Towards effective computation

Eamonn O’Brien

University of Auckland


August 2009
Geometry following Aschbacher: general strategy

\[ G = \langle X \rangle \leq \text{GL}(d, q). \]

1. Determine (at least one of) its Aschbacher categories.
2. If \( N \triangleleft G \) exists, process \( N \) and \( G/N \) recursively.
3. Otherwise \( G \) is either classical group in natural representation or \( T \leq G/Z \leq \text{Aut}(T) \) where \( T \) is simple.
$G = \langle X \rangle \leq \text{GL}(d, q)$. 

1. Determine (at least one of) its Aschbacher categories.
2. If $N \triangleleft G$ exists, process $N$ and $G/N$ recursively.
3. Otherwise $G$ is either classical group in natural representation or $T \leq G/Z \leq \text{Aut}(T)$ where $T$ is simple.
   - “Reduce” from $G$ to quasisimple group $L$. 

Towards effective computation
Geometry following Aschbacher: general strategy

\[ G = \langle X \rangle \leq \text{GL}(d, q). \]

1. Determine (at least one of) its Aschbacher categories.
2. If \( N \triangleleft G \) exists, process \( N \) and \( G/N \) recursively.
3. Otherwise \( G \) is either classical group in natural representation or \( T \leq G/Z \leq \text{Aut}(T) \) where \( T \) is simple.

- “Reduce” from \( G \) to quasisimple group \( L \).
- Name \( L \).
Geometry following Aschbacher: general strategy

\[ G = \langle X \rangle \leq \text{GL}(d, q). \]

1. Determine (at least one of) its Aschbacher categories.
2. If \( N \triangleleft G \) exists, process \( N \) and \( G/N \) recursively.
3. Otherwise \( G \) is either classical group in natural representation or \( T \leq G/Z \leq \text{Aut}(T) \) where \( T \) is simple.
   - “Reduce” from \( G \) to quasisimple group \( L \).
   - Name \( L \).
   - Set up “effective” isomorphisms between \( L \) and its standard copy \( S \).
The composition tree for $G$

Bäärnhielm, Leedham-Green & O'B
Neunhöffer & Seress
The composition tree for $G$

Bäärnhielm, Leedham-Green & O’B
Neunhöffer & Seress

$H$

$K \overleftarrow{I}$

- Node: section $H$ of $G$. 

Towards effective computation
The composition tree for $G$

Bäärnhielm, Leedham-Green & O’B
Neunhöffer & Seress

$H$

$K \overset{I}{\longrightarrow}$

- Node: section $H$ of $G$.
- Image $I$: image under homomorphism or isomorphism. Images correspond to Aschbacher category, but also others e.g determinant map.
The composition tree for $G$

Bäärnhielm, Leedham-Green & O’B
Neunhöffer & Seress

$H$

$K \overset{I}{\longrightarrow}$

- Node: section $H$ of $G$.
- Image $I$: image under homomorphism or isomorphism. Images correspond to Aschbacher category, but also others e.g determinant map.
- Kernel $K$. 
The composition tree for $G$

Bäärnhielm, Leedham-Green & O’B
Neunhöffer & Seress

$H$

$K \overset{I}{\longrightarrow}$

- Node: section $H$ of $G$.

- Image $I$: image under homomorphism or isomorphism. Images correspond to Aschbacher category, but also others e.g determinant map.

- Kernel $K$.

- Leaf is “composition factor” of $G$: simple modulo scalars. Cyclic not necessarily of prime order.
Tree is constructed in right depth-first order.
Tree is constructed in right depth-first order.

If node $H$ is not a leaf, construct recursively subtree rooted at $I$, then subtree rooted at $K$. 
Tree is constructed in **right depth-first order**.

If node $H$ is not a leaf, construct recursively subtree rooted at $I$, then subtree rooted at $K$.

$$
\begin{array}{c}
H \\
\mid \\
I_1 \\
\end{array}
$$
Tree is constructed in right depth-first order.

If node $H$ is not a leaf, construct recursively subtree rooted at $I$, then subtree rooted at $K$.

```
   H     H
  /     /  \
I1    I1  /  \
 /    |   |
I2    I2
```
Tree is constructed in right depth-first order.

If node $H$ is not a leaf, construct recursively subtree rooted at $I$, then subtree rooted at $K$.

```
        H
       /|
      / |\n     I1 I1 I1
    / |   \|
   I2 K2 I2
```
Tree is constructed in **right depth-first order**.

If node $H$ is not a leaf, construct recursively subtree rooted at $I$, then subtree rooted at $K$. 

```
        H  
       /   
      /    
     H    H
    /     /   
   /     /    
  I1    I1    I1
    |      |     |  
   I2    K2    K1
     |      |      |  
    I2    I2    I1
```
Assume $\phi : H \longrightarrow I$ where $K = \ker \phi$. 

