# Centralizer and Normalizer in Matrix Groups

Alexander Hulpke

Department of Mathematics
Colorado State University
Fort Collins, CO, 80523-1874

7/09

# Assumptions and Goal

- A matrix group $G$ and a subgroup $H \leq G$, both given by generators.
- A composition tree, refining a chief series $G = G_0 > G_1 > \cdots > G_k = \langle 1 \rangle$ ($G_i \lhd G$) through $\mathrm{Soc}^*(G)$ and $O_\infty(G)$.
- Effective homomorphisms $\varphi_i \colon G_i \to G_i/G_{i+1}$.
- Generators for $H \cap G_i$. (Presentation for $H^{\varphi_i}$.)

Compute (generators for) $N_G(H)$. Utilize existing algorithms.

Conjugacy classes, (maximal) subgroups, automorphism group algorithms actually transfer **easier** from permutation groups.

# Centralizer / Element Conjugacy

- Work down composition series (Mecky&Neubüser, 1989).
- Radical factor remains: Large part is conjugacy under direct product.
- If GL: Module automorphism/isomorphism (Smith, 1994; Leedham-Green/Steel)
- Backtrack (Sims, 1969)
- Element orbits (Cooperman, 2009)

# Normalizer (in practice) for permutation groups

# Normalizer (in practice) for permutation groups


Backtrack just works too well for permutation groups as a generic algorithm for alternatives to be used.
This approach is not feasible for matrix groups.

# Existing Normalizer Methods

Orbit/Stabilizer Schreier (1927)

Backtrack for permutation groups Sims (1969), Butler&Cannon (1985), Leon (1996), Theißen (1997)

Lifting for solvable groups Glasby& Slattery (1990)

Lifting for permutation groups Luks&Miyazaki (2002), Bounded nonabelian composition factors for polynomial complexity.

Reduction from $S_n$ or $GL_n(q)$ to subgroups Roney-Dougal (2004), H. (2005), Miyamoto (2006+)

# Reusing existing algorithms for matrix groups

All methods do at some point a stabilizer computation. The cost of this is proportional to the stabilizer index.

The basic idea therefore is to reduce down from $G$ to smaller subgroups.
In a classical group we can take appropriate maximal subgroups.

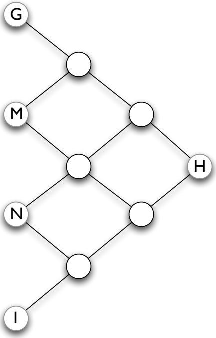
The basic reduction uses normal subgroups: If $M \lhd G$, then

Subgroup reduction $N_G(H) \leq N_G(H \cap M)$

Factor reduction $M \cdot N_G(H)/M \leq N_{G/M}(MH/M)$

# In the context of a series

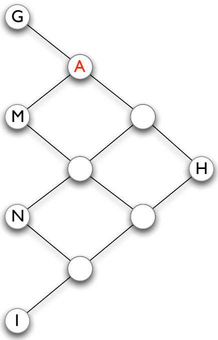
Following Glasby/Slattery, we iteratively normalize all subgroups we can obtain from $H$ by intersection and closure with groups in the series.



# In the context of a series

Following Glasby/Slattery, we iteratively normalize all subgroups we can obtain from $H$ by intersection and closure with groups in the series.

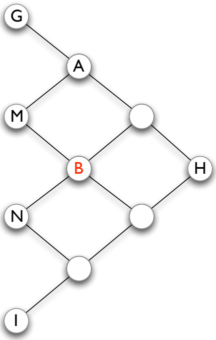
First normalize in a factor group $N_{G/M}(HM/M)$.



# In the context of a series

Following Glasby/Slattery, we iteratively normalize all subgroups we can obtain from *H* by intersection and closure with groups in the series.

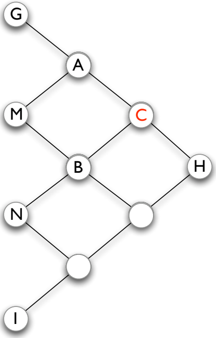
Then normalize the intersection with the normal subgroup



# In the context of a series

Following Glasby/Slattery, we iteratively normalize all subgroups we can obtain from *H* by intersection and closure with groups in the series.

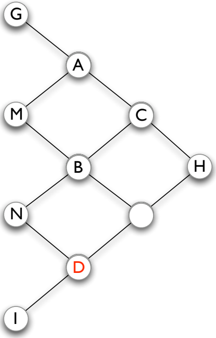
Use the subgroup reduction



# In the context of a series

Following Glasby/Slattery, we iteratively normalize all subgroups we can obtain from $H$ by intersection and closure with groups in the series.

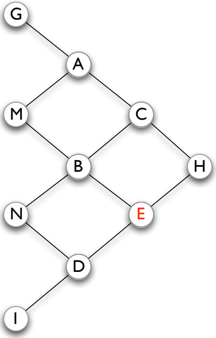
If one factor layer is done, go to the next layer. This is a "down" step



# In the context of a series

Following Glasby/Slattery, we iteratively normalize all subgroups we can obtain from $H$ by intersection and closure with groups in the series.

