Constructing the automorphism group of a finite group

Eamonn O'Brien

University of Auckland

June 2022

The problem

Given finite G, construct $Aut(G)$.

The problem

Given finite G, construct $Aut(G)$.

Shoda (1928), Hulpke (1997): G abelian

The problem

Given finite G, construct $Aut(G)$.

Shoda (1928), Hulpke (1997): G abelian

Felsch & Neubüser (1968): Choose a generating set for G and systematically list maps defined on this generating set.

The problem

Given finite G, construct $Aut(G)$.

Shoda (1928), Hulpke (1997): G abelian

Felsch & Neubüser (1968): Choose a generating set for G and systematically list maps defined on this generating set.

Cannon and Neubüser (1970s), Robertz (1976): exploited BSGS machinery for permutation group G, obtain $Aut(G)$ acting on unions of certain conjugacy classes of G.

The problem

Given finite G, construct $Aut(G)$.

Shoda (1928), Hulpke (1997): G abelian

Felsch & Neubüser (1968): Choose a generating set for G and systematically list maps defined on this generating set.

Cannon and Neubüser (1970s), Robertz (1976): exploited BSGS machinery for permutation group G, obtain $Aut(G)$ acting on unions of certain conjugacy classes of G.

Cannon and Holt (2003): use structure of $G/O_\infty(G)$ to obtain answer, and then lift results through elementary abelian layers.

Finite soluble group

Smith (1994) and Slattery.

G soluble defined by power-conjugate presentation: lift computations through normal series with elementary abelian layers.

Finite soluble group

Smith (1994) and Slattery.

G soluble defined by power-conjugate presentation: lift computations through normal series with elementary abelian layers.

Howden (2008): soluble case reduced to p-groups.

Finite soluble group

Smith (1994) and Slattery.

G soluble defined by power-conjugate presentation: lift computations through normal series with elementary abelian layers.

Howden (2008): soluble case reduced to p-groups.

Hard case: G finite p-group.

O'B (1993); Eick, Leedham-Green, O'B (2003).

Lower exponent-p central series

The lower p-central series of a p-group G is defined by
$\mathcal{P}_0(G) = G$
$\mathcal{P}_{i+1}(G) = [\mathcal{P}_i(G), G]\mathcal{P}_i(G)^p$ for $i \geq 0$

Lower exponent-p central series

The lower p-central series of a p-group G is defined by
$\mathcal{P}_0(G) = G$
$\mathcal{P}_{i+1}(G) = [\mathcal{P}_i(G), G]\mathcal{P}_i(G)^p$ for $i \geq 0$

Factors are elementary abelian p-groups.

If $\mathcal{P}_c(G) = 1$, then G has p-class c.

Lower exponent-p central series

The lower p-central series of a p-group G is defined by
$\mathcal{P}_0(G) = G$
$\mathcal{P}_{i+1}(G) = [\mathcal{P}_i(G), G]\mathcal{P}_i(G)^p$ for $i \geq 0$

Factors are elementary abelian p-groups.

If $\mathcal{P}_c(G) = 1$, then G has p-class c.

Let $G_i = G/\mathcal{P}_i(G)$.

Lower exponent-p central series

The lower p-central series of a p-group G is defined by
$\mathcal{P}_0(G) = G$
$\mathcal{P}_{i+1}(G) = [\mathcal{P}_i(G), G]\mathcal{P}_i(G)^p$ for $i \geq 0$

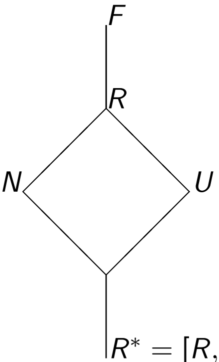
Factors are elementary abelian p-groups.

If $\mathcal{P}_c(G) = 1$, then G has p-class c.

Let $G_i = G/\mathcal{P}_i(G)$.

Proceed by induction down the lower p-central series:
$G_1 = G/\mathcal{P}_1(G)$ is elementary abelian of order p^d, and
$Aut(G_1) \cong GL(d, p)$.

