

The Involution Jumper

Max Neunhöffer



University of St Andrews

GCC09, Perth, 9.1.2009

on the occasion of Cheryl Praeger's 60th birthday

Starting point

Problem

Let $1 < N \triangleleft G = \langle g_1, \dots, g_k \rangle$ be a finite group with order oracle and N be a normal subgroup.

Starting point

Problem

Let $1 < N \triangleleft G = \langle g_1, \dots, g_k \rangle$ be a *finite group with order oracle* and N be a *normal subgroup*.

Produce a non-trivial element of N *as a word in the g_i*

Starting point

Problem

Let $1 < N \triangleleft G = \langle g_1, \dots, g_k \rangle$ be a *finite group with order oracle* and N be a *normal subgroup*.

Produce a non-trivial element of N *as a word in the g_i* with “*high probability*”.

Starting point

Problem

Let $1 < N \triangleleft G = \langle g_1, \dots, g_k \rangle$ be a *finite group with order oracle* and N be a *normal subgroup*.

Produce a non-trivial element of N *as a word in the g_i* with “*high probability*”.

- We are looking for a **randomised algorithm**.

Starting point

Problem

Let $1 < N \triangleleft G = \langle g_1, \dots, g_k \rangle$ be a *finite group with order oracle* and N be a *normal subgroup*.

Produce a non-trivial element of N *as a word in the g_i* with “*high probability*”.

- We are looking for a **randomised algorithm**.
- Assume we can generate **uniformly distributed random elements** in G .

Starting point

Problem

Let $1 < N \triangleleft G = \langle g_1, \dots, g_k \rangle$ be a finite group with order oracle and N be a normal subgroup.

Produce a non-trivial element of N as a word in the g_i with “high probability”.

- We are looking for a randomised algorithm.
- Assume we can generate uniformly distributed random elements in G .
- “High probability” means for the moment “higher than $1/[G : N]$ ”.

Starting point

Problem

Let $1 < N \triangleleft G = \langle g_1, \dots, g_k \rangle$ be a finite group with order oracle and N be a normal subgroup.

Produce a non-trivial element of N as a word in the g_i with “high probability”.

- We are looking for a **randomised algorithm**.
- Assume we can generate **uniformly distributed random elements** in G .
- “High probability” means **for the moment** “higher than $1/[G : N]$ ”.
- Assume **no more knowledge** about G or N .

Starting point

Problem

Let $1 < N \triangleleft G = \langle g_1, \dots, g_k \rangle$ be a finite group with order oracle and N be a normal subgroup.

Produce a non-trivial element of N as a word in the g_i with “high probability”.

- We are looking for a **randomised algorithm**.
- Assume we can generate **uniformly distributed random elements** in G .
- “High probability” means **for the moment** “higher than $1/[G : N]$ ”.
- Assume **no more knowledge** about G or N .
- I shall tell you later why we want to do this.

The Involution
Jumper

Max Neunhöffer

What is the Involution Jumper?

Initial question

Involution Jumper

What's that?

Jumping classes

Back to our question

Problems

Applications

What is the Involution Jumper?

Input: $G = \langle g_1, \dots, g_k \rangle$ and an involution $x \in G$.
repeat

$y := \text{RANDOM}(G)$

$c := x^{-1}y^{-1}xy$ **and** $o := \text{ORDER}(c)$

if o **is even** **then**

return $c^{o/2}$

else

$z := y \cdot c^{(o-1)/2}$ **and** $o' := \text{ORDER}(z)$

if o' **is even** **then**

return $z^{o'/2}$

until patience lost

return FAIL

What is the Involution Jumper?

Input: $G = \langle g_1, \dots, g_k \rangle$ and an involution $x \in G$.
repeat

$y := \text{RANDOM}(G)$

$c := x^{-1}y^{-1}xy$ **and** $o := \text{ORDER}(c)$

if o **is even** **then**

return $c^{o/2}$

else

$z := y \cdot c^{(o-1)/2}$ **and** $o' := \text{ORDER}(z)$

if o' **is even** **then**

return $z^{o'/2}$

until patience lost

return FAIL

Note: If $xy = yx$ then $c = 1_G$ and $o = 1$ and $z = y$.