SOMETIME EASY TO OBTAIN THEORETICAL GENERATING SETS FOR $\ker \phi$. 

E.G. SMALLER FIELD, SEMILINEAR, NORMALISER OF SYMPLECTIC-TYPE GROUP. 

OTHERWISE, CONSTRUCT NORMAL GENERATING SET FOR $K$, BY EVALUATING RELATORS IN PRESENTATION FOR $I$ AND TAKE NORMAL CLOSURE. 

SO WE NEED A PRESENTATION FOR $I$. 

TO OBTAIN PRESENTATION FOR NODE: NEED ONLY PRESENTATION FOR ASSOCIATED KERNEL AND IMAGE. 

SO INDUCTIVELY NEED TO KNOW PRESENTATIONS ONLY FOR THE LEAVES – OR COMPOSITION FACTORS.
Assume $\phi : H \rightarrow I$ where $K = \ker \phi$.

$\begin{tikzpicture}
  \node (H) at (0,0) {$H$};
  \node (K) at (-1,-1) {$K$};
  \node (I) at (1,-1) {$I$};
  \draw (H) -- (K); \draw (K) -- (I);
\end{tikzpicture}$

Sometime easy to obtain theoretically generating sets for $\ker \phi$.
E.g. Smaller Field, Semilinear, normaliser of symplectic-type group.

Otherwise, construct normal generating set for $K$, by evaluating relators in presentation for $I$ and take normal closure.

So we need a presentation for $I$.

To obtain presentation for node: need only presentation for associated kernel and image.

So inductively need to know presentations only for the leaves – or composition factors.
Assume $\phi : H \rightarrowtail I$ where $K = \ker \phi$.

Sometime easy to obtain theoretically generating sets for $\ker \phi$.

E.g. Smaller Field, Semilinear, normaliser of symplectic-type group.
Constructing kernels

Assume $\phi : H \longrightarrow I$ where $K = \ker \phi$.

\[
\begin{array}{c}
H \\
\downarrow

K & \longrightarrow & I
\end{array}
\]

Sometime easy to obtain theoretically generating sets for $\ker \phi$. 
e.g. Smaller Field, Semilinear, normaliser of symplectic-type group.

Otherwise, construct normal generating set for $K$, by evaluating
relators in presentation for $I$ and take normal closure.

So we need a presentation for $I$.

To obtain presentation for node: need only presentation for
associated kernel and image.

So inductively need to know presentations only for the leaves
– or composition factors.
Assume $\phi : H \rightarrow I$ where $K = \ker \phi$.

\[
\begin{array}{c}
H \\
\downarrow
\end{array}
\]

\[
\begin{array}{c}
K \\
\downarrow
\end{array} \
I
\]

Sometime easy to obtain theoretically generating sets for $\ker \phi$.
e.g. Smaller Field, Semilinear, normaliser of symplectic-type group.

Otherwise, construct normal generating set for $K$, by evaluating
relators in presentation for $I$ and take normal closure.

So we need a presentation for $I$. 
Constructing kernels

Assume $\phi : H \hookrightarrow I$ where $K = \ker \phi$.

$H$

$K \triangleleft I$

Sometime easy to obtain theoretically generating sets for $\ker \phi$.

e.g. Smaller Field, Semilineal, normaliser of symplectic-type group.

Otherwise, construct normal generating set for $K$, by evaluating relators in presentation for $I$ and take normal closure.

So we need a presentation for $I$.

To obtain presentation for node: need only presentation for associated kernel and image.
Assume $\phi : H \rightarrow I$ where $K = \ker \phi$.

\[ H \xrightarrow{K} I \]

Sometime easy to obtain theoretically generating sets for $\ker \phi$. e.g. Smaller Field, Semilinear, normaliser of symplectic-type group.

Otherwise, construct normal generating set for $K$, by evaluating relators in presentation for $I$ and take normal closure.

So we need a presentation for $I$.

To obtain presentation for node: need only presentation for associated kernel and image.

So inductively need to know presentations only for the leaves – or composition factors.
Babai and Szemerédi (1984): *length* of a presentation $P = \{X \mid R\}$ is number of symbols to write down the presentation.

$S_n$ generated by $t_k = (k, k+1)$ for $1 \leq k < n$ with relations:

\[ \hat{t}_k^2 = 1 \quad \text{for} \quad 1 \leq k < n, \]
\[ \hat{(t_k - 1)t_k}^3 = 1 \quad \text{for} \quad 1 < k < n, \]
\[ \hat{(t_j t_k)}^2 = 1 \quad \text{for} \quad 1 \leq j < k - 1 < n - 1. \]

Number of relations is $n(n - 1)/2$, and presentation length is $O(n^5)$. $S_n$ acts on deleted permutation module: cost of evaluation of relations is $O(n^5)$. 

Eamonn O'Brien
Towards effective computation
Babai and Szemerédi (1984): *length* of a presentation $P = \{X \mid R\}$ is number of symbols to write down the presentation.