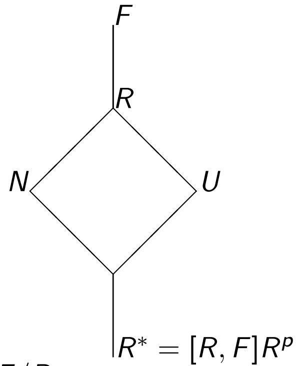
The general framework: G_i to G_{i+1}



$R^* = [R, F]R^p$

$G_i := F/R$

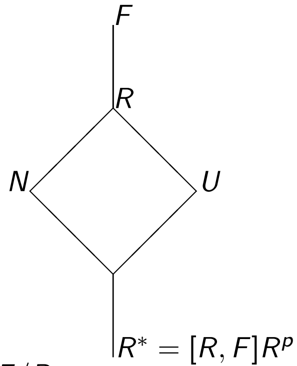
The general framework: G_i to G_{i+1}



$G_i := F/R$

$P := F/[R,F]R^p$ the p-covering group of G_i

The general framework: G_i to G_{i+1}



$G_i := F/R$

$P := F/[R,F]R^p$ the p-covering group of G_i

$M := R/R^*$ the p-multiplicator, characteristic

$N := \mathcal{P}_i(P) \leq M$

$G_{i+1} := P/U$ where U supplements N.

Inductive step: compute $Aut(G_{i+1})$ from $Aut(G_i)$

- Compute the p-covering group P of G_i.

Inductive step: compute $Aut(G_{i+1})$ from $Aut(G_i)$

- Compute the p-covering group P of G_i.

- Each automorphism of G_i lifts to an automorphism of P via natural homomorphism $P \to G_i$ with kernel M.

Inductive step: compute $Aut(G_{i+1})$ from $Aut(G_i)$

- Compute the p-covering group P of G_i.

- Each automorphism of G_i lifts to an automorphism of P via natural homomorphism $P \to G_i$ with kernel M.

- Identify U such that $P/U \cong G_{i+1}$.

Inductive step: compute $Aut(G_{i+1})$ from $Aut(G_i)$

- Compute the p-covering group P of G_i.

- Each automorphism of G_i lifts to an automorphism of P via natural homomorphism $P \to G_i$ with kernel M.

- Identify U such that $P/U \cong G_{i+1}$.

- Let S be the stabiliser of U in $Aut(G_i)$.

Inductive step: compute $Aut(G_{i+1})$ from $Aut(G_i)$

- Compute the p-covering group P of G_i.
- Each automorphism of G_i lifts to an automorphism of P via natural homomorphism $P \to G_i$ with kernel M.
- Identify U such that $P/U \cong G_{i+1}$.
- Let S be the stabiliser of U in $Aut(G_i)$.
- Let A_{i+1} be the subgroup of $Aut(G_{i+1})$ induced by S.

Inductive step: compute $Aut(G_{i+1})$ from $Aut(G_i)$

- Compute the p-covering group P of G_i.

- Each automorphism of G_i lifts to an automorphism of P via natural homomorphism $P \to G_i$ with kernel M.

- Identify U such that $P/U \cong G_{i+1}$.

- Let S be the stabiliser of U in $Aut(G_i)$.

- Let A_{i+1} be the subgroup of $Aut(G_{i+1})$ induced by S.

- $G_{i+1}/\mathcal{P}_i(G_{i+1}) \cong G_i$. Let $T_{i+1} \leq Aut(G_{i+1})$ consisting of those automorphisms which fix G_i and $\mathcal{P}_i(G_{i+1})$ – normal, elementary abelian p-subgroup of $Aut(G_{i+1})$.

Inductive step: compute $Aut(G_{i+1})$ from $Aut(G_i)$

- Compute the p-covering group P of G_i.
- Each automorphism of G_i lifts to an automorphism of P via natural homomorphism $P \to G_i$ with kernel M.
- Identify U such that $P/U \cong G_{i+1}$.
- Let S be the stabiliser of U in $Aut(G_i)$.
- Let A_{i+1} be the subgroup of $Aut(G_{i+1})$ induced by S.
- $G_{i+1}/\mathcal{P}_i(G_{i+1}) \cong G_i$. Let $T_{i+1} \leq Aut(G_{i+1})$ consisting of those automorphisms which fix G_i and $\mathcal{P}_i(G_{i+1})$ – normal, elementary abelian p-subgroup of $Aut(G_{i+1})$.