What is the Involution Jumper?

Input: $G = \langle g_1, \dots, g_k \rangle$ and an involution $x \in G$.
repeat

$y := \text{RANDOM}(G)$

$c := x^{-1}y^{-1}xy$ **and** $o := \text{ORDER}(c)$

if o **is even** **then**

return $c^{o/2}$

else

$z := y \cdot c^{(o-1)/2}$ **and** $o' := \text{ORDER}(z)$

if o' **is even** **then**

return $z^{o'/2}$

until patience lost

return FAIL

Note: If $xy = yx$ then $c = 1_G$ and $o = 1$ and $z = y$.

But this happens rarely.

What does the Involution Jumper do?

Input: $G = \langle g_1, \dots, g_k \rangle$ and an involution $x \in G$.

- **If it does not fail**, it returns an **involution** $\tilde{x} \in G$.

What does the Involution Jumper do?

Input: $G = \langle g_1, \dots, g_k \rangle$ and an involution $x \in G$.

- **If it does not fail**, it returns an **involution** $\tilde{x} \in G$.
- $x\tilde{x} = \tilde{x}x$

What does the Involution Jumper do?

Input: $G = \langle g_1, \dots, g_k \rangle$ and an involution $x \in G$.

- **If it does not fail**, it returns an **involution** $\tilde{x} \in G$.
- $x\tilde{x} = \tilde{x}x$
- Every involution of $C_G(x)$ occurs **with probability** > 0 .

What does the Involution Jumper do?

Input: $G = \langle g_1, \dots, g_k \rangle$ and an involution $x \in G$.

- If it does not fail, it returns an involution $\tilde{x} \in G$.
- $x\tilde{x} = \tilde{x}x$
- Every involution of $C_G(x)$ occurs with probability > 0 .
- Using product replacement to produce random elements, this is a practical method for
 - permutation groups,
 - matrix groups and
 - projective groups,

What does the Involution Jumper do?

Input: $G = \langle g_1, \dots, g_k \rangle$ and an involution $x \in G$.

- **If it does not fail**, it returns an **involution** $\tilde{x} \in G$.
 - $x\tilde{x} = \tilde{x}x$
 - Every involution of $C_G(x)$ occurs **with probability** > 0 .
 - Using **product replacement** to produce random elements, this is **a practical method** for
 - permutation groups,
 - matrix groups and
 - projective groups,
- if nothing goes wrong.**

What does the Involution Jumper do?

Input: $G = \langle g_1, \dots, g_k \rangle$ and an involution $x \in G$.

- **If it does not fail**, it returns an **involution** $\tilde{x} \in G$.
 - $x\tilde{x} = \tilde{x}x$
 - Every involution of $C_G(x)$ occurs **with probability** > 0 .
 - Using **product replacement** to produce random elements, this is **a practical method** for
 - permutation groups,
 - matrix groups and
 - projective groups,
- if nothing goes wrong.**
- It needs **an involution to start with.**

What does the Involution Jumper do?

Input: $G = \langle g_1, \dots, g_k \rangle$ and an involution $x \in G$.

- **If it does not fail**, it returns an **involution** $\tilde{x} \in G$.
 - $x\tilde{x} = \tilde{x}x$
 - Every involution of $C_G(x)$ occurs **with probability** > 0 .
 - Using **product replacement** to produce random elements, this is **a practical method** for
 - permutation groups,
 - matrix groups and
 - projective groups,
- if nothing goes wrong.**
- It needs **an involution to start with**.
 - It needs the **order oracle** desperately.

Jumping between classes

Notation: Let x^G denote the conjugacy class of x in G .

Initial question

Involution Jumper

What's that?

Jumping classes

Back to our question

Problems

Applications

Jumping between classes

Notation: Let x^G denote the conjugacy class of x in G .

Lemma

Let $x, a \in G$ be involutions and $g \in G$. Then

$$\text{Prob}(IJ(x) \in a^G) = \text{Prob}(IJ(x^g) \in a^G).$$

Jumping between classes

Notation: Let x^G denote the **conjugacy class of x in G** .

