plane diagram D correspond to *subsequences* between vertices and negative labels in the Gauss code of D. For each negative label $(-j)$, we know two subsequences that meet at $(-j)$ and also we can find the i-th subsequence containing the positive label j, so we can write the corresponding relation $x_i x_j x_i^{-1} = x_{j+1}$.

For each vertex v of deg ≥ 3, we can find subsequences in the code that start or finish with the symbol v and write the product of corresponding generators (if the subsequence starts with v) or their inverses (if the subsequence finishes with v).

Proof of Theorem 2. At Stage 1 of the KGG algorithm in Section 4, for each degree 2 vertex B of a polygonal graph $K \subset \mathbb{R}^3$, we compute the distance between the 2 neighbours A, C of B in total time $O(n)$, where n is the length of K. We sort all degree 2 vertices $B \in K$ by the increasing distances AC in time $O(n \log n)$.

Starting with a vertex B with a shortest segment AC, we check if the 3-chain ABC can be replaced by the single edge AC, which requires five 3-by-3 determinants for every other edge DE of K. If $\triangle ABC$ doesn't meet all edges DE, we remove B and update the sorted distances AC in time $O(\log n)$. The time of Stage 1 is $O(n^2)$.

At Stage 2 we check all pairwise intersections of $m \leq n$ projected edges in the simplified graph $K' \subset \mathbb{R}^3$, which requires $O(m^2)$ time. Stage 3 is linear in the length of a Gauss code which has $c = O(m^2)$ crossings. Hence the total time is $O(n^2)$. $\square$

5 Experimental results: recognizing knots in protein backbones

Table 2 shows how the numbers of vertices and crossings of a protein backbone K are reduced by Stage 1 of the KGG algorithm from Section 4. The knot types are 0 (unknot), 3_1 (trefoil knot), 4_1 (figure-eight knot) and 6_1 (Stevedore's knot).

The classical KMT algorithm for a polygonal chain of n edges has the running time $O(n^2)$. The time to compute the Alexander polynomial of a knotted graph with k crossings is $O(k^3)$, where $k = O(m^2)$ for a simplified graph of a length m.

Recall that the backbone of a protein is a polygonal chain of carbon atoms ordered as in a given PDB file. We linearly extend terminal edges of a backbone and join them away from all other vertices to get a closed knot. Table 2 shows the numbers of vertices and crossing after reductions by the KGG algorithm. In some cases the KTT algorithm outputs a few more crossings. because the Reidemeister move R1 wasn't used. In all cases the KGG algorithm is faster despite these extra moves.

The last 6 rows in Table 2 are for longest proteins from PDB. Even simplified backbones are too long and we hope to determine their knot types in the future.

The KGG algorithm can be extended to visualize knotted proteins using 3-page embeddings [11] and to compute abelian invariants of the Knotted Graph Group using GAP [3, section 47.15]. The C++ code is at author's website http://kurlin.org.

Table 2 Reduction in the number of vertices and crossings by the KTT and KGG algorithms

PDB code	original #vertices	original #crossings	reduced #vertices	reduced #crossings	knot type	KTT time in seconds	KGG time in seconds
1yrl	1875	1144	37	43	4_1	0.82	0.81
4n2x	1788	1033	81	211	6_1	1.05	1.01
1qmg	2049	1455	44	71	4_1	1.03	1.02
3wj8	1788	972	79	180	6_1	1.08	1.07
4d67	6548	5485	97	301	?	14.45	14.12
4uwa	13296	17288	99	391	?	59.79	57.05
4ujc	11938	10180	217	731	?	61.77	59.91
4uwe	13288	25449	114	686	?	61.48	61.05
4ujd	11938	10565	212	755	?	72.42	70.27
4ug0	11675	10073	206	617	?	81.59	78.23

Table 3 Knot types and Gauss codes of the reduced backbones of knotted proteins from PDB.

PDB code	original #crossings	knot type and Gauss code after KGG	PDB code	original #crossings	knot type and Gauss code after reduction by KGG
1v2x	39	$\mathbf{3}_1$ (1 -2 3 -1 2 -3)	3nou	304	$\mathbf{4}_1$ (-1 2 3 -4 5 1 -2 -5 4 -3)
3oil	102	$\mathbf{3}_1$ (1 -2 3 -1 2 -3)	3not	300	$\mathbf{4}_1$ (-1 2 3 -4 5 1 -2 -5 4 -3)

6 Conclusions, discussion and open problems for future work

We have designed a new easy-to-implement KGG algorithm in Section 4 to compute the Knotted Graph Group $\pi_1(\mathbb{R}^3 - K)$ for any polygonal graph $K \subset \mathbb{R}^3$ given by a sequence of points in $\mathbb{R}^3$. The experimental results in Section 5 confirm substantial reductions in the complexity of knotted backbones. Our approach strikes right in the middle of a wide range of topological objects. Namely, the KGG algorithm works for arbitrary knotted graphs, which are more general than knots or links and runs faster than memory expensive methods designed for regular 2D complexes [1].