Each generator is single symbol, relator is a string of symbols, exponents written in binary.
Babai and Szemerédi (1984): *length* of a presentation $P = \{X \mid R\}$ is number of symbols to write down the presentation.

Each generator is single symbol, relator is a string of symbols, exponents written in binary.

**Example**

$S_n$ generated by $t_k = (k, k+1)$ for $1 \leq k < n$ with relations:

- $t_k^2 = 1$ for $1 \leq k < n$,
- $(t_{k-1}t_k)^3 = 1$ for $1 < k < n$,
- $(t_jt_k)^2 = 1$ for $1 \leq j < k - 1 < n - 1$.

Number of relations is $n(n-1)/2$, and presentation length is $O(n^2)$. 

Babai and Szemerédi (1984): length of a presentation $P = \{X \mid R\}$ is number of symbols to write down the presentation.

Each generator is single symbol, relator is a string of symbols, exponents written in binary.

**Example**

$S_n$ generated by $t_k = (k, k+1)$ for $1 \leq k < n$ with relations:

- $t_k^2 = 1$ for $1 \leq k < n$,
- $(t_{k-1} t_k)^3 = 1$ for $1 < k < n$,
- $(t_j t_k)^2 = 1$ for $1 \leq j < k-1 < n-1$.

Number of relations is $n(n-1)/2$, and presentation length is $O(n^2)$.

$S_n$ acts on deleted permutation module: cost of evaluation of relations is $O(n^5)$. 
Theorem (Guralnick, Kantor, Kassabov, Lubotzky, 2008)

Every non-abelian finite simple group of rank $n$ over $\mathrm{GF}(q)$, with possible exception of Ree groups $^2G_2(q)$, has a presentation with a bounded number of generators and relations and total length $O(\log n + \log q)$.
Theorem (Guralnick, Kantor, Kassabov, Lubotzky, 2008)

Every non-abelian finite simple group of rank $n$ over $\text{GF}(q)$, with possible exception of Ree groups $^2G_2(q)$, has a presentation with a bounded number of generators and relations and total length $O(\log n + \log q)$.

Exploits results of:

- Campbell, Robertson and Williams (1990): $\text{PSL}(2, p^n)$ has presentation on (at most) 3 generators and a bounded number of relations.
- Hulpke and Seress (2003): $\text{PSU}(3, q)$
Previous best: Babai *et al.* (1997) presentation of length $O(\log^2 |G|)$. Modifications of Curtis-Steinberg-Tits presentations for groups of Lie rank at least 2.
Previous best: Babai et al. (1997) presentation of length $O(\log^2 |G|)$. Modifications of Curtis-Steinberg-Tits presentations for groups of Lie rank at least 2.

**Constructive version** (L-G and O’B, ongoing): explicit short presentations for the classical groups on our standard generators. Complete for $\text{SL}, \text{Sp}, \text{SU}$. 
Short presentations for $S_n$ and $A_n$

Theorem (GKKL, 2006; Bray-Conder-LG-O'B, 2006)

$S_n$ and $A_n$ have presentations with a bounded number of generators and relations, and length $O(\log n)$.

Theorem (Bray-Conder-LG-O'B, 2006)

Let $p$ be an odd prime, and let $\lambda$ be a primitive element of $\text{GF}(p)$, with inverse $\mu$. Then

$\{ a, c, t | a^p, ac, ac^2 - 1, (a^{(p+1)/2}c^4)^2, t^2, [t, a], [t, ca\lambda], [t, c]_3, (ttc^2ca)^2, (ttc^2ca\lambda)^2, (act)^{p+1} \}$

is a 3-generator 10-relator presentation of length $O(\log p)$ for $S_{p+2}$, in which $atc$ stands for an $(p+2)$-cycle and $t$ stands for a transposition.
Short presentations for $S_n$ and $A_n$

**Theorem (GKKL, 2006; Bray-Conder-LG-O’B, 2006)**

$A_n$ and $S_n$ have presentations with a bounded number of generators and relations, and length $O(\log n)$. 

Let $p$ be an odd prime, and let $\lambda$ be a primitive element of $\mathbb{GF}(p)$, with inverse $\mu$. Then

\[
\begin{align*}
\{a, c, t | & a^p, acacac - 1,
(a (p+1)/2)c^4c^2, t^2, [t, a], [t, ca^\lambda], [t, c]^3, (ttc^ttca^\lambda)^2, (ttc^ttca^\lambda)^2, (atc)^{p+1}\} \\
\end{align*}
\]

is a 3-generator 10-relator presentation of length $O(\log p)$ for $S_{p+2}$, in which $attc$ stands for a $(p+2)$-cycle and $t$ stands for a transposition.
Short presentations for $S_n$ and $A_n$

**Theorem (GKKL, 2006; Bray-Conder-LG-O’B, 2006)**

$A_n$ and $S_n$ have presentations with a bounded number of generators and relations, and length $O(\log n)$.

