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

Let $1 < N \triangleleft G = \langle g_1, \ldots, g_k \rangle$ 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 $1 < N \triangleleft G = \langle g_1, \ldots, g_k \rangle$ 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 \( 1 < N \triangleleft G = \langle g_1, \ldots, g_k \rangle \) 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 $1 < N \triangleleft G = \langle g_1, \ldots, g_k \rangle$ 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 $G$. 
Problem

Let $1 < N \triangleleft G = \langle g_1, \ldots, g_k \rangle$ 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 $1 < N \triangleleft G = \langle g_1, \ldots, g_k \rangle$ 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 $N$. 
Problem

Let $1 < N \triangleleft G = \langle g_1, \ldots, g_k \rangle$ 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.
What is the Involution Jumper?

Max Neunhöffer

Initial question

Involution Jumper

What's that?

Jumping classes

Back to our question

Problems

Applications

Input:

\[ G = \langle g_1, \ldots, g_k \rangle \]

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 \( \text{FAIL} \)

Note: If \( xy = yx \) then \( c = 1 \)

\[ G \]

and \( o = 1 \) and \( z = y \).

But this happens rarely.
What is the Involution Jumper?

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

repeat

1. $y := \text{RANDOM}(G)$
2. $c := x^{-1}y^{-1}xy$ and $o := \text{ORDER}(c)$
3. if $o$ is even then
   1. return $c^{o/2}$
4. else
   1. $z := y \cdot c^{(o-1)/2}$ and $o' := \text{ORDER}(z)$
   2. if $o'$ is even then
      1. return $z^{o'/2}$
5. until patience lost

return **FAIL**

Note: If $xy = yx$ then $c = 1$ and $o = 1$ and $z = y$. But this happens rarely.
What is the Involution Jumper?

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

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

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.
What does the Involution Jumper do?

Input: $G = \langle g_1, \ldots, g_k \rangle$ 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: $G = \langle g_1, \ldots, g_k \rangle$ 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:** $G = \langle g_1, \ldots, g_k \rangle$ 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: $G = \langle g_1, \ldots, g_k \rangle$ 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: $G = \langle g_1, \ldots, g_k \rangle$ 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:** \( G = \langle g_1, \ldots, g_k \rangle \) 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.
The Involution Jumper
Max Neunhöffer

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

$$\text{Prob}(\text{IJ}(x) \in aG) = \text{Prob}(\text{IJ}(xg) \in aG).$$

or equivalently

Lemma
Let $x \in G$ be an involution. Then the distribution of $\text{IJ}(x)$ only depends on $xG$ and not on the choice of $x$ in $xG$. 

Proof: 
$f(x, y) :=
\begin{cases}
[x, y]_k & \text{if } \text{ORDER}([x, y]) = 2^n \\
(x[y][x, y])_k & \text{if } \text{ORDER}([x, y]) = 2^n + 1 > 1 \text{ and } \text{ORDER}(y[x, y]) = 2^n \\
y & \text{if } xy = yx \text{ and } \text{ORDER}(y) = 2^n
\end{cases}

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

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

**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

\[
\operatorname{Prob}(IJ(x) \in a^G) = \operatorname{Prob}(IJ(x^g) \in a^G).
\]

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: \( f(x, y) := \)

\[
\left\{
\begin{array}{ll}
[x, y]^k & \text{if } \operatorname{ORDER}([x, y]) = 2k \\
(y[x, y]^k)^l & \text{if } \operatorname{ORDER}([x, y]) = 2k + 1 > 1 \text{ and } \operatorname{ORDER}([y[x, y]^k]) = 2l \\
y^k & \text{if } xy = yx \text{ and } \operatorname{ORDER}(y) = 2k
\end{array}
\right.
\]

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.*
Let $1 < N \triangleleft G = \langle g_1, \ldots, g_k \rangle$ 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”. 
Let $1 < N \triangleleft G = \langle g_1, \ldots, g_k \rangle$ 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

<table>
<thead>
<tr>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Let</em> $1 &lt; N \lhd G = \langle g_1, \ldots, g_k \rangle$ <em>be a finite group with order oracle</em> and <em>N be a normal subgroup</em>. <em>Produce a non-trivial element of N as a word in the</em> $g_i$ <em>with “high probability”.</em></td>
</tr>
</tbody>
</table>

- 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 $N$.**
Let $1 < N \triangleleft G = \langle g_1, \ldots, g_k \rangle$ 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:

<table>
<thead>
<tr>
<th>$G$</th>
<th>$N$</th>
<th># hops*</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_5 \wr S_{10}$</td>
<td>$S_5 \times 10$</td>
<td>1.91</td>
</tr>
<tr>
<td>$\text{GL}(3,3) \wr S_6 &lt; \text{GL}(18,3)$</td>
<td>$\text{GL}(3,3) \times 6$</td>
<td>1.17</td>
</tr>
<tr>
<td>$\text{Sp}(6,3) \otimes 2.O(7,3) &lt; \text{GL}(48,3)$</td>
<td>$\text{Sp}(6,3) \otimes 1$</td>
<td>1.83</td>
</tr>
</tbody>
</table>

* 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 $\text{SL}(2, 2^n)$ 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 $\text{SL}(2, 2^n)$ for big $n$),
- the involution classes of $N$ have a small probability in the limit distribution (when does this happen?),
- $N$ has odd order.

Fortunately: Centralisers of involutions seem to contain enough involutions.
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 whole method is in trouble, if at least one of the following happens:
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 \( \text{SL}(2, 2^n) \) 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.
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 \texttt{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 $\text{SL}(2, 2^n)$ 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 $\text{FAIL}$ 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 $G < \text{GL}(n, q)$ and $Z := G \cap Z(\text{GL}(n, q))$ 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 $G < \text{GL}(n, q)$ and $Z := G \cap Z(\text{GL}(n, q))$ 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 $C_2$.)
Reductions for imprimitive matrix groups

Assume $G < \text{GL}(n, q)$ and $Z := G \cap Z(\text{GL}(n, q))$ and $V := \mathbb{F}_q^n$ be the natural module, such that:
- $V$ is absolutely irreducible, and
- there is an $N$ with $Z < N \vartriangleleft G$ 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 $C_2$.)

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.