A *theta-curve* is a knotted graph $\theta \subset \mathbb{R}^3$ with 2 vertices joined by 3 edges as in the Greek character θ. The enumeration of theta-curves with up to 7 crossings was manually completed [17] by analyzing the Alexander polynomial and 3 knots obtained from $\theta \subset \mathbb{R}^3$ by removing one of 3 edges. That is why we believe that abelian invariants of the quickly computable Knotted Graph Group $\pi_1(\mathbb{R}^3 - K)$ can be enough for enumerating more complicated theta-curves and general graphs.

Our robust computation of the fundamental group $\pi_1(\mathbb{R}^3 - K)$ for any knotted graph $K \subset \mathbb{R}^3$ opens the following new possibilities for further research.

- Automatically enumerate all isotopy classes of knotted graphs $K \subset \mathbb{R}^3$ with a few essential vertices and up to a maximum possible number of crossings. A good starting point is to check the manual classification of theta-curves in [17].
- Build distributions for isotopy classes of large random knots modelled as in [16], when the Alexander polynomial can be too weak to distinguish different knots.
- Study the persistence and stability of abelian invariants similarly to the persistence of the group $\pi_1(\mathbb{R}^3 - K)$ in a filtration of knot neighborhoods from [15].

Acknowledgements This work was done during the EPSRC-funded secondment at Microsoft Research Cambridge, UK. We thank all reviewers for valuable comments and helpful suggestions.

References

1. Brendel, P., Dlotko, P., Ellis, G., Juda, M., Mrozek, M.: Computing fundamental groups from point clouds. Appl. Algebra in Engineering, Communication, Computing, **26**, 27-48 (2015).
2. Crowell, R., Fox, R.: Introduction to Knot Theory. Grad. Texts Maths, **57**. Springer (1963).
3. Ellis, G.: HAP — Homological Algebra Programming package for GAP. Version 1.10.13 (2013). Available for download at http://www.gap-systems.org/Packages/hap.html.
4. Fenn, R.: Tackling the trefoils, J. Knot Theory Ramifications, **21**, 1240004 (2012).
5. Gordon, C., Luecke, J.: Knots are determined by their complements. J. Amer. Math. Soc., **2**, 371–415 (1989).
6. Jamroz, M., Niemyska, W., Rawdon, E., Stasiak, A., Millett, K., Sulkowski, P., Sulkowska, J.: KnotProt: a database of proteins with knots and slipknots. Nucleic Acids Res., **1**, 1–9 (2014).
7. Kauffman, L.: Invariants of graphs in three-space. Trans. AMS, **311**, 697–710 (1989).
8. Khatib, F., Weirauch, M., Rohl, C.: Rapid knot detection and application to protein structure prediction. Bioinformatics, **14**, 252–259 (2006).
9. Koniaris K., Muthukumar, M.: Self-entanglement in ring polymers. J. Chem. Phys. **95**, 2871–2881 (1991).
10. Kurlin, V., Smithers C.,: A linear time algorithm for embedding arbitrary knotted graphs into a 3-page book. To appear in the Springer book of the series CCIS: Communications in Computer and Information Science (2016).
11. Kurlin, V.: A linear time algorithm for visualizing knotted structures in 3 pages. Proceedings of IVAPP: Information Visualization Theory and Applications, p. 5–16 (2015), Berlin.
12. Kurlin, V.: Gauss paragraphs of classical links and a characterization of virtual link groups. Math. Proc. Cambridge Phil. Society, **145**, 129–140 (2008).
13. Kurlin, V.: Three-page encoding and complexity theory for spatial graphs. J Knot Theory Ramifications, **16**, 59–102 (2007).
14. Kurlin, V., Lines, D.: Peripherally specified homomorphs of link groups. J Knot Theory Ramifications, **16**, 719–740 (2007).
15. Letscher, D.: On Persistent Homotopy, Knotting and the Alexander Module. Proc. ITCS 2012.
16. Millett, K. C., Rawdon. E.J., Stasiak, A.: Linear random knots and their scaling behaviour. Macromolecules, **38**, 601–606 (2005).
17. Moriuchi, H.: An enumeration of theta-curves with up to 7 crossings. Proceedings of the East Asian School of Knots, Links and Related Topics, Seoul Korea (2004).
18. Taylor, W.: A deeply knotted protein structure and how it might fold. Nature, **406**, 916–919 (2000).
19. Waldhausen, F.: On irreducible 3-manifolds which are sufficiently large. Annals of Math. (2), **87**, 56–88 (1968).
20. Whitten, W.: Knot complements and groups. Topology, **26**, 41–44 (1987).