

COMPUTING WITH KNOT QUANDLES

GRAHAM ELLIS AND CÉDRIC FRAGNAUD

ABSTRACT. The number $Col_Q(K)$ of colourings of a knot K by a finite quandle Q has been used in the literature to distinguish between knot types. In this paper we suggest a refinement $Col_Q^F(K)$ to this knot invariant involving any computable functor F from finitely presented groups to finitely generated abelian groups. We are mainly interested in the functor $F = ab$ that sends each finitely presented group H to its abelianization $H^{ab} = H/[H.H]$. We describe algorithms needed for computing the refined invariant and illustrate implementations that have been made available as part of the HAP package for the GAP system for computational algebra. We use these implementations to investigate the performance of the refined invariant on prime knots with ≤ 11 crossings.

1. INTRODUCTION

For elements x, y in a group one can define the two operations $x * y = y^{-1}xy$ and $y \bar{*} x = yxy^{-1}$ which satisfy

- (Q1) $x * x = x$,
- (Q2) $(x * y) \bar{*} y = x = (x \bar{*} y) * y$,
- (Q3) $(x * y) * z = (x * z) * (y * z)$.

Recall that a *quandle* is a set Q equipped with two binary operations $*$ and $\bar{*}$ such that axioms Q1, Q2, Q3 hold for all $x, y, z \in Q$. The axioms imply that the right multiplication mapping $R_y: Q \rightarrow Q, x \mapsto x * y$ is bijective for each element y in a quandle. In particular, for each $y \in Q$ there exists a unique element \bar{y} such that $R_y(x * \bar{y}) = x$ and

$$x * \bar{y} = x \bar{*} y.$$

Thus a quandle can also be defined as a set Q with a single binary operation $*$ such that Q1 holds and such that the right multiplication mapping $R_y: Q \rightarrow Q, x \mapsto x * y$ is an automorphism (preserving the operation $*$) for all $y \in Q$.

Recall that a *knot* is a continuous embedding $K: \mathbb{S}^1 \hookrightarrow \mathbb{R}^3$ of the circle into Euclidean 3-space, and that two knots $K, K': \mathbb{S}^1 \hookrightarrow \mathbb{R}^3$ are *oriented equivalent* if there exists an orientation preserving homeomorphism $h: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ such that $K' = hK$. If a possibly non-orientation preserving homeomorphism h exists then the two knots are *unoriented equivalent*. A knot is *tame* if it is oriented equivalent to a piecewise linear polygonal curve. Henceforth we use the term *knot* to mean *tame knot*. A *knot type* is the oriented equivalence class of a knot. We write $K = K'$ when two knots K and K' belong to the same knot type, and we write $K \neq K'$ otherwise. Two knots K, K' can be added using the map $\iota: \mathbb{S}^1 \rightarrow \mathbb{S}^1 \vee \mathbb{S}^1$ which continuously wraps a circle once around a wedge of two circles. The sum is the composite map $K + K': \mathbb{S}^1 \xrightarrow{\iota} \mathbb{S}^1 \vee \mathbb{S}^1 \xrightarrow{K \vee K'} \mathbb{R}^3$ and induces a well-defined addition of knot types. A knot P is *prime* if it is not oriented equivalent to any sum $K + K'$ with $P \neq K$.

In Section 3 below we recall that for each knot K there is an associated knot quandle $Q(K)$ which is well-defined up to isomorphism. This is a *knot invariant*: oriented equivalent knots yield isomorphic quandles. In fact it is a *complete knot invariant*: if two knots K, K' have isomorphic associated quandles $Q(K) \cong Q(K')$ then the two knots are oriented equivalent. A proof of this was provided by Joyce [9] using a result of Waldhausen [14].

The knot quandle $Q(K)$ has infinite cardinality, and the description recalled in Section 3 is in terms of a finite number of quandle generators and relations. There are difficulties to deciding if two finitely presented knot quandles $Q(K)$ and $Q(K')$ are isomorphic. One approach to the

The second author was supported by an undergraduate summer internship at NUI Galway.

difficulties involves choosing a finite quandle Q and considering the finite set $\text{Hom}(Q(K), Q)$ of all quandle homomorphisms $c: Q(K) \rightarrow Q$. Such a homomorphism c is called a *colouring* of the knot K by the quandle Q . The cardinality $|\text{Hom}(Q(K), Q)|$ of this set is an invariant of the knot type of K , and is easy to compute given a finite presentation for $Q(K)$. Following [3] we denote this cardinality by $\text{Col}_Q(K)$. Computer calculations were used in [3] to establish the following result.

Theorem 1.1. [3] *There is a collection of 26 finite quandles Q_1, \dots, Q_{26} for which the invariants $\text{Col}_{Q_1}(K), \dots, \text{Col}_{Q_{26}}(K)$ together suffice to distinguish between all 2977 unoriented equivalence classes of prime knots on ≤ 12 crossings. The largest of these 26 quandles has cardinality 182.*

The integer $\text{Col}_Q(K)$ is one of the simplest invariants of a knot type that can be constructed from the finite set $\text{Hom}(Q(K), Q)$ arising from a finite quandle Q . Our aim in this paper is to advertize the performance of a refined invariant, which we denote $\text{Col}_Q^F(K)$, that can be derived from the set $\text{Hom}(Q(K), Q)$ and any functor F from the category of finitely presented groups to the category of finitely generated abelian groups. We are mainly interested in the functor $F = {}^{ab}$ that sends each finitely presented group H to its abelianization $H^{ab} = H/[H, H]$. We describe algorithms needed for computing the refined invariant and illustrate implementations that have been made available as part of the HAP package [4] for the GAP system for computational algebra [5]. We use these implementations to investigate the performance of the invariant on prime knots with ≤ 11 crossings.