**Theorem (Bray-Conder-LG-O’B, 2006)**

Let $p$ be an odd prime, and let $\lambda$ be a primitive element of $\text{GF}(p)$, with inverse $\mu$. Then

$$\{ a, c, t \mid a^p, acacac^{-1}, \left(a^{(p+1)/2}ca^4c\right)^2, t^2, [t, a],$$

$$[t, ca^\lambda ca^\mu c], [t, c]^3, (tt^c tt^{-c}a\lambda)^2, (tt^c tt^{-c}a\lambda)^2, (at^c)^{p+1} \}$$

is a 3-generator 10-relator presentation of length $O(\log p)$ for $S_{p+2}$, in which $att^c$ stands for a $(p+2)$-cycle and $t$ stands for a transposition.
Previous best results: length $O(n \log n)$ (Moore, 1897)

**Theorem (GKKL, 2008)**

$A_n$ has presentation on 3 generators, 4 relations, length $O(\log n)$. 

Problem Is there a $O(\log n)$ presentation for $S_n$ on $(1, 2)$ and $(1, 2, \ldots, n)$ with a uniformly bounded number of relators?
Previous best results: length $O(n \log n)$ (Moore, 1897)

**Theorem (GKKL, 2008)**

$A_n$ has presentation on 3 generators, 4 relations, length $O(\log n)$.

$S_n$: presentation of length $O(n^2)$ on $(1, 2)$ and $(1, 2, \ldots, n)$ and 78 relations.
Previous best results: length $O(n \log n)$ (Moore, 1897)

**Theorem (GKKL, 2008)**

$A_n$ has presentation on 3 generators, 4 relations, length $O(\log n)$.

$S_n$: presentation of length $O(n^2)$ on $(1, 2)$ and $(1, 2, \ldots, n)$ and 78 relations.

**Problem**

*Is there a $O(\log n)$ presentation for $S_n$ on $(1, 2)$ and $(1, 2, \ldots, n)$ with a uniformly bounded number of relators?*
Given \( G = \langle X \rangle \leq \text{GL}(d, q) \) as input.

**Output:**

- a composition series: \( 1 = G_0 \triangleleft G_1 \triangleleft G_2 \cdots \triangleleft G_m = G \).
- A representation \( S_k = \langle X_k \rangle \) of \( G_k/G_{k-1} \)
- Effective maps \( \tau_k : G_k \rightarrow S_k, \phi_k : S_k \rightarrow G_k \)
  \( \tau_k \) epimorphism with kernel \( G_{k-1} \)
- Map to write \( g \in G \) as word in \( X \).
Given $G = \langle X \rangle \leq \text{GL}(d, q)$ as input.

**Output:**

- a composition series: $1 = G_0 \triangleleft G_1 \triangleleft G_2 \cdots \triangleleft G_m = G$.
- A representation $S_k = \langle X_k \rangle$ of $G_k / G_{k-1}$
- Effective maps $\tau_k : G_k \to S_k$, $\phi_k : S_k \to G_k$
  $\tau_k$ epimorphism with kernel $G_{k-1}$
- Map to write $g \in G$ as word in $X$.

Construct presentation for group defined by tree and verify that $G$ satisfies the relations.
$G$ has characteristic series $\mathcal{C}$ of subgroups:

$$1 \leq O_{\infty}(G) \leq S^*(G) \leq P(G) \leq G$$
$G$ has characteristic series $\mathcal{C}$ of subgroups:

$$1 \leq O_\infty(G) \leq S^*(G) \leq P(G) \leq G$$

$O_\infty(G)$ = largest soluble normal subgroup of $G$, soluble radical
Characteristic structure

$G$ has characteristic series $\mathcal{C}$ of subgroups:

$$1 \leq O_\infty(G) \leq S^*(G) \leq P(G) \leq G$$

$O_\infty(G)$ = largest soluble normal subgroup of $G$, soluble radical

$S^*(G)/O_\infty(G) = \text{Socle} \left( G/O_\infty(G) \right) = T_1 \times \ldots \times T_k$ where $T_i$ non-abelian simple

$\phi: G \rightarrow \text{Sym}(k)$ is repn of $G$ induced by conjugation on \{T_1, \ldots, T_k\} and $P(G)/S^*(G) \leq \text{Out}(T_1) \times \ldots \times \text{Out}(T_k)$ and so is soluble

$G/P(G) \leq \text{Sym}(k)$ where $k \leq \log |G|/\log 60$
Characteristic structure

$G$ has characteristic series $\mathcal{C}$ of subgroups:

$$1 \leq O_\infty(G) \leq S^*(G) \leq P(G) \leq G$$

$O_\infty(G) =$ largest soluble normal subgroup of $G$, soluble radical

$S^*(G)/O_\infty(G) =$ Socle $(G/O_\infty(G)) = T_1 \times \ldots \times T_k$ where $T_i$ non-abelian simple