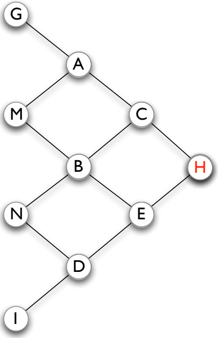
Repeat “up” steps in each layer



# In the context of a series

Following Glasby/Slattery, we iteratively normalize all subgroups we can obtain from *H* by intersection and closure with groups in the series.

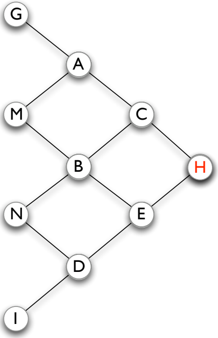
Eventually *H* is normalized.



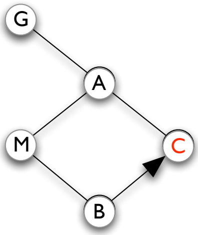
# In the context of a series

Following Glasby/Slattery, we iteratively normalize all subgroups we can obtain from *H* by intersection and closure with groups in the series.

Eventually *H* is normalized. We "only" need to understand the **Up** and **Down** steps



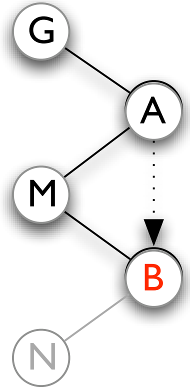
# The Up step




We have normalized subgroups $A, M, B \lhd G$, as well as a complement $C$ to $M/B$ in $A/B$. We now need to normalize $C$.
If $M/B$ is abelian (it is a factor of a composition factor of $G$), complements are described by 1-cocyles, and we just need to stabilize this cocycle (in an affine action).

Because $M \lhd G$, the $M$-orbits form blocks. (Abelian case: 1-coboundaries.)
If $[A : M]$ is coprime to $[M : B]$, there is only one class of complements, and one can get $N(C)$ from $N(A)$ by "generator correction" (if we can determine $M$-conjugacy) or as Sylow subgroup.

# The Down Step




We need to normalize $B \leq M$. As normalization of $A$ before was only in the factor group, we cannot use any of the information obtained before.

If $M/N$ is abelian, $B/N$ is a subspace that needs to be stabilized. This is where the complexity analysis for permutation groups requires bounded composition factors.

$M$-orbits do not help if $M/N$ is abelian.

# Beyond the reductions

In the radical factor, as well as in the down step we are left with the situation to normalize a (possibly comparatively small) subgroup $B$.
Using the subgroup reduction, we can replace $B$ by a characteristic subgroup of $B$. WLOG: $B$ is elementary.

After using invariant geometry/combinatorics, we are left with the backtrack/stabilizer situation.

However, if $B$ is comparatively small, we can try to use this to our advantage (Generalizing what has been done for long time for normalizing cyclic groups).

# Using the automorphism group

Because $B$ is WLOG elementary, we can write down $\mathrm{Aut}(B)$. For small $B$, this group is much smaller than $G$. The following approach (H., 2008) replaces a search through $G$ by a search through $\mathrm{Aut}(B)$.

- We assume that we can find $C_G(B)$ and conjugating elements $x$ for $B^x = D$. (If $G = \mathrm{GL}$ this **is** cheap.)
- $N_G(B)$ induces automorphisms of $B$. The kernel of the map $\varphi\colon N_G(B) \to \mathrm{Aut}(B)$ is $C_G(B)$.
- Finding preimages under $\varphi$ is equivalent to finding conjugating elements. This can be used constructively to test for $\mathrm{Image}(\varphi)$ in a backtrack search.
- If we have $\mathrm{Image}(\varphi)$, we can reconstruct $N_G(B)$.
- Only need representatives for outer automorphisms of $B$.

# Finding Image($\varphi$)

A pure backtrack search for Image($\varphi$) $\leq$ Aut($B$) is still expensive.

We can try to partition the elements of $B$ into classes under $G$-conjugacy, finer than those given by Aut($B$).
Criteria include:

- Orders of preimages (if working in factor group).
- Cycle structures (if permutation representation).
- Normal Form (if matrix group).
- $G$-class (if easily identified).
- The classes in which products fall can be used for further refinement.

For permutation groups this works well, using cycle structures.

# Practical issues that will come up when implementing

The following issues are conveniently ignored by a theoretical description of an inductive process:

Usable composition tree for matrix groups Including leaves.

Re-Use of the composition tree When descending from $G$ to a subgroup $S \leq G$, we need a composition tree for $S$. How does one do this efficiently, without too much recomputation and redundant storage?

Number of Generators Stabilizer computations form Schreier generators, their number goes up proportional to the subgroups index: Subgroup element test, random subproducts.