# Algorithms for twisted conjugacy classes of polycyclic-by-finite groups

Karel Dekimpe, Sam Tertooy

20th February 2020

**Abstract**

We construct two practical algorithms for twisted conjugacy classes of polycyclic-by-finite groups. The first algorithm determines whether two elements of a group are twisted conjugate for two given endomorphisms, under the condition that the Reidemeister coincidence number of these endomorphisms is finite. The second algorithm determines representatives of the Reidemeister coincidence classes of two endomorphisms if their Reidemeister coincidence number is finite, or returns "fail" if the Reidemeister coincidence number is infinite.

## 1 Introduction

Let $G$ and $H$ be groups and let $\varphi$, $\psi : H \to G$ be group homomorphisms. The coincidence group $\mathrm{Coin}(\varphi, \psi)$ of the pair $(\varphi, \psi)$ is the subgroup of $H$ defined by

$$\mathrm{Coin}(\varphi, \psi) = \{h \in H \mid \varphi(h) = \psi(h)\}.$$

Define an equivalence relation $\sim_{\varphi,\psi}$ on $G$ by

$$\forall g_1, g_2 \in G : g_1 \sim_{\varphi,\psi} g_2 \iff \exists h \in H : g_1 = \psi(h) g_2 \varphi(h)^{-1}.$$

The equivalence classes $[g]_{\varphi,\psi}$ are called the *Reidemeister (coincidence) classes* of the pair $(\varphi, \psi)$ or the $(\varphi, \psi)$*-twisted conjugacy classes*. The set of Reidemeister classes is denoted by $\mathfrak{R}(\varphi, \psi)$. The *Reidemeister (coincidence) number* $R(\varphi, \psi)$ is the cardinality of $\mathfrak{R}(\varphi, \psi)$ and is therefore always a positive integer or infinity.

This equivalence relation originates in topological coincidence theory, see [9] for a survey. One of the aims of coincidence theory is, given two continuous maps $f$, $g : X \to Y$ between topological spaces $X$, $Y$, to calculate the number

$$MC(f, g) := \min_{f' \simeq f, g' \simeq g} \#\{x \in X \mid f'(x) = g'(x)\},$$

i.e. the least number of coincidence points among any pair of maps $(f', g')$, with $f'$ in the homotopy class of $f$ and $g'$ in the homotopy class of $g$. The *Nielsen coincidence number* $N(f, g)$, defined as the number of essential coincidence classes of the pair $(f, g)$, is a lower bound for $MC(f, g)$. The *Reidemeister coincidence number* $R(f, g)$, defined as the number of coincidence classes (essential or otherwise) of the pair $(f, g)$, is an upper bound for the Nielsen coincidence number. While the Nielsen number will always be finite and can be zero, the Reidemeister number is either positive or infinite. In general, Nielsen numbers are quite

Research supported by long term structural funding – Methusalem grant of the Flemish Government

difficult to compute, whereas Reidemeister numbers are much easier to calculate. The Reidemeister coincidence number $R(f, g)$ of continuous maps $f$ and $g$ equals the Reidemeister coincidence number $R(f_*, g_*)$ of the induced group homomorphisms $f_*$, $g_* : \pi_1(X) \to \pi_1(Y)$ between the fundamental groups of $X$ and $Y$.

If $f$, $g : M \to M$ are continuous self-maps of an orientable infra-nilmanifold or an infra-solvmanifold $M$ of type (R), or if $g = \mathrm{id}_M$ and $f$ is a continuous self-map of any infra-solvmanifold $M$, then the Nielsen coincidence number $N(f, g)$ equals the Reidemeister coincidence number $R(f, g)$ if the latter is finite, see [4, 5, 7]. The fundamental group of an infra-solvmanifold is a (torsion-free) polycyclic-by-finite group, and conversely, every torsion-free polycyclic-by-finite group is the fundamental group of some infra-solvmanifold [1].

In [13], an authentication scheme is proposed that relies on the "apparent hardness of the twisted conjugacy problem", i.e. given $g_1 \sim_{\varphi,\psi} g_2$, it it assumed to be difficult to calculate some $h$ such that $g_1 = \psi(h)g_2\varphi(h)^{-1}$. Polycyclic groups have been suggested as the platform groups for various cryptosystems, including this authentication scheme [11].

The main goal of this paper is to construct two practical algorithms for endomorphisms of polycyclic-by-finite groups. The first algorithm, which we will call RepTwistConj (short for *Representative for Twisted Conjugation*), takes as input two endomorphisms $\varphi$, $\psi : G \to G$ with finite Reidemeister number $R(\varphi, \psi)$ and two elements $g_1$, $g_2$ of a polycyclic-by-finite group $G$, and returns the following output:

- if $g_1 \sim_{\varphi,\psi} g_2$: an element $h \in G$ such that $g_1 = \psi(h)g_2\varphi(h)^{-1}$,
- if $g_1 \not\sim_{\varphi,\psi} g_2$: "`fail`".