Theorem

$Aut(G_{i+1}) = A_{i+1} T_{i+1}$.

Where's the problem?

How **difficult** is the inductive step?

Where's the problem?

How **difficult** is the inductive step?

Easily write down a generating set for T_{i+1}.

Where's the problem?

How **difficult** is the inductive step?

Easily write down a generating set for T_{i+1}.

Main task: compute the stabiliser of U in $Aut(G_i)$, where $Aut(G_i)$ acts as a group of automorphisms on M.

Where's the problem?

How **difficult** is the inductive step?

Easily write down a generating set for T_{i+1}.

Main task: compute the stabiliser of U in $Aut(G_i)$, where $Aut(G_i)$ acts as a group of automorphisms on M.

$M = R/[R, F]R^p$ is elementary abelian p-group.

M is an $Aut(G_i)$-module and U is explicit subspace of M.

Action of $A_i := Aut(G_i)$ is explicit as matrix group on M and U is explicit subspace of M.

Action of $A_i := Aut(G_i)$ is explicit as matrix group on M and U is explicit subspace of M.

Task: compute the stabiliser of U under the action of A_i.

Action of $A_i := Aut(G_i)$ is explicit as matrix group on M and U is explicit subspace of M.

Task: compute the stabiliser of U under the action of A_i.

Standard approach: Construct the orbit of U under action of A_i and use standard orbit-stabiliser algorithm to list generators for the stabiliser.

Action of $A_i := Aut(G_i)$ is explicit as matrix group on M and U is explicit subspace of M.

Task: compute the stabiliser of U under the action of A_i.

Standard approach: Construct the orbit of U under action of A_i and use standard orbit-stabiliser algorithm to list generators for the stabiliser.

Central problem: Orbit is frequently too large to construct – and generating set is too large.

Stabiliser under unipotent group

Let A be the automorphism group of a p-group H.

Stabiliser under unipotent group

Let A be the automorphism group of a p-group H.

M is A-module; $U \leq M$.

Stabiliser under unipotent group

Let A be the automorphism group of a p-group H.

M is A-module; $U \leq M$.

$C := C_A(H/\Phi(H))$ is a normal p-subgroup of A: those automorphisms which induce trivial action on $H/\Phi(H)$.

Stabiliser under unipotent group

Let A be the automorphism group of a p-group H.

M is A-module; $U \leq M$.

$C := C_A(H/\Phi(H))$ is a normal p-subgroup of A: those automorphisms which induce trivial action on $H/\Phi(H)$.

Define *canonical copy* of U under action of unipotent group C.

Stabiliser under unipotent group

Let A be the automorphism group of a p-group H.

M is A-module; $U \leq M$.

$C := C_A(H/\Phi(H))$ is a normal p-subgroup of A: those automorphisms which induce trivial action on $H/\Phi(H)$.

Define *canonical copy* of U under action of unipotent group C.

Construct canonical copy, simultaneously write down its stabiliser.

Stabiliser under unipotent group

Let A be the automorphism group of a p-group H.

M is A-module; $U \leq M$.

$C := C_A(H/\Phi(H))$ is a normal p-subgroup of A: those automorphisms which induce trivial action on $H/\Phi(H)$.

Define *canonical copy* of U under action of unipotent group C.

Construct canonical copy, simultaneously write down its stabiliser.

Schwingel, Costi: `Unipotent stabiliser` algorithm.

Stabiliser under unipotent group

Let A be the automorphism group of a p-group H.

M is A-module; $U \leq M$.

$C := C_A(H/\Phi(H))$ is a normal p-subgroup of A: those automorphisms which induce trivial action on $H/\Phi(H)$.

Define *canonical copy* of U under action of unipotent group C.

Construct canonical copy, simultaneously write down its stabiliser.

Schwingel, Costi: `Unipotent stabiliser` algorithm.

If Aut (G_i) is a p-group, then "easy" to compute the stabiliser.

Stabiliser under unipotent group

Let A be the automorphism group of a p-group H.

M is A-module; $U \leq M$.