$\phi : G \to \text{Sym}(k)$ is repn of $G$ induced by conjugation on

$\{T_1, \ldots, T_k\}$ and $P(G) = \ker \phi$
$G$ has characteristic series $C$ of subgroups:

$$1 \leq O_\infty(G) \leq S^*(G) \leq P(G) \leq G$$

$O_\infty(G) =$ largest soluble normal subgroup of $G$, soluble radical

$S^*(G)/O_\infty(G) = \text{Socle}(G/O_\infty(G)) = T_1 \times \ldots \times T_k$ where $T_i$ non-abelian simple

$\phi : G \rightarrow \text{Sym}(k)$ is repn of $G$ induced by conjugation on

$\{T_1, \ldots, T_k\}$ and $P(G) = \ker \phi$

$P(G)/S^*(G) \leq \text{Out}(T_1) \times \ldots \times \text{Out}(T_k)$ and so is soluble
Characteristic structure

G has characteristic series $C$ of subgroups:

$$1 \leq O_\infty(G) \leq S^*(G) \leq P(G) \leq G$$

$O_\infty(G)$ = largest soluble normal subgroup of $G$, soluble radical

$S^*(G)/O_\infty(G) =$ Socle $(G/O_\infty(G)) = T_1 \times \ldots \times T_k$ where $T_i$ non-abelian simple

$\phi : G \rightarrow \text{Sym}(k)$ is repn of $G$ induced by conjugation on

$\{T_1, \ldots, T_k\}$ and $P(G) = \ker \phi$

$P(G)/S^*(G) \leq \text{Out}(T_1) \times \ldots \times \text{Out}(T_k)$ and so is soluble

$G/P(G) \leq \text{Sym}(k)$ where $k \leq \log |G| / \log 60$
Black-box model pioneered by Babai and Beals.
Black-box model pioneered by Babai and Beals.

Babai, Beals, Seress (2009): can construct $\mathcal{C}$ directly in black-box groups in polynomial time (subject to Discrete Log solution and some other restrictions).
Black-box model pioneered by Babai and Beals.

Babai, Beals, Seress (2009): can construct $C$ directly in black-box groups in polynomial time (subject to Discrete Log solution and some other restrictions).

Ongoing work with Holt and Roney-Dougal:

- refine composition series obtained from “geometric model” to obtain chief series reflecting this characteristic structure.
- exploit $\text{COMPOSITIONTREE}$ and resulting $C$ as infrastructure for algorithms to solve “real” problems.
Exploiting the characteristic series $C$

Cannon, Holt et al. (2000s): use $C$ in practical algorithms.

$$1 \leq L := O_\infty(G) \leq S^*(G) \leq P(G) \leq G$$
Cannon, Holt et al. (2000s): use $C$ in practical algorithms.

$$1 \leq L := O_\infty(G) \leq S^*(G) \leq P(G) \leq G$$

Also compute series

$$1 = N_0 \triangleleft N_1 \triangleleft \cdots \triangleleft N_r = L \triangleleft G$$

where $N_i \triangleleft G$ and $N_i/N_{i-1}$ is elementary abelian.
Exploiting the characteristic series $\mathcal{C}$

Cannon, Holt et al. (2000s): use $\mathcal{C}$ in practical algorithms.

$$1 \leq L := O_\infty(G) \leq S^*(G) \leq P(G) \leq G$$

Also compute series

$$1 = N_0 \triangleleft N_1 \triangleleft \cdots \triangleleft N_r = L \triangleleft G$$

where $N_i \triangleleft G$ and $N_i/N_{i-1}$ is elementary abelian.

$G/L$ has a trivial Fitting subgroup, so is a TF-group. Framework sometimes called **Trivial Fitting model of computation**.
The TF-model

\[
1 = N_0 \triangleleft N_1 \triangleleft \cdots \triangleleft N_r = L \leq S^*(G) \leq P(G) \leq G
\]

where \( N_i \trianglelefteq G \) and \( N_i/N_{i-1} \) is elementary abelian.
The TF-model

\[ 1 = N_0 \vartriangleleft N_1 \vartriangleleft \cdots \vartriangleleft N_r = L \leq S^*(G) \leq P(G) \leq G \]

where \( N_i \unlhd G \) and \( N_i/N_{i-1} \) is elementary abelian.

Given a problem:
1 = N_0 \triangleleft N_1 \triangleleft \cdots \triangleleft N_r = L \leq S^*(G) \leq P(G) \leq G

where \( N_i \trianglelefteq G \) and \( N_i/N_{i-1} \) is elementary abelian.

Given a problem:

Solve problem first in \( G/L = G/N_r \), and then, successively, solve it in \( G/N_i \), for \( i = r-1, \ldots, 0 \).
1 = N_0 \triangleleft N_1 \triangleleft \cdots \triangleleft N_r = L \leq S^*(G) \leq P(G) \leq G

where \( N_i \trianglelefteq G \) and \( N_i/N_{i-1} \) is elementary abelian.