The second algorithm, which we will call RepsReidClasses (short for *Representatives of Reidemeister Classes*), takes as input two endomorphisms $\varphi$, $\psi : G \to G$ of a polycyclic-by-finite group $G$ and returns the following output:

- if $R(\varphi, \psi) < \infty$: a finite subset $\{g_1, \dots, g_n\} \subseteq G$ for which $g_i \not\sim_{\varphi,\psi} g_j$ when $i \neq j$ and $\mathfrak{R}(\varphi, \psi) = \{[g_1]_{\varphi,\psi}, \dots, [g_n]_{\varphi,\psi}\}$,
- if $R(\varphi, \psi) = \infty$: "`fail`".

Together, these algorithms completely determine the Reidemeister coincidence classes when the Reidemeister coincidence number is finite. In particular, this allows us to calculate Reidemeister coincidence numbers of polycyclic-by-finite groups, and thus Reidemeister coincidence numbers of infra-solvmanifolds as well. Moreover, these algorithms demonstrate that if a polycyclic group is used as platform group for the authentication scheme from [13], then the endomorphisms should be picked such that they have infinite Reidemeister coincidence number.

## 2 Preliminaries

Throughout this paper, we will use the notation $\iota_x$ to describe the inner automorphism $G \to G : g \mapsto xgx^{-1}$.

**Lemma 2.1.** *Let $G$ be a group, $\varphi, \psi \in \mathrm{End}(G)$ and $g_1, g_2 \in G$. For any $x \in G$, we have that*

$$g_1 \sim_{\varphi,\psi} g_2 \iff g_1x^{-1} \sim_{\iota_x\varphi,\psi} g_2x^{-1},$$

*and moreover $\{h \in G \mid g_1 = \psi(h)g_2\varphi(h)^{-1}\} = \{h \in G \mid g_1x^{-1} = \psi(h)g_2x^{-1}(\iota_x\varphi)(h)^{-1}\}$.*

*Proof.* For any $h \in G$, we have that

$$\begin{aligned} g_1 = \psi(h)g_2\varphi(h)^{-1} &\iff g_1x^{-1} = \psi(h)g_2x^{-1}x\varphi(h)^{-1}x^{-1} \\ &\iff g_1x^{-1} = \psi(h)g_2x^{-1}(\iota_x\varphi)(h)^{-1}. \end{aligned}$$ □

By taking $x = g_2$ in the above lemma, we obtain the following corollary.

**Corollary 2.2.** *Let $g_1, g_2 \in G$ and $\varphi, \psi \in \mathrm{End}(G)$. Then $g_1 \sim_{\varphi,\psi} g_2$ if and only if $g_1 g_2^{-1} \sim_{\iota_{g_2}\varphi,\psi} 1$.*

Thus, it suffices to solve the twisted conjugacy problem in the case where one of the elements is the identity. This does, however, involve composing one of the homomorphisms with an inner automorphism. The following corollary shows that this does not impact the finiteness of the Reidemeister coincidence number of the endomorphisms.

**Corollary 2.3.** *Let $g \in G$ and let $\varphi, \psi \in \mathrm{End}(G)$. Then the map $\mu_g : \mathfrak{R}(\iota_g\varphi, \psi) \to \mathfrak{R}(\varphi, \psi) : [x]_{\iota_g\varphi,\psi} \mapsto [xg]_{\varphi,\psi}$ is a bijection, and therefore $R(\varphi, \psi) = R(\iota_g\varphi, \psi)$.*

Therefore, should we have an algorithm $\textsc{RepTwistConjToId}(\varphi, \psi, g)$ that takes as input two endomorphisms $\varphi$, $\psi$ with finite Reidemeister number $R(\varphi, \psi)$ and an element $g$, and returns the following output:

- if $g \sim_{\varphi,\psi} 1$: an element $h \in G$ such that $g = \psi(h)\varphi(h)^{-1}$,
- if $g \not\sim_{\varphi,\psi} 1$: “`fail`”.

then we may construct the algorithm $\textsc{RepTwistConj}$ as in algorithm 1.

**Algorithm 1** Determining $h$ such that $g_1 = \psi(h) g_2 \varphi(h)^{-1}$

```
function RepTwistConj($\varphi, \psi, g_1, g_2$)
    return RepTwistConjToId($\iota_{g_2}\varphi, \psi, g_1 g_2^{-1}$)
end function
```

The following theorem will be crucial in constructing both $\textsc{RepTwistConjToId}$ and $\textsc{RepsReidClasses}$ for polycyclic and polycyclic-by-finite groups.

**Theorem 2.4** (see [12, §2])**.** *Let $G$ be group, let $N$ be a normal subgroup of $G$ and let $\varphi, \psi \in \mathrm{End}(G)$ such that $\varphi(N) \subseteq N$ and $\psi(N) \subseteq N$. We denote the restrictions of $\varphi$ and $\psi$ to $N$ by $\varphi|_N$ and $\psi|_N$, and the induced endomorphisms on $G/N$ by $\bar{\varphi}$ and $\bar{\psi}$. We then get the following commutative diagram with exact rows:*

$$\begin{tikzcd}
1 \arrow[r] & N \arrow[r, "i"] \arrow[d, shift right, "\psi|_N"'] \arrow[d, shift left, "\varphi|_N"] & G \arrow[r, "p"] \arrow[d, shift right, "\psi"'] \arrow[d, shift left, "\varphi"] & G/N \arrow[r] \arrow[d, shift right, "\bar{\psi}"'] \arrow[d, shift left, "\bar{\varphi}"] & 1 \\
1 \arrow[r] & N \arrow[r, "i"] & G \arrow[r, "p"] & G/N \arrow[r] & 1
\end{tikzcd}$$