The motivation for introducing the invariant $\text{Col}_Q^F(K)$ came from [1] where computations involving the knot group $G(K) = \pi_1(\mathbb{R}^3 \setminus K)$ were used to distinguish between prime knots. We recall in Section 3 that a finite presentation for the group $G(K)$ can be derived from a *knot diagram* (i.e. a projection of the image of K onto a 2-dimensional plane in \mathbb{R}^3 such that at most two distinct points on the knot project to any one point in the plane). Given a finite group G we say that a group homomorphism $c: G(K) \rightarrow G$ is a *colouring* of the knot K by the group G . The set $\text{Hom}(G(K), G)$ of such colourings is a knot invariant. In keeping with the above we set $\text{Col}_G(K) = |\text{Hom}(G(K), G)|$ and note that this integer is a readily computed knot invariant. Let us suppose that $G \leq S_n$ is a permutation group of degree $\leq n$. Each colouring c provides an action $(i, g) \mapsto i^g$ of the finitely presented infinite group $G(K)$ on the set $\{1, \dots, n\}$. Let $H_c = \{g \in G(K) : 1^g = 1\}$ be the subgroup of $G(K)$ that stabilizes 1. The coset space $G(K)/H_c$ has one coset for each integer in the orbit of 1 under the action of $G(K)$. In particular, H_c is a finite index subgroup of $G(K)$. Using a version of the Reidemeister-Schreier procedure we can find a finite presentation of H_c . From this presentation we compute the list (d_1, \dots, d_k) of abelian invariants of the abelian group $F(H_c) = \mathbb{Z}_{d_1} \oplus \dots \oplus \mathbb{Z}_{d_k}$. Here d_i divides d_{i+1} and \mathbb{Z}_{d_i} is infinite cyclic when $d_i = 0$. Let F_c denote the integer vector $F_c = (d_1, \dots, d_k, |G(K) : H_c|)$ where $|G(K) : H_c|$ is the index of H_c in $G(K)$. The set of non-negative integer vectors

$$(1) \quad \text{Col}_G^F(K) = \{F_c : c \in \text{Hom}(G(K), G)\}$$

is an invariant of the knot type of K . Computer calculations were used in the preprint [1] to establish a result which can be reformulated in the language of group colourings as follows.

Theorem 1.2. [1] *For $G = S_7$ and the abelianization functor $F(H) = H^{ab} = H/[H, H]$, the invariant $\text{Col}_G^F(K) = \text{Col}_{S_7}^{ab}(K)$ suffices to distinguish between all 59937 unoriented equivalence classes of prime knots on ≤ 14 crossings.*

The paper is structured as follows. In Section 2 we describe background theory and a GAP implementation for enumerating finite quandles and finite connected quandles, and for computing automorphism groups of finite (connected) quandles. In Section 3 we describe background theory and an implementation for computing with knot quandles and knot groups. In Section 4 we transfer the group-theoretic definition (1) into the setting of quandles, and investigate the performance of both the group-theoretic and quandle-theoretic invariants $\text{Col}_G^F(K)$ and $\text{Col}_Q^F(K)$ on prime knots with ≤ 11 crossings. If one wishes to distinguish between prime knots on just ≤ 11 crossings then several of the 26 invariants $\text{Col}_{Q_i}(K)$ in Theorem 1.1 can be omitted. However, by reproducing the computations underlying Theorem 1.1 one can show that if the invariant $\text{Col}_{Q_{26}}(K)$ corresponding to the quandle Q_{26} of order 182 is dropped from the list of 26 invariants then the remaining 25

invariants do not distinguish between all prime knots on ≤ 11 crossings. In Theorem 1.2 the group $G = S_7$ can be replaced by the group $G = S_6$ of order 720 if one wishes to distinguish between prime knots on just ≤ 11 crossings. In Section 4 we establish the following new result involving only quandles of order ≤ 24 . (See Proposition 4.1 for a more detailed version.)

Theorem 1.3. *There is a list of 26 connected quandles P_1, \dots, P_{26} of orders $|P_i| \leq 24$ such that the knot invariants $Col_{P_1}^{ab}(K), \dots, Col_{P_{26}}^{ab}(K)$ together distinguish between all 801 unoriented equivalence classes of prime knots on ≤ 11 crossings.*

Acknowledgements. In addition to relying on the core GAP system, our HAP code for quandles relies on work of some others. Our function for testing isomorphism of connected quandles is taken almost directly from code in [8] which in turn is based on code in the `Loops` package [11] for GAP. Our enumeration of connected quandles of orders $31 \leq n \leq 47$ is taken directly from the `Rig` package [12]. Our function for constructing pure cubical prime knots was constructed using arc presentations for knots on ≤ 11 available in the Knot Atlas [10] which we accessed via the KnotInfo website [2]. Our function for producing supplemented Gauss codes for prime knots on ≤ 12 crossings was constructed using their planar diagram representations available in the Knot Atlas [10].

2. FINITE QUANDLES

A *magma* is a set with a binary operation $*$. We regard a quandle Q as a magma for which $x * x = x$ (all $x \in Q$) and right multiplication by y is an automorphism (all $y \in Q$). We use the `IsMagma` data type in the GAP system [5] for computational algebra to encode quandles. Our functions for quandles are distributed with GAP as part of the HAP package [4].

Example 2.1. For any group G and any integer $n \geq 1$ there is a quandle operation on G defined by $x * y = y^{-n}xy^n$ for $x, y \in G$. The following commands construct this quandle for G equal to the symmetric group S_6 and $n = 1$.

```
gap> G:=SymmetricGroup(6);
Sym( [ 1 .. 6 ] )
gap> Q:=ConjugationQuandle(G,1);
<magma with 15 generators>
gap> Size(Q);
720
```

The commands establish that the quandle is generated by 15 elements and has order 720. ■

Example 2.2. Let G be a group with subgroup $H < G$ and with element $\ell \in G$ such that $\ell^{-1}h\ell = h$ for all $h \in H$. Let $Q = G/H = \{Hx : x \in G\}$ denote the set of right cosets. There is a quandle operation on Q given by $(Hx) * (Hy) = H\ell^{-1}xy^{-1}\ell y$. The following commands construct this quandle for the finitely presented group $G = \langle a, b \mid aba = bab \rangle$, subgroup $H = \langle a, b^{-1}a^2b^{-1}, b^{-1}a^{-2}b^{-1} \rangle$ and $\ell = a$.

```
gap> F:=FreeGroup("a","b");;a:=F.1;;b:=F.2;;
gap> G:=F/[a*b*a*(b*a*b)^-1];;a:=G.1;;b:=G.2;;
gap> H:=Group([ a, b^-1*a^2*b^-1, b^-1*a^-2*b^-1 ]);;
gap> Q:=CosetsQuandle(G,H,a);
<magma with 2 generators>
gap> Display(MultiplicationTable(Q));
[ [ 1, 5, 4, 2, 3, 1 ],
  [ 4, 2, 2, 6, 1, 5 ],
  [ 5, 3, 3, 1, 6, 4 ],
  [ 3, 1, 6, 4, 4, 2 ],
  [ 2, 6, 1, 5, 5, 3 ],
  [ 6, 4, 5, 3, 2, 6 ] ]
```