Given a \textbf{problem}:

Solve problem first in \( G/L = G/N_r \), and then, successively, solve it in \( G/N_i \), for \( i = r - 1, \ldots, 0 \).

\( H := G/L \) is a TF-group.

So \( H \) has a socle \( S \) which is direct product of non-abelian simple groups \( T_i \) and these are permuted under conjugation by \( H \).
The TF-model

1 = N_0 \vartriangleleft N_1 \vartriangleleft \cdots \vartriangleleft N_r = L \leq S^*(G) \leq P(G) \leq G

where \( N_i \vartriangleleft G \) and \( N_i/N_{i-1} \) is elementary abelian.

**Given a problem:**

Solve problem first in \( G/L = G/N_r \), and then, successively, solve it in \( G/N_i \), for \( i = r-1, \ldots, 0 \).

\( H := G/L \) is a TF-group.

So \( H \) has a socle \( S \) which is direct product of non-abelian simple groups \( T_i \) and these are permuted under conjugation by \( H \).

Problem may have nice solution for \( H \).
The TF-model

\[ 1 = N_0 \triangleleft N_1 \triangleleft \cdots \triangleleft N_r = L \leq S^*(G) \leq P(G) \leq G \]

where \( N_i \triangleleft G \) and \( N_i/N_{i-1} \) is elementary abelian.

Given a problem:

Solve problem first in \( G/L = G/N_r \), and then, successively, solve it in \( G/N_i \), for \( i = r-1, \ldots, 0 \).

\( H := G/L \) is a TF-group. 

So \( H \) has a socle \( S \) which is direct product of non-abelian simple groups \( T_i \) and these are permuted under conjugation by \( H \).

Problem may have nice solution for \( H \).

In many cases, easy to reduce the computation for TF-group \( H \) to almost simple groups.
Examples of practical algorithms using TF-model

- Determine conjugacy classes of elements of $G$; (Cannon & Souvignier, 1997)
- Determine maximal subgroups of $G$; (Cannon & Holt, 2004) and (Eick & Hulpke, 2001)
- Determine the automorphism group of $G$; (Cannon & Holt, 2003)
- Determine conjugacy classes of subgroups of $G$; (Cannon, Cox & Holt, 2001)

Most algorithms are representation-independent. Implementations use BSGS and Random Schreier for associated computations: so limited in range. Plan to use CompositionTree for these.
Examples of practical algorithms using TF-model

- Determine conjugacy classes of elements of $G$; (Cannon & Souvignier, 1997)
- Determine maximal subgroups of $G$; (Cannon & Holt, 2004) and (Eick & Hulpke, 2001)
- Determine the automorphism group of $G$; (Cannon & Holt, 2003)
- Determine conjugacy classes of subgroups of $G$; (Cannon, Cox & Holt, 2001)

Most algorithms are representation-independent. Implementations use BSGS and Random Schreier for associated computations; so limited in range. Plan to use CompositionTree for these.
Examples of practical algorithms using TF-model

- Determine conjugacy classes of elements of $G$; (Cannon & Souvignier, 1997)
- Determine maximal subgroups of $G$; (Cannon & Holt, 2004) and (Eick & Hulpke, 2001)

Most algorithms are representation-independent. Implementations use BSGS and Random Schreier for associated computations: so limited in range. Plan to use CompositionTree for these.
Examples of practical algorithms using TF-model

- Determine conjugacy classes of elements of $G$; (Cannon & Souvignier, 1997)
- Determine maximal subgroups of $G$; (Cannon & Holt, 2004) and (Eick & Hulpke, 2001)
- Determine the automorphism group of $G$; (Cannon & Holt, 2003)

Most algorithms are representation-independent. Implementations use BSGS and Random Schreier for associated computations: so limited in range. Plan to use CompositionTree for these.
Examples of practical algorithms using TF-model

- Determine conjugacy classes of elements of $G$; (Cannon & Souvignier, 1997)
- Determine maximal subgroups of $G$; (Cannon & Holt, 2004) and (Eick & Hulpke, 2001)
- Determine the automorphism group of $G$; (Cannon & Holt, 2003)
- Determine conjugacy classes of subgroups of $G$; (Cannon, Cox & Holt, 2001)

Most algorithms are representation-independent. Implementations use BSGS and Random Schreier for associated computations: so limited in range. Plan to use CompositionTree for these.
Examples of practical algorithms using TF-model

- Determine conjugacy classes of elements of $G$; (Cannon & Souvignier, 1997)
- Determine maximal subgroups of $G$; (Cannon & Holt, 2004) and (Eick & Hulpke, 2001)
- Determine the automorphism group of $G$; (Cannon & Holt, 2003)
- Determine conjugacy classes of subgroups of $G$; (Cannon, Cox & Holt, 2001)

Most algorithms are representation-independent.
Examples of practical algorithms using TF-model

- Determine conjugacy classes of elements of $G$; (Cannon & Souvignier, 1997)
- Determine maximal subgroups of $G$; (Cannon & Holt, 2004) and (Eick & Hulpke, 2001)
- Determine the automorphism group of $G$; (Cannon & Holt, 2003)
- Determine conjugacy classes of subgroups of $G$; (Cannon, Cox & Holt, 2001)

Most algorithms are representation-independent.