*This diagram induces the following exact sequence of pointed sets:*

$$\begin{tikzcd}
1 \arrow[r] & \mathrm{Coin}(\varphi|_N, \psi|_N) \arrow[r, "i"] & \mathrm{Coin}(\varphi, \psi) \arrow[r, "p"] & \mathrm{Coin}(\bar{\varphi}, \bar{\psi}) \arrow[dll, rounded corners, "\delta"] & \\
& \mathfrak{R}(\varphi|_N, \psi|_N) \arrow[r, "\hat{\imath}"] & \mathfrak{R}(\varphi, \psi) \arrow[r, "\hat{p}"] & \mathfrak{R}(\bar{\varphi}, \bar{\psi}) \arrow[r] & 1
\end{tikzcd}$$

*where all maps are evident except $\delta$, which is defined as $\delta(\bar{g}) = [\psi(g)\varphi(g)^{-1}]_{\varphi|_N,\psi|_N}$.*

The corollary below is a straightforward generalisation of statements (1) and (2) in [10, Lemma 1.1].

**Corollary 2.5.** *Consider the situation from theorem 2.4. We obtain the following properties:*

*(1)* $R(\varphi, \psi) \geq R(\bar{\varphi}, \bar{\psi})$,

*(2) if* $\# \operatorname{Coin}(\bar{\varphi}, \bar{\psi}) < \infty$ *and* $R(\varphi, \psi) < \infty$, *then* $R(\varphi|_N, \psi|_N) < \infty$.

# 3 Reduction to normal subgroup and quotient

It is possible to reduce the twisted conjugacy problem on a group to the twisted conjugacy problem on a well-chosen normal subgroup and on the quotient by that subgroup.

**Theorem 3.1.** *Consider the situation from theorem 2.4. Let* $g \in G$. *If* $\bar{g} \sim_{\bar{\varphi},\bar{\psi}} \bar{1}$, *then there exists an* $n \in N$ *such that* $n \sim_{\varphi,\psi} g$ *and*

$$g \sim_{\varphi,\psi} 1 \iff \exists \bar{h} \in \operatorname{Coin}(\bar{\varphi}, \bar{\psi}) : \psi(h)^{-1} n \varphi(h) \sim_{\varphi|_N, \psi|_N} 1,$$

*where* $h$ *is any element of* $p^{-1}(\bar{h})$.

*Proof.* If $\bar{g} \sim_{\bar{\varphi},\bar{\psi}} \bar{1}$, then there exists some $\bar{k} \in G/N$ such that

$$\bar{g} = \bar{\psi}(\bar{k}) \bar{\varphi}(\bar{k})^{-1} \iff \bar{\psi}(\bar{k})^{-1} \bar{g} \bar{\varphi}(\bar{k}) = \bar{1}.$$

Let $k \in G$ be any preimage of $\bar{k}$, then $n := \psi(k)^{-1} g \varphi(k)$ is an element of $N$ and clearly $n \sim_{\varphi,\psi} g$. Now, using the exact sequence from theorem 2.4, we find that

$$\begin{aligned}
[g]_{\varphi,\psi} = [1]_{\varphi,\psi} &\iff [n]_{\varphi,\psi} = [1]_{\varphi,\psi} \\
&\iff \hat{\imath}([n]_{\varphi|_N,\psi|_N}) = [1]_{\varphi,\psi} \\
&\iff \exists \bar{h} \in \operatorname{Coin}(\bar{\varphi}, \bar{\psi}) : [n]_{\varphi|_N,\psi|_N} = [\psi(h)\varphi(h)^{-1}]_{\varphi|_N,\psi|_N} \\
&\iff \exists \bar{h} \in \operatorname{Coin}(\bar{\varphi}, \bar{\psi}) : [\psi(h)^{-1} n \varphi(h)]_{\varphi|_N,\psi|_N} = [1]_{\varphi|_N,\psi|_N},
\end{aligned}$$

where we used the normality of $N$ to obtain the last equivalence. □

Thus, we can construct algorithm 2, called RepTwistConjToIdByNormal, which reduces the twisted conjugacy problem on $G$ to the twisted conjugacy problem on a normal subgroup $N$ and on the quotient $G/N$. In order for this algorithm to work, we require 4 conditions on the endomorphisms $\varphi, \psi$ and the normal subgroup $N$ given as input:

(i) $\varphi(N) \subseteq N$ and $\psi(N) \subseteq N$, such that $\bar{\varphi}$, $\bar{\psi}$, $\varphi|_N$ and $\psi|_N$ are well-defined,

(ii) $\# \operatorname{Coin}(\bar{\varphi}, \bar{\psi}) < \infty$, because line 9 iterates over all elements of this group.