In this example the quandle Q is generated by two elements; it has order 6 and the final command displays its multiplication table. ■

One approach to enumerating finite quandles of order n is to run through all $n \times n$ multiplication tables in which the i th column is a permutation of the integers $\{1, \dots, n\}$ that fixes i , and to select those tables that yield a magma satisfying the quandle axioms. The process will yield a collection T_n of all multiplication tables of quandles of order n with some pairs of tables in T_n yielding isomorphic quandles. To produce the collection T_n one does not need to check all $((n-1)!)^n$ multiplication tables: if some entries in the first k columns of a multiplication table violate the quandle axioms then there is no need to check the remaining multiplication tables with those same first k columns. Given any quandle Q with underlying set $\{1, \dots, n\}$ and permutation $\sigma \in S_n$ we can create an isomorphic quandle Q^σ on the same underlying set, for which the isomorphism $Q \xrightarrow{\cong} Q^\sigma$ is given by $i \mapsto \sigma(i)$. We can choose some total ordering on the collection T_n and let Q^{\min} denote the minimum of the quandles Q^σ for $\sigma \in S_n$. Then $\{Q \in T_n : Q = Q^{\min}\}$ is an irredundant set of representatives for the isomorphism classes of quandles of order n . This strategy was used in [7] to obtain a list of all quandles of order ≤ 5 . A subsequent paper [6] describes an efficient *C* implementation of the strategy capable of listing all quandles of order ≤ 8 , the quandle multiplication tables being made available at an accompanying URL. We have encoded the strategy in HAP for quandles of size ≤ 6 .

Example 2.3. The following commands use the above strategy to establish that there are precisely 73 quandles of order 6.

```
gap> S:=Quandles(6);;
gap> Length(S);
73
```

■

The automorphism group $Aut(Q)$ of a quandle Q of order n is isomorphic to a subgroup of the symmetric group S_n . For small values of n one can compute $Aut(Q)$ directly from this statement: simply check to see which elements of S_n induce an automorphism of Q . For each element $y \in Q$ we refer to the automorphism $R_y: Q \rightarrow Q, x \mapsto x * y$ as a *right multiplication*. The subgroup $Inn(Q)$ of $Aut(Q)$ generated by all right multiplications is called the *inner automorphism group*. Note that a composite of right multiplications is not necessarily a right multiplication; we use the term *inner automorphism* to refer to a composite of right multiplications.

A quandle Q is said to be *homogeneous* if the action of $Aut(Q)$ on Q has precisely one orbit. We say that Q is *connected* if the action of $Inn(Q)$ on Q has precisely one orbit. We say that Q is *Latin* if the multiplication table of Q is a Latin square. Clearly Latin \Rightarrow connected \Rightarrow homogeneous.

Example 2.4. The following commands establish that there are precisely two homogeneous quandles of order 6, both of which are connected.

```
gap> S:=Quandles(6);;
gap> Collected(List(S,IsHomogeneous));
[ [ true, 2 ], [ false, 71 ] ]
gap> Collected(List(S,IsConnected));
[ [ true, 2 ], [ false, 71 ] ]
```

■

In Section 3 we consider infinite quandles $Q(K)$ associated knots. It is known that these quandles are connected. Thus, for any homomorphism $c: Q(K) \rightarrow Q$ to a finite quandle Q the image $\text{im}(c)$ is a connected quandle. This is one reason for focussing attention on connected finite quandles. Another reason is that the richer theory implied by the connectivity condition helps with computations.

An enumeration of connected quandles of order ≤ 35 was provided in [13] and the multiplication tables of these quandles is available in the GAP package Rig [12]. The basic idea of [13] is to derive the enumeration of connected quandles of order n from the existing enumeration of transitive permutation groups of degree n . Some algebraic refinements to the method were described in [8] and used to enumerate all connected quandles of order ≤ 47 . To work with these connected quandles it suffices to have access to their multiplication tables, which we provide with HAP. However, there are some advantages to also providing functions for reconstructing the enumeration

from the existing enumeration of transitive groups available in `GAP`: such functions can be used to construct connected quandles of order $n \geq 48$ given appropriate transitive permutation groups of degree n ; such functions can be used to compute properties of connected quandles such as their automorphism groups. Currently the default distribution of `GAP` contains all transitive groups of degree ≤ 30 . Files containing transitive groups of higher degree are available on the internet (see [8] for details). (We mention that the `Rig` package is incompatible with various core `GAP` packages, due to a significant number of clashes of reserved words, and can not be loaded with these core packages. A motivation for overlapping `HAP`'s functionality in the area of connected quandles with that of the `Rig` package is that this overlap allows users to investigate connected quandles with the full functionality of the `GAP` system.)

Before recalling the algebraic details underlying the enumeration of connected quandles we first provide an example.

Example 2.5. The following commands construct the conjugation quandle $(Q, *)$ on the set of permutations S_5 with operation $x * y = y^{-1}xy$ and then construct the subquandles of Q arising as the ‘components’ of Q . These *components* are the orbits under the action of the inner automorphism group $\text{Inn}(Q)$. Note that a component is a quandle but is not necessarily a connected quandle. There are seven components P_1, \dots, P_7 , of which six are connected. The largest component P_5 has order 30, and the penultimate command identifies this as quandle number 17 arising in our enumeration of connected quandles of order 17. The final command involves enumerating the list of connected quandles of order 30 and then identifying P_5 in this list. Once this list is computed it is automatically stored and never recomputed during the `GAP` session. The final command shows that $\text{Aut}(P_5)$ is isomorphic to the symmetric group S_5 .

```
gap> Q:=ConjugationQuandle(SymmetricGroup(5),1);
<magma with 10 generators>
gap> C:=ConnectedComponentsQuandle(Q);;
gap> List(C,Size);
[ 1, 10, 20, 15, 30, 20, 24 ]
gap> List(C,IsConnected);
[ true, true, true, true, true, true, false ]
gap> P5:=C[5];
<magma with 3 generators>
gap> IdConnectedQuandle(P5);
[ 30, 17 ]
```