Lemma

Let $x, a \in G$ be involutions and $g \in G$. Then

$$\text{Prob}(IJ(x) \in a^G) = \text{Prob}(IJ(x^g) \in a^G).$$

or equivalently

Lemma

*Let $x \in G$ be an involution. Then the distribution of $IJ(x)^G$ only depends on x^G and **not on the choice of x in x^G** .*

Jumping between classes

Notation: Let x^G denote the **conjugacy class of x in G** .

Lemma

Let $x, a \in G$ be involutions and $g \in G$. Then

$$\text{Prob}(IJ(x) \in a^G) = \text{Prob}(IJ(x^g) \in a^G).$$

or equivalently

Lemma

Let $x \in G$ be an involution. Then the distribution of $IJ(x)^G$ only depends on x^G and **not on the choice of x in x^G** .

Proof: $f(x, y) :=$

$$\begin{cases} [x, y]^k & \text{if } \text{ORDER}([x, y]) = 2k \\ (y[x, y]^k)' & \text{if } \text{ORDER}([x, y]) = 2k + 1 > 1 \text{ and} \\ & \text{ORDER}([y[x, y]^k]) = 2l \\ y^k & \text{if } xy = yx \text{ and } \text{ORDER}(y) = 2k \end{cases}$$

and we have $f(x^g, y^g) = f(x, y)^g$ whenever f is defined.

A Markov chain \mathcal{M}

The **states** are the **conjugacy classes of involutions in G** .

A Markov chain \mathcal{M}

The **states** are the **conjugacy classes of involutions in G** .

The **transition** is done as follows: At a class a^G :

- **Pick** an arbitrary involution $x \in a^G$.
- **Compute** $\tilde{x} := IJ(x)$ **until** $\tilde{x} \neq \text{FAIL}$.
- **Next state** is \tilde{x}^G .

A Markov chain \mathcal{M}

The **states** are the **conjugacy classes of involutions in G** .

The **transition** is done as follows: At a class a^G :

- **Pick** an arbitrary involution $x \in a^G$.
- **Compute** $\tilde{x} := IJ(x)$ **until** $\tilde{x} \neq \text{FAIL}$.
- **Next state** is \tilde{x}^G .

By the lemma, the **distribution** of the class \tilde{x}^G **does not depend on the choice of x** .

A Markov chain \mathcal{M}

The **states** are the **conjugacy classes of involutions in G** .

The **transition** is done as follows: At a class a^G :

- **Pick** an arbitrary involution $x \in a^G$.
- **Compute** $\tilde{x} := IJ(x)$ **until** $\tilde{x} \neq \text{FAIL}$.
- **Next state** is \tilde{x}^G .

By the lemma, the **distribution** of the class \tilde{x}^G **does not depend on the choice of x** .

Theorem

*The above Markov chain \mathcal{M} is **irreducible** and **aperiodic** and thus has a **stationary distribution** in which every state has non-zero probability.*

Back to the original question

Problem

Let $1 < N \triangleleft G = \langle g_1, \dots, g_k \rangle$ be a finite group with order oracle and N be a normal subgroup.

Produce a non-trivial element of N as a word in the g_i with “high probability”.

Back to the original question

Problem

Let $1 < N \triangleleft G = \langle g_1, \dots, g_k \rangle$ be a *finite group with order oracle* and N be a *normal subgroup*.

Produce a non-trivial element of N **as a word in the g_i** with *“high probability”*.

- **If** we find an involution in G to start with
- **and** N contains at least one involution class,

Back to the original question

Problem

Let $1 < N \triangleleft G = \langle g_1, \dots, g_k \rangle$ be a finite group with order oracle and N be a normal subgroup.

Produce a non-trivial element of N as a word in the g_i with “high probability”.

- If we find an involution in G to start with
 - and N contains at least one involution class,
- the IJ will eventually jump onto an involution class in N .

Back to the original question

Problem

Let $1 < N \triangleleft G = \langle g_1, \dots, g_k \rangle$ be a finite group with order oracle and N be a normal subgroup.

Produce a non-trivial element of N **as a word in the g_i** with “high probability”.

- If we find an involution in G to start with
- **and** N contains at least one involution class,

the IJ will eventually jump onto an involution class in N .