(iii) RepTwistConjToId is implemented for input $\bar{\varphi}, \bar{\psi}$, because line 3 calls this,

(iv) RepTwistConjToId is implemented for input $\varphi|_N, \psi|_N$, because line 11 calls this.

Note that we currently do not require that $R(\varphi, \psi) < \infty$.

Making use of the exact sequence from theorem 2.4, we may describe the set of Reidemeister classes $\mathfrak{R}(\varphi, \psi)$ in terms of Reidemeister classes of a well-chosen normal subgroup and of the quotient by that subgroup.

**Theorem 3.2.** *Consider the situation from theorem 2.4. The set of Reidemeister classes of the pair* $(\varphi, \psi)$ *is given by*

$$\mathfrak{R}(\varphi, \psi) = \bigsqcup_{[\bar{g}]_{\bar{\varphi},\bar{\psi}} \in \mathfrak{R}(\bar{\varphi},\bar{\psi})} (\mu_g \circ \hat{\imath}_g)(\mathfrak{R}(\iota_g \varphi|_N, \psi|_N)),$$

**Algorithm 2** Determining $h$ such that $g = \psi(h)\varphi(h)^{-1}$

```
1:  function RepTwistConjToIdByNormal($\varphi, \psi, g, N$)
2:      $p :=$ projection $G \to G/N$
3:      $\bar{k} :=$ RepTwistConjToId($\bar{\varphi}, \bar{\psi}, p(g)$)
4:      if $\bar{k} =$ fail then
5:          return fail
6:      end if
7:      $k :=$ any element in $p^{-1}(\bar{k})$
8:      $n := \psi(k)^{-1} g \varphi(k)$
9:      for $\bar{h} \in \mathrm{Coin}(\bar{\varphi}, \bar{\psi})$ do
10:         $h :=$ any element in $p^{-1}(\bar{h})$
11:         $l :=$ RepTwistConjToId($\varphi|_N, \psi|_N, \psi(h)^{-1} n \varphi(h)$)
12:         if $l \neq$ fail then
13:             return $khl$
14:         end if
15:     end for
16:     return fail
17: end function
```

*where $\hat{\imath}_g$ is the map*

$$\hat{\imath}_g : \mathfrak{R}(\iota_g\varphi|_N, \psi|_N) \to \mathfrak{R}(\iota_g\varphi, \psi) : [x]_{\iota_g\varphi|_N, \psi|_N} \to [x]_{\iota_g\varphi, \psi}$$

*and $\mu_g$ is the map from corollary 2.3.*

*Proof.* From the surjectivity of $\hat{p}$, we have that

$$\mathfrak{R}(\varphi, \psi) = \bigsqcup_{[\bar{g}]_{\bar{\varphi}, \bar{\psi}} \in \mathfrak{R}(\bar{\varphi}, \bar{\psi})} \hat{p}^{-1}([\bar{g}]_{\bar{\varphi}, \bar{\psi}}). \tag{1}$$

Let $\hat{p}_g$ be the map

$$\hat{p}_g : \mathfrak{R}(\iota_g\varphi, \psi) \to \mathfrak{R}(\iota_{\bar{g}}\bar{\varphi}, \bar{\psi}) : [x]_{\iota_g\varphi, \psi} \to [\bar{x}]_{\iota_{\bar{g}}\bar{\varphi}, \bar{\psi}},$$

then by corollary 2.3 and the exact sequence from theorem 2.4 we obtain that

$$\hat{p}^{-1}([\bar{g}]_{\bar{\varphi}, \bar{\psi}}) = \mu_g(\hat{p}_g^{-1}([\bar{1}]_{\iota_{\bar{g}}\bar{\varphi}, \bar{\psi}})) = (\mu_g \circ \hat{\imath}_g)(\mathfrak{R}(\iota_g\varphi|_N, \psi|_N)). \tag{2}$$

The result now follows by combining (1) and (2). □

Similar to the previous algorithm, we can construct algorithm 3. This time, we require 5 conditions on the endomorphisms $\varphi, \psi$ and the normal subgroup $N$ given as input in order for this algorithm to work as intended:

(i) $\varphi(N) \subseteq N$ and $\psi(N) \subseteq N$, such that $\bar{\varphi}$, $\bar{\psi}$, $\varphi|_N$ and $\psi|_N$ are well-defined,

(ii) RepsReidClasses is implemented for input $\bar{\varphi}, \bar{\psi}$, because line 3 calls this,

(iii) RepsReidClasses is implemented for input $\iota_g\varphi|_N, \psi|_N$, because line 10 calls this,

(iv) If $R(\bar{\varphi}, \bar{\psi}) < \infty$ and $R(\iota_g\varphi|_N, \psi|_N) = \infty$ for some $g \in G$, then $R(\varphi, \psi) = \infty$, because line 12 makes this assumption,

(v) RepTwistConj is implemented for input $\iota_g\varphi, \psi$, because line 16 calls this.