The following command can be used to import connected quandles directly from a data base; for quandles of order $31 \leq n \leq 47$ the data base is taken from the `Rig` package.

```
gap> Q:=ConnectedQuandle(47,1,"import");
This function uses data from the Rig package for racks, quandles and Nichols
algebras due to L. Vendramin.
<magma with 2 generators>
```

■

We briefly recall from [8] the theory underlying the enumeration of connected quandles. Suppose that we start with a connected quandle Q of order n . The inner automorphism group $G = \text{Inn}(Q)$ acts transitively on the set Q . Choose some element $e \in Q$ and let G_e denote the stabilizer of e in G . Since Q is connected the isomorphism type of G_e does not depend on the choice of e . The right multiplication map $R_e: Q \rightarrow Q, x \mapsto x * e$ lies in G_e . For $x \in Q$ we write x^{R_e} rather than $r_e(x) = x * e$. For any $g \in G_e$ and any $x \in Q$ we have $x^{gR_e} = x^g * e = x^g * e^g = (x * e)^g = x^{R_e g}$ and hence R_e lies in the centre $Z(G_e)$ of G_e . Since Q is connected it also follows that the element R_e normally generates G . The quadruple (G, Q, e, R_e) is an example of the following notion.

Definition 2.1. [8] A *quandle envelope* is a quadruple (G, Q, e, ρ) where G is a group that acts transitively on a set Q with distinguished element $e \in Q$ and where ρ is an element of the centre $Z(G_e)$ of the stabilizer group of e ; the element ρ is required to normally generate G .

The construction of a quandle envelope from a connected quandle is explained above. Conversely, suppose that (G, Q, e, ρ) is a quandle envelope. Since G acts transitively on Q we can find,

for each $y \in Q$, an element $\hat{y} \in G$ such that $e^{\hat{y}} = y$. For $x, y \in Q$ we define

$$x * y = x^{\rho^{\hat{y}}}.$$

One can check that under this operation $*$ the set Q becomes a quandle. This leads to the following result, a detailed proof of which is given in [8].

Theorem 2.1. [8] *The following data are equivalent:*

1. A connected quandle Q with preferred element $e \in Q$.
2. A quandle envelope (G, Q, e, ρ) .

It is shown in [8] that for any quandle envelope the derived subgroup G' acts transitively on Q and the quotient G/G' is cyclic. Thus, given an enumeration of transitive permutation groups of degree n , we can select those groups G for which G' acts transitively on $Q = \{1, \dots, n\}$ and for which the quotient G/G' is cyclic. For each selected group we can construct all $\rho \in Z(G_1)$ for which $(G, Q, 1, \rho)$ is a quandle envelope. We can then construct the corresponding connected quandles. In this way the enumeration of transitive groups of degree n can be used to provide a list of all connected quandles of order n ; the list will contain pairs of isomorphic quandles. We can use a (slightly refined version of) the quandle isomorphism test described above to remove isomorphic copies.

Example 2.6. The following commands illustrate how one can construct a connected quandle Q of order 5 from a transitive permutation group G of degree 5.

```
gap> Q:=[1..5];;
gap> G:=Group([ (2,5)(3,4), (1,4,2,5,3) ]);;
gap> z:=Random(Stabilizer(G,1));
(2,5)(3,4)
gap> IsQuandleEnvelope(Q,G,1,z);
true
gap> Q:=QuandleQuandleEnvelope(Q,G,1,z);
<magma with 5 generators>
```

■

To compute the automorphism group of a connected quandle Q we follow [8] in considering, for any group G , subgroup $H \leq G$ and element $x \in G$, the subgroup

$$Aut(G)_{x,H} = \{\alpha \in Aut(G) : \alpha(x) = x, \alpha(H) = H\} \leq Aut(G)$$

of the automorphism group of G . The following is proved in [8].

Proposition 2.2. [8] *Let Q be a connected quandle. Choose an element $e \in Q$ and let $G = Inn(Q)$ be the right multiplication group of Q . Then there is a group isomorphism*

$$Aut(Q) \cong G \rtimes Aut(G)_{R_e, G_e} / \{(g, \iota_g) : \alpha \in G_e\}$$

where $\iota_g : G \rightarrow G, x \mapsto gxg^{-1}$.

Example 2.7. The following commands use Proposition 2.2 to show that, for the quandle P_5 of Example 2.5, the automorphism group $Aut(P_5)$ is isomorphic to the symmetric group S_5 .

```
gap> P5:=ConnectedQuandle(30,17,"import");;
gap> A:=AutomorphismGroupQuandleAsPerm(P5);;
gap> StructureDescription(A);
"S5"
```

The above commands construct a permutation group A isomorphic to the automorphism group $Aut(P_5)$. One can alternatively construct a version of $Aut(P_5)$ whose action on P_5 is implemented.

```
gap> P5:=ConnectedQuandle(30,17,"import");;
gap> Aut:=AutomorphismGroupQuandle(P5);;
gap> a:=Random(Aut);;
gap> x:=Random(P5);
m12
gap> x^a;
m28
```

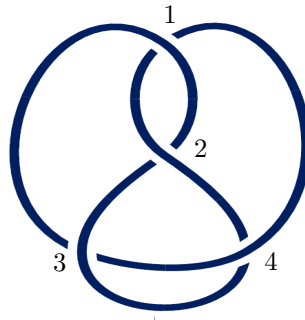


FIGURE 1. A 4-crossing knot diagram

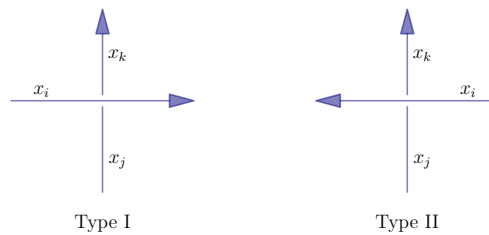


FIGURE 2. The two types of knot crossing.

Some further functionality for finite connected quandles, relating to their fundamental group and second homology, has been implemented in HAP. Details are available at the package website [4].