$C := C_A(H/\Phi(H))$ is a normal p-subgroup of A: those automorphisms which induce trivial action on $H/\Phi(H)$.

Define *canonical copy* of U under action of unipotent group C.

Construct canonical copy, simultaneously write down its stabiliser.

Schwingel, Costi: `Unipotent stabiliser` algorithm.

If Aut (G_i) is a p-group, then "easy" to compute the stabiliser.

Martin and Helleloid (2007): for "most" finite p-groups H, Aut (H) is a p-group.

Stabiliser under unipotent group

Let A be the automorphism group of a p-group H.

M is A-module; $U \leq M$.

$C := C_A(H/\Phi(H))$ is a normal p-subgroup of A: those automorphisms which induce trivial action on $H/\Phi(H)$.

Define *canonical copy* of U under action of unipotent group C.

Construct canonical copy, simultaneously write down its stabiliser.

Schwingel, Costi: `Unipotent stabiliser` algorithm.

If Aut (G_i) is a p-group, then "easy" to compute the stabiliser.

Martin and Helleloid (2007): for "most" finite p-groups H, Aut (H) is a p-group.

Difficult cases: p-groups of small class, particularly class 2.

Preprocessing step

We compute $Aut(G)$ by induction on the lower p-central series.

Preprocessing step

We compute $Aut(G)$ by induction on the lower p-central series.

Initial step: start with $Aut(G_1) \cong GL(d, p)$.

Preprocessing step

We compute $Aut(G)$ by induction on the lower p-central series.

Initial step: start with $Aut(G_1) \cong GL(d, p)$.

Start instead with $L \leq GL(d, p)$ such that the subgroup K of $Aut(G_1)$ induced by $Aut(G)$ is contained in L.

Preprocessing step

We compute $Aut(G)$ by induction on the lower p-central series.

Initial step: start with $Aut(G_1) \cong GL(d, p)$.

Start instead with $L \leq GL(d, p)$ such that the subgroup K of $Aut(G_1)$ induced by $Aut(G)$ is contained in L.

If we can construct L such that $K \leq L < GL(d, p)$, then supply L as input.

Preprocessing step

We compute $Aut(G)$ by induction on the lower p-central series.

Initial step: start with $Aut(G_1) \cong GL(d, p)$.

Start instead with $L \leq GL(d, p)$ such that the subgroup K of $Aut(G_1)$ induced by $Aut(G)$ is contained in L.

If we can construct L such that $K \leq L < GL(d, p)$, then supply L as input.

How to do this? Use characteristic subgroups.

Construct characteristic subgroups of G

Construct characteristic subgroups of G: including G', $Z(G)$, Ω.

Construct characteristic subgroups of G

Construct characteristic subgroups of G: including G', $Z(G)$, Ω.

Restrict this collection to $G_1 = G/\Phi(G)$.

Construct characteristic subgroups of G

Construct characteristic subgroups of G: including G', $Z(G)$, Ω.

Restrict this collection to $G_1 = G/\Phi(G)$.

Obtain list $\mathcal{L}$ of subspaces of $V = GF(p)^d$ which are invariant under $\mathrm{GL}(d, p)$.

Construct characteristic subgroups of G

Construct characteristic subgroups of G: including G', $Z(G)$, Ω.

Restrict this collection to $G_1 = G/\Phi(G)$.

Obtain list $\mathcal{L}$ of subspaces of $V = GF(p)^d$ which are invariant under $\mathrm{GL}(d, p)$.

Now write down the subgroup of $GL(d, p)$ which stabilises each subspace in $\mathcal{L}$.

Construct characteristic subgroups of G

Construct characteristic subgroups of G: including G', $Z(G)$, Ω.

Restrict this collection to $G_1 = G/\Phi(G)$.

Obtain list $\mathcal{L}$ of subspaces of $V = GF(p)^d$ which are invariant under $\mathrm{GL}(d, p)$.

Now write down the subgroup of $GL(d, p)$ which stabilises each subspace in $\mathcal{L}$.

Brooksbank & O'B (2007): construct a system of equations in matrix algebra which must be satisfied by the stabiliser, solve this system to obtain group of units.