**Algorithm 3** Determining representatives of $\mathfrak{R}(\varphi, \psi)$

```
function RepsReidClassesByNormal($\varphi, \psi, N$)
    $p :=$ projection $G \to G/N$
    $\mathfrak{R}(\bar{\varphi}, \bar{\psi}) :=$ RepsReidClasses($\bar{\varphi}, \bar{\psi}$)
    if $\mathfrak{R}(\bar{\varphi}, \bar{\psi}) =$ fail then
        return fail
    end if
    $\mathfrak{R}(\varphi, \psi) := \varnothing$
    for $\bar{g} \in \mathfrak{R}(\bar{\varphi}, \bar{\psi})$ do
        $g :=$ any element in $p^{-1}(\bar{g})$
        $\mathfrak{R}(\iota_g\varphi|_N, \psi|_N) :=$ RepsReidClasses($\iota_g\varphi|_N, \psi|_N$)
        if $\mathfrak{R}(\iota_g\varphi|_N, \psi|_N) =$ fail then
            return fail
        end if
        $\hat{\imath}_g(\mathfrak{R}(\iota_g\varphi|_N, \psi|_N)) := \varnothing$
        for $h \in \mathfrak{R}(\iota_g\varphi|_N, \psi|_N)$ do
            if $\forall k \in \hat{\imath}_g(\mathfrak{R}(\iota_g\varphi|_N, \psi|_N))$ : RepTwistConj($\iota_g\varphi, \psi, h, k$) = fail then
                $\hat{\imath}_g(\mathfrak{R}(\iota_g\varphi|_N, \psi|_N)) := \hat{\imath}_g(\mathfrak{R}(\iota_g\varphi|_N, \psi|_N)) \cup \{h\}$
            end if
        end for
        $\mathfrak{R}(\varphi, \psi) := \mathfrak{R}(\varphi, \psi) \cup \mu_g(\hat{\imath}_g(\mathfrak{R}(\iota_g\varphi|_N, \psi|_N)))$
    end for
    return $\mathfrak{R}(\varphi, \psi)$
end function
```

# 4 Abelian Groups

If the group $G$ is abelian, the set of Reidemeister classes can actually be interpreted as a quotient group of $G$.

**Theorem 4.1.** *Let $G$ be an abelian group and $\varphi, \psi \in \operatorname{End}(G)$. Then $\mathfrak{R}(\varphi, \psi) = \operatorname{coker}(\psi - \varphi)$.*

*Proof.* Let $g_1, g_2 \in G$. Then

$$\begin{aligned} g_1 \sim_{\varphi,\psi} g_2 &\iff \exists h \in G : g_1 = \psi(h) + g_2 - \varphi(h) \\ &\iff \exists h \in G : g_1 - g_2 = (\psi - \varphi)(h) \\ &\iff g_1 + \operatorname{im}(\psi - \varphi) = g_2 + \operatorname{im}(\psi - \varphi). \end{aligned}$$

□

Thus, we can define RepTwistConjToId and RepsReidClasses for finitely generated, abelian groups as in algorithms 4 and 5.

**Algorithm 4** Determining $h$ such that $g = \psi(h)\varphi(h)^{-1}$ if $G$ is abelian

```
function RepTwistConjToId($\varphi, \psi, g$)
    if $g \in \operatorname{im}(\psi - \varphi)$ then
        $h :=$ any element in $(\psi - \varphi)^{-1}(g)$
        return $h$
    end if
    return fail
end function
```

**Algorithm 5** Determining representatives of $\mathfrak{R}(\varphi, \psi)$ if $G$ is abelian

```
function RepsReidClasses($\varphi, \psi$)
    if $[G : \operatorname{im}(\psi - \varphi)] = \infty$ then
        return fail
    end if
    $\mathfrak{R}(\varphi, \psi) := \varnothing$
    $p :=$ projection $G \to G/\operatorname{im}(\psi - \varphi)$
    for $\bar{g} \in G/\operatorname{im}(\psi - \varphi)$ do
        $g :=$ any element in $p^{-1}(\bar{g})$
        $\mathfrak{R}(\varphi, \psi) := \mathfrak{R}(\varphi, \psi) \cup \{g\}$
    end for
    return $\mathfrak{R}(\varphi, \psi)$
end function
```

The following proposition and corollary will be necessary when dealing with abelian quotients of polycyclic groups.

**Proposition 4.2.** *Let $G$ be a finitely generated, abelian group and let $\varphi \in \operatorname{End}(G)$. Then the Hirsch length of the kernel of $\varphi$ equals the Hirsch length of the cokernel of $\varphi$.*

*Proof.* It is well known that for any polycyclic group with normal subgroup $N$, $h(G) = h(N) + h(G/N)$. Since $\operatorname{im}(\varphi) \cong G/\ker(\varphi)$ and $\operatorname{coker}(\varphi) = G/\operatorname{im}(\varphi)$, we obtain

$$h(\ker(\varphi)) + h(\operatorname{im}(\varphi)) = h(G) = h(\operatorname{im}(\varphi)) + h(\operatorname{coker}(\varphi)).$$

Subtracting $h(\operatorname{im}(\varphi))$ from both sides gives us the desired result. □

**Corollary 4.3.** *Let $G$ be a finitely generated, abelian group and let $\varphi, \psi \in \operatorname{End}(G)$. Then $R(\varphi, \psi)$ is finite if and only if $\operatorname{Coin}(\varphi, \psi)$ is finite.*

*Proof.* Note that $\mathfrak{R}(\varphi, \psi) = \operatorname{coker}(\psi - \varphi)$ (see theorem 4.1) and that $\operatorname{Coin}(\varphi, \psi) = \ker(\psi - \varphi)$. By proposition 4.2, if either of these groups has Hirsch length 0, then so does the other. □

# 5 Polycyclic groups

One way to define a polycyclic group, is to state that all of its subgroups are finitely generated and that its derived series terminates at the trivial subgroup. This derived series will be exceptionally useful in the context of twisted conjugacy, as every group in this series is fully invariant and the factors are finitely generated, abelian groups.

