Basic concepts

Eamonn O'Brien

University of Auckland

August 2011

Determine the order of a matrix

Let $g \in \mathrm{GL(d, q)}$.
Find $n \geq 1$ such that $g^n = 1$.

$\mathrm{GL(d, q)}$ has elements of order $q^d - 1$, Singer cycles, ...

so not practical to compute powers of g until we obtain the identity.

To find $|g|$: probably requires factorisation of numbers of form $q^i - 1$, a hard problem.

Babai & Beals (1999):

Theorem

If the set of primes dividing a multiplicative upper-bound B for $|g|$ is known, then the precise value of $|g|$ can be determined in polynomial time.

Celler & Leedham-Green (1995): compute $|g|$ in time $O(d^4 \log q)$ subject to factorisation of $q^i - 1$ for $1 \leq i \leq d$.

- First compute a "good" multiplicative upper bound B for $|g|$.

Determine and factorise minimal polynomial for g as

$$m(x) = \prod_{i=1}^{t} f_i(x)^{m_i}$$

where $\deg(f_i) = d_i$ and $\beta = \lceil \log_p \max m_i \rceil$.

$B := \prod_{i=1}^{t} \mathrm{lcm}(q^{d_i} - 1) \times p^{\beta}$

Lemma

Let $B = \prod_{i=1}^{t} \mathrm{lcm}(q^{d_i} - 1) \times p^{\beta}$. *Then* $|g|$ *divides* B.

To see this, reduce g to Jordan normal form over the algebraic closure of $\mathrm{GF}(q)$.

Each eigenvalue lies in an extension field of $\mathrm{GF}(q)$ of dimension d_i and so has multiplicative order dividing $q^{d_i} - 1$.

If a block has size $\gamma_i > 1$, then the p-part of the order of the block is p^{δ} where $\delta = \lceil \log_p \gamma_i \rceil$.

Can we use B to learn $|g|$?

1. Factorise $B = \prod_{i=1}^{m} p_i^{\alpha_i}$ where the primes p_i are distinct.
2. If $m = 1$, then calculate $g^{p_1^j}$ for $j = 1, 2, \ldots, \alpha_1 - 1$ until the identity is constructed.
3. If $m > 1$ then express $B = uv$, where u, v are coprime and have approximately the same number of distinct prime factors. Now g^u has order k dividing v and g^k has order ℓ say dividing u, and $|g|$ is $k\ell$. Hence the algorithm proceeds by recursion on m.

Let $m(x)$ be the minimal polynomial of g. The F_q-algebra generated by g is isomorphic to $F_q[x]/(f(x))$.

It suffices to calculate the multiplicative order of x in the ring.

Hence multiplications can be done in $O(d^2)$ field multiplications.

Celler & Leedham-Green prove the following:

Theorem

If we **can compute** *a factorisation of B, then the cost of the algorithm is $O(d^4 \log q \log\log q^d)$ field operations.*

If we don't complete the factorisation, then obtain *pseudo-order* of g – the order $\times$ some large primes.

Suffices for most theoretical and practical purposes.

Implementations in both GAP and Magma use databases of factorisations of numbers of the form $q^i - 1$, prepared as part of the Cunningham Project.

Example

$$A = \begin{pmatrix} 2 & 5 & 1 & 2 \\ 0 & 1 & 6 & 1 \\ 4 & 0 & 2 & 2 \\ 3 & 3 & 6 & 6 \end{pmatrix}$$

with entries in $\mathrm{GF}(7)$. A has minimal polynomial

$$m(x) = x^4 + 3x^3 + 6x^2 + 6x + 4 = (x+4)^2(x^2 + 2x + 2)$$

Hence $e_1 = 1, e_2 = 2$ and $\beta = \lceil \log_7 2 \rceil = 1$. Hence

$B = (7^1 - 1)(7^2 - 1)7^1 = 336$.

Now $336 = 2^4 \cdot 3 \cdot 7 = uv$ where $u = 2^4$ and $v = 3 \cdot 7$.

A^u has order dividing v. Reapply: $|A^u| = 21$.

A^v has order dividing u. Reapply: $|A^v| = 8$.

Conclude $|A| = 168$.

Even order?

Assume we know B, multiplicative upper bound to $|g|$.

If we just know B, then we can learn in polynomial time the *exact* power of 2 (or of any specified prime) which divides $|g|$.

By repeated division by 2, write $B = 2^m b$ where b is odd.

Now compute $h = g^b$, and determine (by powering) its order which divides 2^m.

In particular, can deduce in polynomial time if g has *even order*.

Computing powers of matrices

We can compute large powers n of g in at most $2\lfloor \log_2 n \rfloor$ multiplications by the standard doubling algorithm:

- $g^n = g^{n-1}g$ if n is odd
- $g^n = g^{(n/2)2}$ if n is even.

Black-box algorithm.

Frobenius normal form or . . .

Rational canonical form of a square matrix A is a canonical form that reflects the structure of the minimal polynomial of A. Can be constructed over given field, no need to extend field.

Definition

A is equivalent to $\begin{pmatrix} C_1 & 0 & \dots & 0 \\ 0 & C_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & C_\ell \end{pmatrix}$.

Each block C_i is the companion matrix of monic $f_i \in F[x]$ and $f_i | f_{i+1}$ for $1 \leq i \leq \ell$.
The minimal polynomial of A is f_ℓ and char poly is $f_1 \cdot f_2 \dots f_\ell$.

Frobenius normal form N of A is sparse.

Hence multiplication by N costs just $O(d^2)$ field operations.