Locating characteristic subgroups

Some groups have few characteristic subgroups.
Taunt, Glasby–Pálfy–Schneider: p-groups with unique proper nontrivial characteristic subgroup.

Locating characteristic subgroups

Some groups have few characteristic subgroups.
Taunt, Glasby–Pálfy–Schneider: p-groups with unique proper nontrivial characteristic subgroup.

Wilson (2013), Maglione (2015): new families of characteristic subgroups.

Locating characteristic subgroups

Some groups have few characteristic subgroups.
Taunt, Glasby–Pálfy–Schneider: p-groups with unique proper nontrivial characteristic subgroup.

Wilson (2013), Maglione (2015): new families of characteristic subgroups.

Generalize the N-series of Lazard: new subgroups are located via correspondences with certain graded Lie rings.

Theorem (Maglione, 2015)

Let $S \leq GL(d, q)$ be the group of upper unitriangular matrices. Adjoint refinements of lower central series of S gives a characteristic series of length $\Theta(d^2)$ with factors of order p or p^2.

The hard case

G d-generator p-class 2, exponent p, no known characteristic structure.

The hard case

G d-generator p-class 2, exponent p, no known characteristic structure.

$H = G/\Phi(G)$, and $Aut(H) \cong GL(d, p)$.

The hard case

G d-generator p-class 2, exponent p, no known characteristic structure.

$H = G/\Phi(G)$, and $Aut(H) \cong GL(d, p)$.

Let $V = \mathrm{GF}(p)^d$. Now p-multiplicator M is $V \Lambda V$ and $G = P/U$ where $U \leq M$.

The hard case

G d-generator p-class 2, exponent p, no known characteristic structure.

$H = G/\Phi(G)$, and $Aut(H) \cong GL(d,p)$.

Let $V = \mathrm{GF}(p)^d$. Now p-multiplicator M is $V \Lambda V$ and $G = P/U$ where $U \leq M$.

So $A := GL(d,p)$ acts on the alternating square $\Lambda(V)$.

The hard case

G d-generator p-class 2, exponent p, no known characteristic structure.

$H = G/\Phi(G)$, and $Aut(H) \cong GL(d, p)$.

Let $V = \mathrm{GF}(p)^d$. Now p-multiplicator M is $V \Lambda V$ and $G = P/U$ where $U \leq M$.

So $A := GL(d, p)$ acts on the alternating square $\Lambda(V)$.

The space of alternating forms of degree d on V is naturally isomorphic with the dual vector space $(\Lambda(V))^\star$.

So can identify U with set of bilinear forms.

The (revised) challenge

$A := GL(d, p)$ acts on alternating square representation $\Lambda(V)$; **compute stabiliser of U in A.**

The (revised) challenge

$A := GL(d, p)$ acts on alternating square representation $\Lambda(V)$;
compute stabiliser of U in A.

Assume U has dimension 1: if bilinear form has full rank, stabiliser is $Sp(d, q)$. In all cases, trivial to write down.

The (revised) challenge

$A := GL(d, p)$ acts on alternating square representation $\Lambda(V)$; **compute stabiliser of U in A.**

Assume U has dimension 1: if bilinear form has full rank, stabiliser is $Sp(d, q)$. In all cases, trivial to write down.

1-dimensional spaces partitioned into orbits by rank of form.

The (revised) challenge

$A := GL(d, p)$ acts on alternating square representation $\Lambda(V)$; **compute stabiliser of U in A.**

Assume U has dimension 1: if bilinear form has full rank, stabiliser is $Sp(d, q)$. In all cases, trivial to write down.

1-dimensional spaces partitioned into orbits by rank of form.

Possible strategy for larger dimensional U:

- Basis of U determines set of bilinear forms.
- Construct intersection I of corresponding symplectic groups.
- Normaliser in $GL(d, p)$ of I contains stabiliser.

2-dimensional space

B, M & W (2017): polynomial time algorithm to construct stabiliser of 2-dimensional subspace.

2-dimensional space

B, M & W (2017): polynomial time algorithm to construct stabiliser of 2-dimensional subspace.

Uses a large body of machinery, including: classifications of pairs of forms by Scharlau (1976); projective equivalence under pseudo-isometries developed by Vishnevetski (1980); structure of algebra of adjoints.