3. KNOT QUANDLES AND GROUPS

A knot type can be represented as a knot diagram such as that of Figure 1. A knot diagram can be represented on a computer by the sequence of integers, known as a *Gauss code*, constructed as follows. The crossings in the knot diagram are first numbered in any fashion and the knot is oriented by assigning a direction to it. Then, starting at any point on the knot we travel once around the knot listing the crossings that are met in the order that they are met. The integers in the list are given signs: a sign of -1 is used when a crossing is met as an over crossing; a sign of $+1$ is used when a crossing is met as an under crossing. A Gauss code for the knot diagram of Figure 1 is

$$-1, 2, -3, 4, -2, 1, -4, 3.$$

The Gauss code does not contain enough information to reconstruct the knot type: it does not allow one to distinguish between the two types of crossing shown in Figure 2. To overcome this difficulty we supplement the Gauss code with a list of signs, one for each crossing. A $+1$ is used to record a crossing of Type I and a -1 is used for a crossing of Type II. The list of signs for the knot of Figure 1 is

$$1, 1, -1, -1.$$

Example 3.1. The following command constructs the supplemented Gauss code for the prime knot on four crossings. Up to mirror image there is only one prime knot type on four crossings, namely that of Figure 1.

```
gap> GaussCodeKnot(4,1);
[[ [-1, 2, -3, 4, -2, 1, -4, 3 ]], [ 1, 1, -1, -1 ] ]
```

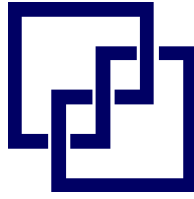


FIGURE 3. Knot diagram for the trefoil knot.

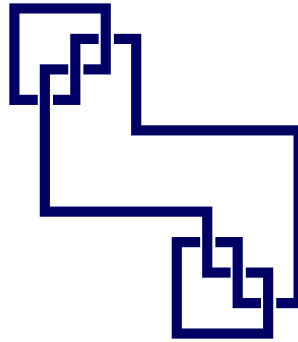


FIGURE 4. Addition of a trefoil knot with its mirror image.

To display knots on a computer it is convenient to consider diagrams such as that of Figure 3. This diagram has several special properties: (i) it is piecewise linear with each linear segment either vertical or horizontal; (ii) no two vertical segments share a common x -coordinate and no two horizontal segments share a common y -coordinate; (iii) all over-crossing points belong to vertical segments and all under-crossing points belong to horizontal segments. A knot diagram satisfying properties (i)-(iii) is called an *arc presentation* of the knot. Any arc presentation can be represented as a list $(m_1, n_1), (m_2, n_2), \dots, (m_k, n_k)$ of pairs of integers where the bottom horizontal segment extends from the m_1 th to the n_1 th vertical segment; the second to bottom horizontal segment extends from the m_2 th vertical segment to the n_2 th vertical segment, and so on. For example, the list $(2, 5), (1, 3), (2, 4), (3, 5), (1, 4)$ specifies the arc presentation of Figure 3. It is particularly easy to implement knot addition and reflection of knots for arc presentations.

Example 3.2. The following commands construct the sum of the trefoil knot and its mirror image, and produce the display of this sum shown in Figure 4. The final command provides the supplemented Gauss code for the sum.

```
gap> K:=PureCubicalKnot(3,1);;
gap> S:=KnotSum(K, ReflectedCubicalKnot(K));;
gap> ViewPureCubicalKnot(S);

gap> GaussCodeOfPureCubicalKnot(S);
[ [ [ 3, -2, 1, 4, -5, 6, -4, 5, -6, -3, 2, -1 ] ], [ -1, -1, -1, 1, 1, 1 ] ]
```

■

To each knot K is associated its *knot group* $G(K)$. Recall that this is a finitely presented group. Given any knot diagram for K with n crossings, the *Wirtinger presentation* for $G(K)$ involves n generators and n relators. The n undercrossings divide the knot diagram into n connected arcs and these arcs correspond to the generators x_i ($1 \leq i \leq n$). To specify the relators we choose one of the two possible directions of travelling around the knot. Each crossing of type I (see Figure 2)

contributes a relator of the form

$$x_i^{-1} x_k^{-1} x_i x_j \quad (\text{Type I})$$

to the presentation and each relator of type II contributes a relator of the form

$$x_i x_k^{-1} x_i^{-1} x_j. \quad (\text{Type II})$$

The isomorphism class of $G(K)$ depends only on the knot type of K . The Wirtinger presentation for $G(K)$ can of course be simplified.

Example 3.3. The following command produces the finite presentation $G(K) \cong \langle x, y \mid xyx = yxy \rangle$ for the knot group of the trefoil knot.

```
gap> W:=WirtingerGroup(GaussCodeKnot(3,1));;
gap> W:=SimplifiedFpGroup(W);
<fp group of size infinity on the generators [ f1, f2 ]>
gap> RelatorsOfFpGroup(W);
[ f1^-1*f2^-1*f1^-1*f2*f1*f2 ]
```

■

The knot group $G(K)$ is an invariant of the knot type of K . However, it can be a challenge to decide if two finitely presented knot groups $G(K)$ and $G(K')$ are isomorphic or not. One approach is to consider easily comparable isomorphism invariants of the knot group. A classic invariant of $G(K)$ is defined (see Definition 3.1) in terms of the chain complex $C_*(X) \otimes_{\mathbb{Z}G'} \mathbb{Q}$ of rational vector spaces, where: X is the 2-dimensional reduced CW complex associated to any finite presentation $\langle \underline{x} \mid \underline{r} \rangle$ for $G(K)$; $C_*(X)$ denotes the cellular chain complex of X ; $G' = [G(K), G(K)]$ is the commutator subgroup of $G(K)$. The chain complex $C_*(X) \otimes_{\mathbb{Z}G'} \mathbb{Q}$ is of the form

$$(2) \quad \bigoplus_{(t,r) \in T \times \underline{r}} \mathbb{Q} \longrightarrow \bigoplus_{(t,x) \in T \times \underline{x}} \mathbb{Q} \longrightarrow \bigoplus_{t \in T} \mathbb{Q}$$

where T a set of representatives for the elements of the infinite cyclic group $G(K)/G'$. Using the bijections $T \cong C_\infty = \langle t \rangle$ one can also view the chain complex (2) as a complex of free $\mathbb{Q}(t)$ -modules