**Proposition 5.1.** *Consider the situation from theorem 2.4, where $G$ and $N$ are chosen in such way that $G/N$ is a finitely generated, abelian group. If $R(\varphi, \psi)$ is finite, then so are $\# \operatorname{Coin}(\bar{\varphi}, \bar{\psi})$, $R(\bar{\varphi}, \bar{\psi})$ and $R(\varphi|_N, \psi|_N)$.*

*Proof.* If $R(\varphi, \psi) < \infty$, then by corollary 2.5(1) $R(\bar{\varphi}, \bar{\psi}) < \infty$ and thus corollary 4.3 gives us that $\# \operatorname{Coin}(\bar{\varphi}, \bar{\psi}) < \infty$. Finally, by corollary 2.5(2) $R(\varphi|_N, \psi|_N)$ is finite as well. □

Algorithm 6 provides an implementation of RepTwistConjToId for polycyclic groups of derived length at least 2, under the restriction that the pair of endomorphisms given as input has finite Reidemeister number.

**Theorem 5.2.** *Let $G$ be a polycyclic group of derived length at least 2 and let $\varphi, \psi \in \operatorname{End}(G)$ such that $R(\varphi, \psi) < \infty$. Then $\varphi$, $\psi$ and $G'$ satisfy the conditions necessary to apply algorithm 2.*

*Proof.* We prove this condition by condition.

(i) This condition is satisfied because the derived subgroup $G'$ is fully invariant.

(ii) Since $R(\varphi, \psi) < \infty$ and $G/G'$ is finitely generated and abelian, proposition 5.1 gives us that this condition is satisfied.

(iii) Algorithm 4 provides an implementation for endomorphisms of $G/G'$.

(iv) We prove this by induction on the derived length $n$ of $G$. If $n = 2$, then $G'$ is abelian, hence algorithm 4 provides an implementation for endomorphisms of $G'$. Now assume that $G$ has derived length $n$ and that this theorem holds if the derived length is at most $n-1$. By proposition 5.1 and the induction hypothesis, $\varphi|_{G'}$, $\psi|_{G'}$ and $G''$ satisfy conditions (i) - (iv), thus algorithm 6 provides an implementation. □

**Algorithm 6** Determining $h$ such that $g = \psi(h)\varphi(h)^{-1}$ if $G$ is polycyclic

```
function RepTwistConjToId($\varphi, \psi, g$)
    return RepTwistConjToIdByNormal($\varphi, \psi, g, G'$)
end function
```

**Proposition 5.3.** *Let $G$ be a polycyclic group and $\varphi, \psi \in \mathrm{End}(G)$. Let $\bar{\varphi}, \bar{\psi}$ be the induced endomorphisms on the abelianisation $G/G'$. Then $R(\varphi, \psi)$ is finite if and only if $R(\bar{\varphi}, \bar{\psi})$ is finite and $R(\iota_g\varphi|_{G'}, \psi|_{G'})$ is finite for every $g \in G$.*

*Proof.* First assume that $R(\varphi, \psi) < \infty$. By corollary 2.3, then $R(\iota_g\varphi, \psi) < \infty$ for all $g \in G$, and by applying proposition 5.1 we indeed find that $R(\bar{\varphi}, \bar{\psi}) < \infty$ and $R(\iota_g\varphi|_{G'}, \psi|_{G'}) < \infty$ for every $g \in G$. Conversely, assume that $R(\bar{\varphi}, \bar{\psi}) < \infty$ and $R(\iota_g\varphi|_{G'}, \psi|_{G'}) < \infty$ for every $g \in G$. By theorem 3.2, $\mathfrak{R}(\varphi, \psi)$ is then a finite union of finite sets and hence $R(\varphi, \psi) < \infty$. □

Algorithm 7 provides an implementation of RepsReidClasses for polycyclic groups of derived length at least 2.

**Theorem 5.4.** *Let $G$ be a polycyclic group of derived length at least 2 and let $\varphi, \psi \in \mathrm{End}(G)$. Then $\varphi$, $\psi$ and $G'$ satisfy satisfy the conditions necessary to apply algorithm 3.*

*Proof.* We prove this condition by condition.

(i) - (iii) This can be proven in the same way as theorem 5.2.

(iv) This follows from proposition 5.3.