2-dimensional space

B, M & W (2017): polynomial time algorithm to construct stabiliser of 2-dimensional subspace.

Uses a large body of machinery, including: classifications of pairs of forms by Scharlau (1976); projective equivalence under pseudo-isometries developed by Vishnevetski (1980); structure of algebra of adjoints.

No classification of orbits.

Graded Algebras

Let K be a finite field. A K-*algebra* A is a K-module equipped with a (possibly nonassociative) K-bilinear product $\circ \colon A \times A \rightarrowtail A$.

Graded Algebras

Let K be a finite field. A K*-algebra* A is a K-module equipped with a (possibly nonassociative) K-bilinear product $\circ \colon A \times A \rightarrowtail A$.

If, as a K-module,

$$A = \bigoplus_{s=0}^{\infty} A_s, \qquad \text{where} \;\; A_s \circ A_t \leq A_{s+t},$$

then A is $\mathbb{N}$-*graded*.

Graded Algebras

Let K be a finite field. A K*-algebra* A is a K-module equipped with a (possibly nonassociative) K-bilinear product $\circ\colon A \times A \rightarrowtail A$.

If, as a K-module,

$$A = \bigoplus_{s=0}^{\infty} A_s, \qquad \text{where} \quad A_s \circ A_t \leq A_{s+t},$$

then A is $\mathbb{N}$*-graded*.

An isomorphism between graded algebras that maps each graded component of one algebra to the corresponding component of the other is a *graded isomorphism*.

Existing uses of graded algebras proceed sequentially through the grading. Starting with the first, consider all possible isomorphisms between corresponding graded components, use the graded product to decide which of them induces an isomorphism between the next components.

Existing uses of graded algebras proceed sequentially through the grading. Starting with the first, consider all possible isomorphisms between corresponding graded components, use the graded product to decide which of them induces an isomorphism between the next components.

Our approach: identify a section of the two graded algebras with *least* number of possible maps.

Existing uses of graded algebras proceed sequentially through the grading. Starting with the first, consider all possible isomorphisms between corresponding graded components, use the graded product to decide which of them induces an isomorphism between the next components.

Our approach: identify a section of the two graded algebras with *least* number of possible maps.

Use practical extension of results of Ivanyos & Qiao to determine which maps between sections lift to isomorphisms of the algebras.

Existing uses of graded algebras proceed sequentially through the grading. Starting with the first, consider all possible isomorphisms between corresponding graded components, use the graded product to decide which of them induces an isomorphism between the next components.

Our approach: identify a section of the two graded algebras with *least* number of possible maps.

Use practical extension of results of Ivanyos & Qiao to determine which maps between sections lift to isomorphisms of the algebras.

Theorem (Brooksbank; O'B; Wilson, 2020)

For each prime p and integer $n > 0$, there is a family of nilpotent matrix Lie algebras of order p^n, containing $p^{O(n^2)}$ non-isomorphic members, for which there is an $O(p^n)$ isomorphism test.

Labelling projective geometry

Exploit graded Lie algebra determined by G.

Labelling projective geometry

Exploit graded Lie algebra determined by G.

Create a labelling which must be preserved under elements of the stabiliser on the points and lines of projective geometry of U.

Labelling projective geometry

Exploit graded Lie algebra determined by G.

Create a labelling which must be preserved under elements of the stabiliser on the points and lines of projective geometry of U.

Example: label each point by rank of associated bilinear form.

Labelling projective geometry

Exploit graded Lie algebra determined by G.

Create a labelling which must be preserved under elements of the stabiliser on the points and lines of projective geometry of U.

Example: label each point by rank of associated bilinear form.

Use labels to define a graph, and construct automorphism group of graph. Lift generators to automorphisms of G.

Labelling projective geometry

Exploit graded Lie algebra determined by G.

Create a labelling which must be preserved under elements of the stabiliser on the points and lines of projective geometry of U.

Example: label each point by rank of associated bilinear form.

Use labels to define a graph, and construct automorphism group of graph. Lift generators to automorphisms of G.

Stabiliser of U is now limited to corresponding subgroup.

Often extremely effective in proving that only scalars stabilise U.