$$(3) \quad \bigoplus_{r \in \underline{r}} \mathbb{Q}(t) \xrightarrow{\partial_2} \bigoplus_{x \in \underline{x}} \mathbb{Q}(t) \longrightarrow \mathbb{Q}(t)$$

over the ring $\mathbb{Q}(t)$ of Laurent polynomials. Since $\mathbb{Q}(t)$ is a principal ideal domain one could use Smith Normal Form to compute the first homology $H_1(C_*(X) \otimes_{\mathbb{Z}G'} \mathbb{Q})$ which is an invariant of G' and hence of $G(K)$. Alternatively, one can compute the invariant of Definition 3.1 defined in terms of the $m \times n$ matrix A over $\mathbb{Q}(t)$ representing the boundary homomorphism ∂_2 in (3). Here m is the number of generators in our presentation for $G(K)$ and n is the number of relators in the presentation.

Definition 3.1. The polynomial

$$p_K(t) = \det_{n-1}(A) / \det_{n-2}(A),$$

where $\det_i(A)$ is the greatest common divisor of all $i \times i$ minors of A , is known as the *Alexander polynomial* of the knot K and is an invariant of the knot group $G(K)$. Note that greatest common divisors, and hence Alexander polynomials $p_K(t)$, are only uniquely determined up to multiplication by powers of t .

The HAP implementation of the Alexander polynomial always returns a polynomial with one constant term and no negative powers of t .

Example 3.4. The following commands use a direct implementation of Definition 3.1 to compute the Alexander polynomial

$$p_K(t) = t^8 - 7t^7 + 23t^6 - 46t^5 + 59t^4 - 46t^3 + 23t^2 - 7t + 1$$

of the first knot K in the list of prime knots on 12 crossings stored in HAP.

```
gap> W:=WirtingerGroup(GaussCodeKnot(12,1));;
gap> W:=SimplifiedFpGroup(W);;
```

```
gap> p:=AlexanderPolynomial(W);
x_1^8-7*x_1^7+23*x_1^6-46*x_1^5+59*x_1^4-46*x_1^3+23*x_1^2-7*x_1+1
```

■

Another readily computed and easily compared invariant of a finitely presented knot group is the number $Col_G(K) = |Hom(G(K), G)|$ of group homomorphisms $c: G(K) \rightarrow G$ to some fixed finite group G . Given a presentation $G(K) = \langle x_1, \dots, x_m \mid r_1, \dots, r_n \rangle$ with associated free group $F = F(x_1, \dots, x_m)$, such homomorphisms correspond to those m -tuples $(f(x_1), \dots, f(x_m)) \in G^m$ that induce homomorphisms $F \rightarrow G$ mapping each relator $r_i \in F$ to the identity element of G . To efficiently compute $Col_G(K)$ from this correspondence one should first attempt to simplify the presentation of $G(K)$ with the aim of reducing the number m of generators as much as possible. The computation of $Col_G(K)$ can be further be aided by involving the automorphism group $Aut(G)$. The action of $Aut(G)$ on G induces an action of $Aut(G)$ on $Hom(G(K), G)$. It suffices to compute the number of orbits of this action on $Hom(G(K), G)$, together with the order of the stabilizer subgroups of orbit representatives. The orbits and stabilizers of the action on $Hom(G(K), G)$ can be derived from the orbits and stabilizers of the induced action of $Aut(G)$ on G^m . The latter can be calculated by inductively computing the orbits and stabilizers of the action of $Aut(G)$ on G^i for $i = 1, \dots, m$. The current implementation of $Col_G(K)$ in HAP adopts this approach, and seems to be practical for presentations of $G(K)$ involving a small number of generators m and groups G of small order.

Example 3.5. There are 801 unoriented equivalence classes of prime knots represented by knot diagrams with ≤ 11 crossings. (Any knot is unoriented equivalent to its mirror image.) The following commands partition these 801 knots into classes, a class consisting of those knots K with a common value of $Col_G(K)$ for $G = S_5$ the symmetric group of degree 5. The final command records the number of classes and the sizes of classes. For instance, there are 51 classes of cardinality 1 and 18 classes of cardinality 2.

```
gap> L:=[];
gap> for n in [3..11] do
> for i in [1..NumberOfPrimeKnots(n)] do
> Add(L,SimplifiedFpGroup(WirtingerGroup(GaussCodeKnot(n,i))));
> od;od;
gap> C:=Classify(L,K->NumberOfHomomorphisms(K,SymmetricGroup(5)));
gap> Collected(List(C,Size));
[ [ 1, 51 ], [ 2, 18 ], [ 3, 5 ], [ 4, 1 ], [ 5, 8 ], [ 6, 4 ], [ 7, 3 ],
  [ 8, 7 ], [ 9, 4 ], [ 11, 1 ], [ 12, 1 ], [ 13, 2 ], [ 14, 2 ], [ 16, 2 ],
  [ 17, 1 ], [ 21, 1 ], [ 26, 1 ], [ 29, 2 ], [ 34, 1 ], [ 51, 1 ],
  [ 62, 1 ], [ 66, 1 ], [ 74, 1 ] ]
```

The following further commands partition the 801 prime knots into equivalence classes where a class now consists of knots K with a common Alexander polynomial.

```
gap> C:=Classify(L,K->AlexanderPolynomial(K));
gap> Collected(List(C,Size));
[ [ 1, 383 ], [ 2, 111 ], [ 3, 40 ], [ 4, 10 ], [ 5, 6 ], [ 6, 1 ] ]
```

This shows that the Alexander polynomial is better than $Col_{S_5}(K)$ for distinguishing between prime knots on ≤ 11 crossings. ■

Example 3.5 is summarized in the following result.

Proposition 3.1. (i) For the symmetric group $G = S_5$ the invariant $Col_G(K)$ partitions the 801 prime knots on ≤ 11 crossings (omitting mirror images) into 119 classes, 51 of which have cardinality 1 and all of which have cardinality ≤ 74 .

(ii) The Alexander polynomial partitions the 801 knots into 551 classes, 383 of which have cardinality 1 and all of which have cardinality ≤ 6 .