(v) Algorithm 6 provides this implementation. □

**Algorithm 7** Determining representatives of $\mathfrak{R}(\varphi, \psi)$ if $G$ is polycyclic

```
function RepsReidClasses($\varphi, \psi$)
    return RepsReidClassesByNormal($\varphi, \psi, G'$)
end function
```

# 6 Polycyclic-by-finite groups

For polycyclic-by-finite groups, we can implement the algorithms RepTwistConjToId and RepsReidClasses as in algorithms 8 and 9, under the assumption that we have an implementation of an algorithm that finds a fully invariant, finite index, polycyclic subgroup $N$ of a given polycyclic-by-finite group $G$.

**Algorithm 8** Determining $h$ such that $g = \psi(h)\varphi(h)^{-1}$ if $G$ is polycyclic-by-finite

```
function RepTwistConjToId($\varphi, \psi, g$)
    $N :=$ fully invariant, finite index, polycyclic subgroup of $G$
    return RepTwistConjToIdByNormal($\varphi, \psi, g, N$)
end function
```

**Theorem 6.1.** *Let $G$ be a polycyclic-by-finite group, let $\varphi, \psi \in \operatorname{End}(G)$ such that $R(\varphi, \psi) < \infty$ and let $N$ be a fully invariant, finite index, polycyclic subgroup of $G$. Then $\varphi$, $\psi$ and $N$ satisfy the conditions needed to apply algorithm 2.*

*Proof.* We prove this condition by condition.

(i) This condition is satisfied because $N$ is fully invariant.

(ii) Since $\operatorname{Coin}(\bar{\varphi}, \bar{\psi})$ is a subgroup of $G/N$, it is finite.

(iii) Since $G/N$ is finite, we can easily implement RepsTwistConjToId for $\bar{\varphi}, \bar{\psi}$, e.g. by exhaustive search.

(iv) Algorithm 6 provides an implementation for endomorphisms of $N$. □

**Algorithm 9** Determining representatives of $\mathfrak{R}(\varphi, \psi)$ if $G$ is polycyclic-by-finite

```
function RepsReidClasses($\varphi, \psi$)
    $N :=$ fully invariant, finite index, polycyclic subgroup of $G$
    return RepsReidClassesByNormal($\varphi, \psi, N$)
end function
```

**Theorem 6.2.** *Let $G$ be a polycyclic-by-finite group, let $\varphi, \psi \in \operatorname{End}(G)$ such that $R(\varphi, \psi) < \infty$ and let $N$ be a fully invariant, finite index, polycyclic subgroup of $G$. Then $\varphi$, $\psi$ and $N$ satisfy the conditions needed to apply algorithm 3.*

*Proof.* We prove this condition by condition.

(i) - (iii) This can be proven in the same way as theorem 6.1.

(iv) This follows from corollary 2.5(2).

(v) Algorithm 8 provides this implementation. □

A (theoretical) algorithm to find the required subgroup $N$ exists, as proven in the following proposition.

**Proposition 6.3.** *There is an algorithm which finds a fully invariant, finite index, polycyclic subgroup $N$ of a polycyclic-by-finite group $G$.*

*Proof.* There exists an algorithm to find a finite index, polycyclic, normal subgroup $P$ of $G$ (see [2, Proposition 2.8]). Let $m$ be the exponent of the finite quotient $G/P$, i.e. the smallest positive integer such that $\bar{g}^m = \bar{1}$ for any $\bar{g} \in G/P$. Then $N := \langle g^m \mid g \in G \rangle$ is a fully invariant, finite index, polycyclic subgroup of $G$. There exists an algorithm to find such a subgroup of a polycyclic-by-finite group (see [2, Proposition 2.10]). □

Unfortunately, to the best of our knowledge, no implementation of a *practical* algorithm to determine such a fully invariant, finite index, polycyclic subgroup $N$ is as of yet available. This also means that algorithms 8 and 9 currently cannot be implemented. However, if one has extra information on the input, one may be able to pick a suitable subgroup $N$ for that specific input. For example, in [3] Reidemeister numbers of the form $R(\varphi, \mathrm{id})$ with $\varphi \in \mathrm{Aut}(G)$ were calculated for crystallographic groups $G$. Algorithm 9 reduces to [3, Algorithm 3] if $G$ is crystallographic, $\varphi \in \mathrm{Aut}(G)$, $\psi = \mathrm{id}$ and $N = \mathrm{Fitt}(G)$, the Fitting subgroup of $G$.

# 7 Implementation in GAP

Algorithms 1 to 7 have been implemented in the computer algebra system GAP [8], as part of a package called `TwistedConjugacy` [14]. Below, we give a short demonstration of how to access our algorithms using this package. By way of example, let $G$ be the group given by the following presentation:

$$G := \left\langle g_1, g_2, g_3, g_4 \;\middle|\; \begin{array}{ll} [g_1, g_2] = g_2^2 & [g_1, g_4] = 1 \\ [g_1, g_3] = g_3^2 & [g_2, g_4] = 1 \\ [g_2, g_3] = g_4^{-2} & [g_3, g_4] = 1 \\ g_1^2 = g_4 & \end{array} \right\rangle.$$