Implementations use BSGS and Random Schreier for associated computations: so limited in range.

Plan to use **CompositionTree** for these.
Almost simple groups: Conjugacy classes

Wall (1963): description of conjugacy classes and centralisers of elements of classical groups.

Murray & Haller (ongoing): algorithms, which given $d$ and $q$, constructs classes for $\text{S}X(d,q) \leq K \leq \text{C}X(d,q)$.

Constructive recognition: provides $\phi: K \mapsto \overrightarrow{\bar{K}}$.

Embed TF-group $H = G/L$ in direct product $W$ of $T_i \wr \text{Sym}(d_i)$, where $T_i$ occurs $d_i$ times as socle factor.

Conjugacy class representatives in wreath products described theoretically (Hulpke 2004; Cannon & Holt, 2006).

Eamonn O'Brien
Towards effective computation
Almost simple groups: Conjugacy classes

Wall (1963): description of conjugacy classes and centralisers of elements of classical groups.
Almost simple groups: Conjugacy classes

Wall (1963): description of conjugacy classes and centralisers of elements of classical groups.

Murray & Haller (ongoing): algorithms, which given $d$ and $q$, constructs classes for $\text{SX}(d, q) \leq K \leq \text{CX}(d, q)$.
Almost simple groups: Conjugacy classes

Wall (1963): description of conjugacy classes and centralisers of elements of classical groups.

Murray & Haller (ongoing): algorithms, which given $d$ and $q$, constructs classes for $SX(d, q) \leq K \leq CX(d, q)$.

Constructive recognition: provides $\phi : K \mapsto \bar{K}$.
Almost simple groups: Conjugacy classes

Wall (1963): description of conjugacy classes and centralisers of elements of classical groups.

Murray & Haller (ongoing): algorithms, which given $d$ and $q$, constructs classes for $SX(d, q) \leq K \leq CX(d, q)$.

Constructive recognition: provides $\phi : K \mapsto \bar{K}$.

Embed TF-group $H = G/L$ in direct product $W$ of $T_i \wr \text{Sym}(d_i)$, where $T_i$ occurs $d_i$ times as socle factor.
Almost simple groups: Conjugacy classes

Wall (1963): description of conjugacy classes and centralisers of elements of classical groups.

Murray & Haller (ongoing): algorithms, which given $d$ and $q$, constructs classes for $SX(d, q) \leq K \leq CX(d, q)$.

Constructive recognition: provides $\phi : K \mapsto \bar{K}$.

Embed TF-group $H = G/L$ in direct product $W$ of $T_i \wr \text{Sym}(d_i)$, where $T_i$ occurs $d_i$ times as socle factor.

Conjugacy class representatives in wreath products described theoretically (Hulpke 2004; Cannon & Holt, 2006).
Example: Automorphism group of $G$

Cannon & Holt, 2003

$H := G/L$ permutes the direct factors of its socle $S$ by conjugation.
Cannon & Holt, 2003

$H := G/L$ permutes the direct factors of its socle $S$ by conjugation.

Embed $H$ in direct product $D$ of $\text{Aut}(T_i) \wr \text{Sym}(d_i)$, where $T_i$ occurs $d_i$ times as socle factor of $S$. 
Cannon & Holt, 2003

$H := G/L$ permutes the direct factors of its socle $S$ by conjugation.

Embed $H$ in direct product $D$ of $\text{Aut}(T_i) \wr \text{Sym}(d_i)$, where $T_i$ occurs $d_i$ times as socle factor of $S$.

$\text{Aut}(H)$ is normaliser of the image of $H$ in $D$. 
Example: Automorphism group of $G$

Cannon & Holt, 2003

$H := G/L$ permutes the direct factors of its socle $S$ by conjugation.

Embed $H$ in direct product $D$ of $\text{Aut}(T_i) \wr \text{Sym}(d_i)$, where $T_i$ occurs $d_i$ times as socle factor of $S$.

$\text{Aut}(H)$ is normaliser of the image of $H$ in $D$.

Now lift results through elementary abelian layers, computing $\text{Aut}(G/N_i)$ successively.
Suppose $N \leq M \leq G$, where both $M, N$ char in $G$ and $M/N$ is elementary abelian of order $p^d$. 

\[ G \]
\[ M \]
\[ N \]
Suppose $N \leq M \leq G$, where both $M, N$ char in $G$ and $M/N$ is elementary abelian of order $p^d$.