To each knot K one can associate the *knot quandle* $Q(K)$. This is a finitely presented quandle introduced by Joyce in [9]. Given any knot diagram for K with n crossings, Joyce's presentation for $Q(K)$ involves n generators and n relations. The n undercrossings divide the knot diagram

into n connected arcs and these arcs correspond to the quandle generators x_i ($1 \leq i \leq n$). To specify the relations we choose one of the two possible directions of travelling around the knot. Each crossing of type I (see Figure 2) contributes a quandle relation of the form

$$x_j = x_k * x_i \quad (\text{Type I})$$

to the presentation and each crossing of type II contributes a relation of the form

$$x_j = x_k \bar{*} x_i . \quad (\text{Type II})$$

In the second type of relation we write $x \bar{*} y$ as shorthand for $x * \bar{y}$ where $R_y(x * \bar{y}) = x$. The isomorphism class of this finitely presented quandle $Q(K)$ depends only on the knot type of K . Joyce's presentation for $Q(K)$ can of course be simplified.

As with knot groups, it can be a challenge to decide if two finitely presented knot quandles $Q(K)$ and $Q(K')$ are isomorphic or not. A readily computed and easily compared invariant of a finitely presented knot quandle is the number $Col_Q(K) = |Hom(Q(K), Q)|$ of quandle homomorphisms $c: Q(K) \rightarrow Q$ to some fixed finite quandle Q . For an efficient computation of $Col_Q(K)$ we can proceed as in the group theoretic case: first try to simplify the presentation of $Q(K)$ with a view to reducing the number of generators, and then inductively invoke the action of $Aut(Q)$ on Q^m .

Example 3.6. Let K and L denote the trefoil knot and its mirror image. The sum $K + L$ is known as the *square knot*. The sum $K + K$ is known as the *granny knot*. The granny and square knots are well-known to have isomorphic knot groups, $G(K + K) \cong G(K + L)$. The following commands use the quandle invariant $Col_Q(K)$ to establish that the granny knot is not oriented equivalent to the square knot. We take Q to be the seventeenth quandle in our enumeration of connected quandles of order 24.

```
gap> Q:=ConnectedQuandle(24,17,"import");;
gap> K:=PureCubicalKnot(3,1);;
gap> L:=ReflectedCubicalKnot(K);;
gap> square:=KnotSum(K,L);;
gap> granny:=KnotSum(K,K);;
gap> gcsquare:=GaussCodeOfPureCubicalKnot(square);;
gap> gcgranny:=GaussCodeOfPureCubicalKnot(granny);;
gap> Qsquare:=PresentationKnotQuandle(gcsquare);;
gap> Qgranny:=PresentationKnotQuandle(gcgranny);;
gap> NumberOfHomomorphisms(Qsquare,Q);
408
gap> NumberOfHomomorphisms(Qgranny,Q);
24
```

■

4. THE REFINED INVARIANT

Let $G \leq S_n$ is a permutation group of degree $\leq n$. Each homomorphism $c: G(K) \rightarrow G$ provides an action $(i, g) \mapsto i^g$ of the finitely presented infinite group $G(K)$ on the set $\{1, \dots, n\}$. Let $H_c = \{g \in G(K) : 1^g = 1\}$ be the subgroup of $G(K)$ that stabilizes 1. The coset space $G(K)/H_c$ has one coset for each integer in the orbit $1^{G(K)} = \{1^g : g \in \text{Im}(c)\}$.

Using standard Cayley graph algorithms one can compute a finite generating set \underline{k} for the kernel $\ker(c)$ of the homomorphism c . One can also readily compute a preferred pre-image representative \tilde{y} for each $y \in \text{im}(f) \leq G$. The finite set $\underline{w} = \underline{k} \cup \{\tilde{y} : y \in \text{im}(f)\}$ generates the subgroup $H_c \leq G(K)$. The Reidemeister-Schreier procedure can be applied to the generating set \underline{w} and a finite presentation for $G(K)$ to produce a finite presentation for the subgroup H_c . Given any computable functor F from finitely presented groups to finitely generated abelian groups we can compute the list (d_1, \dots, d_k) of abelian invariants of the abelian group $F(H_c) = \mathbb{Z}_{d_1} \oplus \dots \oplus \mathbb{Z}_{d_k}$. Here d_i divides d_{i+1} and $d_i = 0$ for an infinite summand. Let $|G(K) : H_c|$ denote the index of H_c in $G(K)$ and set $F_c = (d_1, \dots, d_k, |G(K) : H_c|)$. We can thus compute, as an ordered list of integer vectors,

$$(4) \quad Col_G^F(K) = \{F_c : c \in Hom(G(K), G)\}.$$

The following are three possibilities for the functor F .

- (1) the constant functor $F(H) = 0$ which assigns the trivial group 0 to each finitely presented group H .
- (2) the abelianization functor $F(H) = H^{ab} = H/[H, H]$.
- (3) the direct sum $F(H) = \gamma_c \oplus (\gamma_{c-1}H/\gamma_{c-2}H \oplus \cdots \oplus H/\gamma_2H$ where $\gamma_2H = [H, H]$ and $\gamma_{i+1}H = [\gamma_iH, H]$.

We shall focus on (2) since experimentation on knots with low crossing number suggests that (2) yields a significantly stronger invariant than (1) and as strong an invariant as (3).

From a computational viewpoint it is useful to note that for any homomorphism $c: G(K) \rightarrow G$ and any $g \in G$ one has a conjugate homomorphism $c^g: G(K) \rightarrow G, x \mapsto g^{-1}c(x)g$, and that the corresponding integer vectors F_c and F_{c^g} are identical. Let us define c and c^g to be *conjugate homomorphisms* and set $[G(K), G]$ equal to a set of representatives of the conjugacy classes in $\text{Hom}(G(K), G)$. The following invariant contains the same information as in (4) but is slightly quicker to compute.