This is a polycyclic group of derived length 3, and can be accessed in GAP through the command `ExamplesOfSomePcpGroups` provided by the `polycyclic` package [6]. Let $\varphi$ and $\psi$ be the endomorphisms of $G$ given by

$$\begin{aligned} \varphi(g_1) &= g_1 g_4^{-1}, & \psi(g_1) &= g_1, \\ \varphi(g_2) &= g_3, & \psi(g_2) &= g_2^2 g_3 g_4^2, \\ \varphi(g_3) &= g_2 g_3^3 g_4^3, & \psi(g_3) &= g_2 g_3 g_4, \\ \varphi(g_4) &= g_4^{-1}, & \psi(g_4) &= g_4. \end{aligned}$$

One may load the `TwistedConjugacy` package and construct $G$, $\varphi$ and $\psi$ as follows.

```
gap> LoadPackage("TwistedConjugacy");;
gap> G := ExamplesOfSomePcpGroups( 5 );;
gap> gens := GeneratorsOfGroup( G );;
gap> imgs1 := [ G.1*G.4^-1, G.3, G.2*G.3^3*G.4^3, G.4^-1  ];;
gap> phi := GroupHomomorphismByImages( G, G, gens, imgs1 );
[ g1, g2, g3, g4 ] -> [ g1*g4^-1, g3, g2*g3^3*g4^3, g4^-1 ]
gap> imgs2 := [ G.1, G.2^2*G.3*G.4^2, G.2*G.3*G.4, G.4  ];;
gap> psi := GroupHomomorphismByImages( G, G, gens, imgs2 );
[ g1, g2, g3, g4 ] -> [ g1, g2^2*g3*g4^2, g2*g3*g4, g4 ]
```

The command `RepresentativeTwistedConjugation` provides an implementation of the RepTwistConj algorithm. We can use it to show that $g_1$ and $g_1^2$ are not $(\varphi, \psi)$-twisted conjugate and that $g_1$ and $g_1^3$ are.

```
gap> RepresentativeTwistedConjugation( phi, psi, G.1, G.1^2 );
fail
gap> RepresentativeTwistedConjugation( phi, psi, G.1, G.1^3 );
g1*g4^-1
```

The command `ReidemeisterClasses` provides an implementation of the RepsReidClasses algorithm. We use it to show that $R(\mathrm{id}, \psi) = \infty$ and to calculate representatives of the Reidemeister classes of $(\varphi, \psi)$:

```
gap> ReidemeisterClasses( IdentityMapping( G ), psi );
fail
gap> ReidemeisterClasses( phi, psi );
[ id^G, g1*g2*g3^G, g1*g2^G, g1*g3^G, g1^G, g2*g3^G, g2^G, g3^G ]
```

Note that the "`^G`" in the output above indicates that these elements are representatives of the orbits of a group action. For more information on the `TwistedConjugacy` package for GAP, we refer to the package manual.

# References

[1] Oliver Baues. Infra-solvmanifolds and rigidity of subgroups in solvable linear algebraic groups. *Topology*, 43(4):903–924, 2004.

[2] Gilbert Baumslag, Frank B. Cannonito, Derek J. Robinson, and Dan Segal. The algorithmic theory of polycyclic-by-finite groups. *J. Algebra*, 142(1):118–149, 1991.

[3] Karel Dekimpe, Tom Kaiser, and Sam Tertooy. The reidemeister spectra of low dimensional crystallographic groups. *Journal of Algebra*, 533:353–375, 2019.

[4] Karel Dekimpe and Pieter Penninckx. The finiteness of the Reidemeister number of morphisms between almost-crystallographic groups. *J. Fixed Point Theory Appl.*, 9(2):257–283, 2011.

[5] Karel Dekimpe and Iris Van den Bussche. An averaging formula for nielsen numbers on infra-solvmanifolds. In preparation, 2020.

[6] Bettina Eick, Werner Nickel, and Max Horn. Polycyclic, computation with polycyclic groups, Version 2.14. `https://gap-packages.github.io/polycyclic/`, 2018. Refereed GAP package.

[7] Alexander Fel'shtyn and Jong Bum Lee. The Nielsen and Reidemeister numbers of maps on infra-solvmanifolds of type (R). *Topology Appl.*, 181:62–103, 2015.

[8] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.10.2*, 2019.

[9] Daciberg L. Gonçalves. Coincidence theory. In *Handbook of topological fixed point theory*, pages 3–42. Springer, Dordrecht, 2005.

[10] Daciberg L. Gonçalves and Peter Wong. Twisted conjugacy classes in nilpotent groups. *J. Reine Angew. Math.*, 633:11–27, 2009.

[11] Jonathan Gryak and Delaram Kahrobaei. The status of polycyclic group-based cryptography: a survey and open problems. *Groups Complex. Cryptol.*, 8(2):171–186, 2016.

[12] Seung Won Kim and Jong Bum Lee. Averaging formula for Nielsen coincidence numbers. *Nagoya Math. J.*, 186:69–93, 2007.

[13] Vladimir Shpilrain and Alexander Ushakov. An authentication scheme based on the twisted conjugacy problem. In *Applied Cryptography and Network Security*, pages 366–372, Berlin, Heidelberg, 2008. Springer.

[14] Sam Tertooy. TwistedConjugacy, computation with twisted conjugacy classes, Version 1.0.0. `https://sTertooy.github.io/TwistedConjugacy/`, 2020. GAP package.