A faster power algorithm

1. Construct the Frobenius normal form of g and record change-of-basis matrix C.
2. From the Frobenius normal form, read off the minimal polynomial $m(x)$ of g, and factorise $m(x)$ as a product of irreducible polynomials.
3. Compute multiplicative upper bound, B, to the order of g.
4. If $n > B$, then replace n by $n \bmod B$. By repeated squaring, calculate $x^n \bmod m(x)$ as a polynomial of degree $k-1$, where k is the degree of $m(x)$.
5. Evaluate this polynomial in the Frobenius form of g to give g^n wrt Frobenius basis.
6. Now compute $C^{-1}g^nC$ to return to the original basis.

Complexity of this task

Lemma

Let $g \in \mathrm{GL}(d, q)$ *and let* $0 \leq n < q^d$*. This is a Las Vegas algorithm that computes* g^n *in* $O(d^3 \log d + d^2 \log d \log\log d \log q)$ *field operations.*

The composition tree for G

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

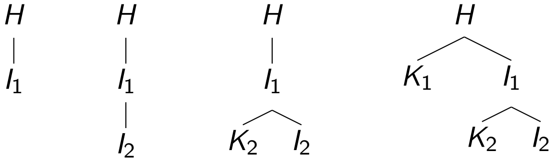
$$H$$

$$K \quad I$$

- Node: section H of G.
- Image I: image under homomorphism or isomorphism.
- Kernel K.
- Leaf is "composition factor" of G: simple modulo scalars.
  Cyclic not necessarily of prime order.

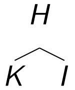
Tree is constructed in right depth-first order.

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



Constructing kernels

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



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

Two approaches to construct kernel.

1. 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.

Random generation of the kernel

Let $x_1, \ldots, x_t$ be generating set for $h \in H$.

Let $y_j = \phi(x_j)$ for $j = 1, \ldots, t$.

Let $h \in H$ and let $i = \phi(h)$.

Write $i = w(y_1, \ldots, y_t)$.

Let $\bar{h} = w(x_1, \ldots, x_t)$.

Now $k = h\bar{h}^{-1} \in K := \ker \phi$.

Choose random $h \in H$ to obtain random generator k of K.

Randomised algorithm to construct the kernel – but assumes that we can write $i = w(y_1, \ldots, y_t)$.

Base cases for recursion

Classical group in natural representation or other **almost simple modulo scalars**: $S \leq H/Z \leq Aut(S)$

Principal focus: *matrix representations in defining characteristic*.

Constructive recognition: the main tasks

$H = \langle X \rangle \leq \mathrm{GL}(d, q)$ where H is (quasi)simple.
So H is perfect and H/Z is simple.

1. Given $h \in H$, express $h = w(X)$.
   ("Constructive membership problem")
2. Given $G = \langle Y \rangle$ where G is representation of H,
   - solve constructive membership problem for G;
   - construct "effective" isomorphisms
     $\phi : H \longmapsto G$
     $\tau : G \longmapsto H$.

Key idea: standard generators.

Using standard generators

Define *standard generators* $\mathcal{S}$ for $H = \langle X \rangle$.

Need algorithms to:

- Construct $\mathcal{S}$ as *words* in X.
- For $h \in H$, express h as $w(\mathcal{S})$ and so as $w(X)$.

If $\langle Y \rangle = G \simeq H$ then:

- Find standard generators $\bar{\mathcal{S}}$ in G as words in Y.
- For $g \in G$, express g as $w(\bar{\mathcal{S}})$ and so as $w(Y)$.

Choose $\mathcal{S}$ so that solving for word in $\mathcal{S}$ is easy.

Now define isomorphism $\phi : H \longmapsto G$ from $\mathcal{S}$ to $\bar{\mathcal{S}}$

Effective: if $h = w(\mathcal{S})$ then $\phi(h) = w(\bar{\mathcal{S}})$.

Similarly $\tau : G \longmapsto H$.

Motivation

Example

$H = \langle X \rangle = \mathrm{SL}(d, q)$
$G = \langle Y \rangle$ is symmetric square repn.

H is our "gold-plated" copy in which we know information.

Examples include

- Conjugacy classes of elements.
- Maximal subgroups.

We know or can obtain these readily as words w in S.

If we know $\bar{S} \subset G$, we can evaluate w in $\bar{S}$.

So we now know this information in our arbitrary copy G.

Application I: Conjugacy classes of classical groups

Example: $H = \langle X \rangle = \mathrm{SX}(d, q)$
$G = \langle Y \rangle$ is symmetric cube.

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

Murray: algorithm, which given d and q, constructs classes for $\mathrm{SX}(d, q)$.

$\phi : H \longmapsto G$ now maps class reps and centralisers to G.

Example

Higman's (1961) count of p-groups of p-class 2.
Eick and O'B (1999): algorithm which, given d and p, counts precisely the number of d-generator p-groups of class 2.
Critical task: for each conjugacy class rep r in $G := \Lambda^2(\mathrm{GL(d, p)})$ use Cauchy-Frobenius theorem to count fixed points for r.

Application II: Maximal subgroups of classical groups

Kleidmann & Liebeck (1990): describe some maximal subgroups of classical groups where $d \geq 13$.

Bray, Holt & Roney-Dougal (ongoing): construct generating sets for geometric maximal subgroups, and all maximals for $d \leq 12$.

So obtain $M \leq H := \mathrm{SX}(d, q)$, classical group in natural representation.

Use $\phi : H \longmapsto G$ to construct image of M in arbitrary representation G.