$$(5) \quad \text{Col}_G^{[F]}(K) = \{F_c : c \in [G(K), G]\}.$$

Example 4.1. Consider the functor $F(G) = G^{ab} = G/[G, G]$. The following commands show that the invariant $\text{Col}_G^{[ab]}(K) = \{H_c^{ab} : c \in [G(K), G]\}$ with $G = S_6$ distinguishes between all 801 unoriented equivalence classes of prime knots on ≤ 11 crossings. This establishes a version of Theorem 1.2 for knots on ≤ 11 crossings.

```
gap> L:=[];;
gap> for n in [3..11] do
> for i in [1..NumberOfPrimeKnots(n)] do
> Add(L,SimplifiedFpGroup(WirtingerGroup(GaussCodeKnot(n,i))));
> od;od;
gap> F:=AbelianInvariants;; G:=SymmetricGroup(6);;
gap> C:=Classify(L,K->RefinedColouring(K,G,F) );;
gap> Collected(List(C,Size));
[ [ 1, 801 ] ]
```

■

Given a finite connected quandle Q , a knot K and a quandle homomorphism $c: Q(K) \rightarrow Q$ one can construct a group homomorphism $grp(c): G(K) \rightarrow \text{Inn}(Q)$ as follows. The generators x_1, \dots, x_n for the Wirtinger presentation of the knot group $G(K)$ of an n -crossing knot can be mapped in the obvious way to the generators of Joyce's presentation for the knot quandle $Q(K)$. Let $Q(x_i)$ denote the quandle generator corresponding to the group generator x_i . Let $R_{Q(x_i)}$ denote the right multiplication map $Q(K) \rightarrow Q(K), y \mapsto y * Q(x_i)$. Let $R_{cQ(x_i)}$ denote the right multiplication map $Q \rightarrow Q, y \mapsto y * c(Q(x_i))$. There is a well-known action of the group $G(K)$ on the quandle $Q(K)$ in which the group generator x_i acts on $y \in Q(K)$ by $y^{x_i} = y * Q(x_i)$ (see [9]). The group homomorphism $grp(c): G(K) \rightarrow \text{Inn}(Q)$ is defined on Wirtinger generators by $x_i \mapsto R_{cQ(x_i)}$. We assume the elements of Q are named $1, 2, \dots, n$ and thus regard $\text{Inn}(Q)$ as a subgroup of S_n . We set $H_{grp(c)} = \{g \in G(K) : 1^{grp(c)(g)} = 1\}$. The connectivity of the finite quandle Q ensures that $H_{grp(c)}$ is well-defined up to conjugacy of subgroups in $G(K)$. We define

$$(6) \quad \text{Col}_Q^F(K) = \{F_{grp(c)} : c \in \text{Hom}(Q(K), Q)\}.$$

Example 4.2. The following commands compute $\text{Col}_Q^F(K)$ for the trefoil knot K , the connected quandle Q of order 3, and functor $F(H) = H/[H, H]$.

```
gap> K:=GaussCodeKnot(3,1);;
gap> Q:=ConnectedQuandle(3,1);;
gap> F:=AbelianInvariants;;
gap> RefinedColouring(K,Q,F);
[ [ 0 ], [ 0, 0 ] ]
```

Using such commands Example 4.1 is readily adapted to establish Proposition 4.1. This proposition provides a more detailed version of the statement in Theorem 1.3. ■

In the following proposition we let $Q_{n,i}$ denote the i th quandle in HAP's enumeration of connected quandles of order n .

Proposition 4.1. *The twenty-six knot invariants $Col_{Q_{n,i}}^{ab}(K)$ for $(n, i) \in \{(4, 1), (5, 1), (5, 3), (7, 1), (7, 2), (7, 3), (7, 5), (8, 2), (9, 2), (9, 3), (9, 6), (11, 1), (11, 2), (11, 3), (11, 5), (11, 7), (12, 6), (12, 7), (13, 1), (13, 2), (13, 4), (13, 5), (17, 2), (17, 5), (19, 5), (24, 7)\}$ together distinguish between all 801 un-oriented equivalence classes of prime knots on ≤ 11 crossings.*

REFERENCES

- [1] P. Brendel, G. Ellis, M. Juda, and M. Mrozek. Fundamental group algorithm for low dimensional tessellated cw complexes. *arXiv:1507.03396*, July 2015.
- [2] J.C. Cha and C. Livingston. *KnotInfo: Table of Knot Invariants*. KnotInfo web page, 2013. (<http://www.indiana.edu/~knotinfo>).
- [3] W. Edwin Clark, Mohamed Elhamdadi, Masahico Saito, and Timothy Yeatman. Quandle colorings of knots and applications. *J. Knot Theory Ramifications*, 23(6):1450035, 29, 2014.
- [4] G. Ellis. *HAP – Homological Algebra Programming, Version 1.12.3*, 2017. (<http://www.gap-system.org/Packages/hap.html>).
- [5] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.5.6*, 2013. (<http://www.gap-system.org>).
- [6] Richard Henderson, Todd Macedo, and Sam Nelson. Symbolic computation with finite quandles. *J. Symbolic Comput.*, 41(7):811–817, 2006.
- [7] Benita Ho and Sam Nelson. Matrices and finite quandles. *Homology Homotopy Appl.*, 7(1):197–208, 2005.
- [8] Alexander Hulpke, David Stanovský, and Petr Vojtěchovský. Connected quandles and transitive groups. *J. Pure Appl. Algebra*, 220(2):735–758, 2016.
- [9] David Joyce. A classifying invariant of knots, the knot quandle. *J. Pure Appl. Algebra*, 23(1):37–65, 1982.
- [10] Knot Atlas. *Knot Atlas*, 2013. (<http://katlas.math.toronto.edu/wiki/>).
- [11] G. Nagy and P. Vojtechovsky. *LOOPS – Computing with quasigroups and loops in GAP, version 3.3.0*, 2016. (<http://www.gap-system.org/Packages/loops.html>).
- [12] L. Vendramin. *Rig – a GAP package for racks, quandles and Nichols algebras, version 0.6*, 2011. (<http://github.com/vendramin/rig/>).
- [13] L. Vendramin. On the classification of quandles of low order. *J. Knot Theory Ramifications*, 21(9):1250088, 10, 2012.
- [14] F. Waldhausen. On irreducible 3-manifolds which are sufficiently large. *Ann. of Math. (2)*, 87:56–88, 1968.

SCHOOL OF MATHEMATICS, NATIONAL UNIVERSITY OF IRELAND, GALWAY
E-mail address: graham.ellis@nuigalway.ie

INSTITUT NATIONAL DES SCIENCE APPLIQUÉES, ROUEN, NORMANDIE, FRANCE