In practice, this works extremely well in many cases:

G	N	# hops*
$S_5 \wr S_{10}$	$S_5^{\times 10}$	1.91
$GL(3, 3) \wr S_6 < GL(18, 3)$	$GL(3, 3)^{\times 6}$	1.17
$Sp(6, 3) \otimes 2.O(7, 3) < GL(48, 3)$	$Sp(6, 3) \otimes 1$	1.83

* average number of IJ hops needed to reach N .

Possible problems

The **whole method is in trouble**, if at least one of the following happens:

- we **do not easily find an involution** in G
(like for example in $SL(2, 2^n)$ for big n),

Possible problems

The **whole method is in trouble**, if at least one of the following happens:

- we **do not easily find an involution** in G
(like for example in $SL(2, 2^n)$ for big n),
- the involution classes of N have a **small probability in the limit distribution**
(when does this happen?),

Possible problems

The **whole method is in trouble**, if at least one of the following happens:

- we **do not easily find an involution** in G
(like for example in $SL(2, 2^n)$ for big n),
- the involution classes of N have a **small probability in the limit distribution**
(when does this happen?),
- the Markov chain **does not converge quick enough** to its limiting distribution
(how quick does it converge?),

Possible problems

The **whole method is in trouble**, if at least one of the following happens:

- we **do not easily find an involution** in G (like for example in $SL(2, 2^n)$ for big n),
- the involution classes of N have a **small probability in the limit distribution** (when does this happen?),
- the Markov chain **does not converge quick enough** to its limiting distribution (how quick does it converge?),
- the **Involution Jumper returns FAIL too often** (when does this happen?),

Possible problems

The **whole method is in trouble**, if at least one of the following happens:

- we **do not easily find an involution** in G (like for example in $SL(2, 2^n)$ for big n),
- the involution classes of N have a **small probability in the limit distribution** (when does this happen?),
- the Markov chain **does not converge quick enough** to its limiting distribution (how quick does it converge?),
- the **Involution Jumper returns FAIL too often** (when does this happen?),
- N has **odd order**.

Possible problems

The **whole method is in trouble**, if at least one of the following happens:

- we **do not easily find an involution** in G (like for example in $SL(2, 2^n)$ for big n),
- the involution classes of N have a **small probability in the limit distribution** (when does this happen?),
- the Markov chain **does not converge quick enough** to its limiting distribution (how quick does it converge?),
- the **Involution Jumper returns FAIL too often** (when does this happen?),
- N has **odd order**.

Fortunately: Centralisers of involutions seem to contain enough involutions.

Reductions for imprimitive matrix groups

Assume $G < GL(n, q)$ and $Z := G \cap Z(GL(n, q))$ and $V := \mathbb{F}_q^n$ be the natural module, such that:

- V is **absolutely irreducible**, and
- there is an N with $Z < N \triangleleft G$ such that

$$V|_N \cong W_1 \oplus \cdots \oplus W_k$$

with **absolutely irreducible N -modules W_i** that are not all isomorphic.

Reductions for imprimitive matrix groups

Assume $G < GL(n, q)$ and $Z := G \cap Z(GL(n, q))$ and $V := \mathbb{F}_q^n$ be the natural module, such that:

- V is **absolutely irreducible**, and
- there is an N with $Z < N \triangleleft G$ such that

$$V|_N \cong W_1 \oplus \cdots \oplus W_k$$

with **absolutely irreducible N -modules W_i** that are not all isomorphic.

(This situation comes up in the matrix group recognition project when we are looking for a reduction for a group in Aschbacher class \mathcal{C}_2 .)

Reductions for imprimitive matrix groups

Assume $G < GL(n, q)$ and $Z := G \cap Z(GL(n, q))$ and $V := \mathbb{F}_q^n$ be the **natural module**, such that:

- V is **absolutely irreducible**, and
- there is an N with $Z < N \triangleleft G$ such that

$$V|_N \cong W_1 \oplus \cdots \oplus W_k$$

with **absolutely irreducible N -modules W_i** that are not all isomorphic.

(This situation comes up in the matrix group recognition project when we are looking for a reduction for a group in Aschbacher class \mathcal{C}_2 .)

We use the IJ, **for each involution x produced**:

- compute $M := N_G(x)$
- use the MeatAxe to check whether V_M is reducible
- if $x \in N$, we find a reduction.