Suppose $A_M = \text{Aut}(G/M)$ is known.
Suppose $N \leq M \leq G$, where both $M, N$ char in $G$ and $M/N$ is elementary abelian of order $p^d$.

• $G$

Suppose $A_M = \text{Aut}(G/M)$ is known.

All automorphisms of $G$ fix both $M$ and $N$. 

• $M$

• $N$
Suppose \( N \leq M \leq G \), where both \( M, N \) char in \( G \) and \( M/N \) is elementary abelian of order \( p^d \).

Suppose \( A_M = \text{Aut}(G/M) \) is known.

All automorphisms of \( G \) fix both \( M \) and \( N \).

\( A_N = \text{Aut}(G/N) \) has normal subgroups \( C \leq B \).

\( B \) induces identity on \( G/M \).

\( C \) induces identity on both \( G/M \) and \( M/N \).
$M/N$ is $\mathbb{F}_p(G/M)$-module.

- $A_N$
- $B$
- $C$
$M/N$ is $\mathbb{F}_p(G/M)$-module.

- Elements of $C$ correspond to derivations from $G/M$ to $M/N$. 

Eamonn O'Brien
  Towards effective computation
$M/N$ is $\mathbb{F}_p(G/M)$-module.

- Elements of $C$ correspond to derivations from $G/M$ to $M/N$.
- Elements of $B/C$ correspond to module automorphisms of $M/N$. Can choose $M$ and $N$ to ensure that these tasks “easy”.

Hardest task: determine $S \leq A_M$ which lifts to $G/N$.

$S \leq A'_M$, subgroup of $A_M$ whose elements preserve the isomorphism type of module $M/N$.

If so, all elements of $A'_M$ lift.

Otherwise, must test each element of $A'_M$ for lifting.
$M/N$ is $\mathbb{F}_p(G/M)$-module.

- Elements of $C$ correspond to derivations from $G/M$ to $M/N$.

- Elements of $B/C$ correspond to module automorphisms of $M/N$. Can choose $M$ and $N$ to ensure that these tasks “easy”.

- Hardest task: determine $S \leq A_M$ which lifts to $G/N$. $S \leq A'$, subgroup of $A_M$ whose elements preserve the isomorphism type of module $M/N$. 

Eamonn O'Brien  
Towards effective computation
\( M/N \) is \( \mathbb{F}_p(G/M) \)-module.

- Elements of \( C \) correspond to derivations from \( G/M \) to \( M/N \).
- Elements of \( B/C \) correspond to module automorphisms of \( M/N \). Can choose \( M \) and \( N \) to ensure that these tasks “easy”.
- Hardest task: determine \( S \leq A_M \) which lifts to \( G/N \). \( S \leq A' \), subgroup of \( A_M \) whose elements preserve the isomorphism type of module \( M/N \).

\( G/N \) split extension of \( M/N \) by \( G/M \)?

If so, all elements of \( A' \) lift.

Otherwise, must test each element of \( A' \) for lifting.
Problem

*Find the order of* $H \leq \text{GL}(6, 5^2)$.
### Problem

*Find the order of \( H \leq \text{GL}(6, 5^2) \).*

... using either of GAP or MAGMA.
**Challenge problems**

**Problem**

*Find the order of $H \leq \text{GL}(6, 5^2)$.*

... using either of GAP or Magma. Good progress, in practice.
### Challenge problems

#### Problem

*Find the order of* $H \leq \text{GL}(6, 5^2)$.

... using either of GAP or Magma. *Good progress, in practice.*

#### Problem

*Given* $g \in \text{GL}(6, 5^2)$ *find its order.*
Challenge problems

Problem

Find the order of $H \leq \text{GL}(6, 5^2)$.

... using either of GAP or MAGMA. Good progress, in practice.

Problem

Given $g \in \text{GL}(6, 5^2)$ find its order.

Yes, in practice.
Challenge problems

Problem

*Find the order of* $H \leq \text{GL}(6, 5^2)$.

... using either of GAP or Magma. Good progress, in practice.

Problem

*Given* $g \in \text{GL}(6, 5^2)$ *find its order.*

Yes, in practice.

Problem

*Find the normaliser in* $\text{GL}(8, 3)$ *of a subgroup of moderate index.*

Not yet ...
Detinko & Flannery (2000s):
\( G \leq \text{GL}(d, R) \) where \( R \) is infinite domain, including \( \mathbb{Z}, \mathbb{Q} \), number fields, function fields.
Decide finiteness, nilpotency, primitivity etc.
“Groups, Representations, Number Theory”

- Hanmer Springs (near Christchurch)
- January 3-10, 2010
- NZMRI Summer meeting feature short lecture courses by:
  - Martin Bridson
  - Michel Broue
  - Persi Diaconis
  - Roger Howe
  - Gus Lehrer
  - Marcus du Sautoy

Organisers: Ben Martin and Eamonn O’Brien