

A constructive recognition algorithm for the special linear group

F. Celler and C. R. Leedham-Green

Abstract

In the first part of this note we present an algorithm to recognise constructively the special linear group. In the second part we give timings and examples.

1 Introduction

It seems possible, using Aschbacher's celebrated analysis of subgroups of classical groups [5], to develop algorithms that will answer basic questions about the group G generated by a subset X of $GL(d, q)$, for modest values of d and q , as is already possible for permutation groups. The best strategy may involve trying to recognise very large subgroups of $GL(d, q)$ by special techniques.

In the case of permutation groups, special techniques are used to recognise the alternating and symmetric groups. This is done by making a random search for elements of a certain cycle type. If such elements are found in a primitive group, the group is known to contain the alternating group. If no such elements are found after a sufficiently long search, one proceeds with the expectation that one is dealing with a smaller group. For linear groups, the corresponding question is to determine whether or not the group in question contains a classical group.

It is possible to recognise the classical groups in a *non-constructive* way as described in [6], [7], and [2]. This still leaves the further problem of exhibiting an explicit isomorphism, that is to say, given that the group $G = \langle X \rangle$ contains a classical group, how can one express a given element A of G as a word in X ? We call an algorithm to solve such a problem a *constructive* recognition algorithm.

The natural idea is to find a suitable generating set Y for G and expressions for the elements of Y as words in X , such that we have an algorithm expressing A as a word in Y . We present an algorithm which allows one to do this in the case of the special linear group; here Y will be a suitable set of transvections as it is well known that $SL(d, q)$ is generated by transvections.

Using Gaussian Elimination it is possible to rewrite an element of $SL(d, q)$ as a product of elementary matrices, which in turn can be written as a product of transvections. Hence, our goal is to find the transvections required in a Gaussian Elimination as words in X , and then use the Gaussian Elimination to rewrite an element of G as a word in these transvections.

The words in X that define the transvections may be very long. It is therefore sensible to give these words as ‘straight line programs’. That is to say, to define a word w in X we define a sequence of words w_1, w_2, \dots, w_n in X , where each w_i is either of the form $x^{\pm 1}$ for $x \in X$, or of the form $w_j^{\pm 1} w_k^{\pm 1}$ for $j, k < i$ or w_j^{-1} for $j < i$. This may reduce the number of multiplications required to define w dramatically. For example if $w = x^n$, then w can be defined in this way by $O(\log n)$ multiplications.

One of the main applications of the constructive recognition algorithm will be in the following setting. In investigating a matrix group H along the lines of Aschbacher’s classification, one ends up with either a classical group, an almost simple group, or a reduction to a smaller group; in the latter case, we get a homomorphism φ of H into S , where S is cyclic, a permutation group, or some matrix group of smaller dimension or over a smaller field. If $\varphi(H)$ is a matrix group containing the special linear group in its natural representation, we can use the constructive algorithm to produce elements of the kernel of φ .

We assume that the algorithm is applied when we have already proved, using the much faster non-constructive recognition algorithm, that $\langle X \rangle$ does contain $SL(d, q)$. As an alternative, the program could report failure if some randomised component of the algorithm failed to complete in the expected time. The effect of proving in advance that G does contain $SL(d, q)$ is to change the algorithm from a Monte Carlo algorithm, when failure would give statistical evidence that G does not contain $SL(d, q)$, to a Las Vegas algorithm, where the randomised features of the algorithm merely make the runtime uncertain.

We also assume $d > 1$ in the following.

The algorithm described in this paper is randomised; however explicit isomorphisms constructed will always be correct. Such algorithms are called Las Vegas algorithms; the output is either correct or the algorithm reports failure (with prescribed probability $\leq \varepsilon$).

To say that a Las Vegas algorithm has (Las Vegas) complexity f , where f is some function of the input, means that, for some ϵ , $0 < \epsilon < 1$, it will, with probability at least $1 - \epsilon$, terminate in time less than kf for some constant k independent of the input. The algorithm will then terminate with probability $1 - \epsilon^2$, in time less than $2kf$, as one could simply run the program twice. So ϵ can be squared by doubling k , and hence the algorithm has complexity f for any ϵ , $0 < \epsilon < 1$, however small. Hence running the algorithm until success has expected complexity $O(f)$. The stated running times and complexities in

the following sections are always to be understood as expected running times and complexities.

2 Constructing a transvection

A transvection is an element of $GL(d, q)$ acting trivially on a hyperplane and on its quotient space; we shall call this hyperplane the *centralised subspace* of the transvection. A transvection is conjugate to $\text{diag}(J_2(1), I_{d-2})$, the block matrix with $J_2(1)$ and I_{d-2} on the diagonal, where $J_2(1)$ is a Jordan block of dimension 2 and eigenvalue 1 and I_{d-2} is the identity of $GL(d-2, q)$. Although only a small proportion of the elements of $GL(d, q)$ are transvections, we show in this section that the transvections are relatively easy to find provided q is not too large.

Let F_q be the finite field with q elements, $q = p^k$, where p is prime.

In order to find a transvection, we look for an element conjugate to $\text{diag}(J_2(\alpha), R)$, the block matrix with $J_2(\alpha)$ and R on the diagonal; $J_2(\alpha)$ is a Jordan block of dimension 2 and eigenvalue $\alpha \neq 0$, R is an element of $GL(d-2, q)$, such that α is not an eigenvalue of R and R is semisimple. As R is semisimple, p does not divide the order $o(R)$ of R . Raising $\text{diag}(J_2(\alpha), R)$ to the least common multiple m of the orders of R and α yields a transvection; R^m is trivial, $J_2(\alpha)^m = \begin{pmatrix} 1 & m\alpha^{-1} \\ 0 & 1 \end{pmatrix}$ and $m\alpha^{-1} \neq 0$ because p does not divide $o(R)$ or $o(\alpha)$. In order to find random elements in $\langle X \rangle = GL(d, q)$ we use the algorithm described in [3], which requires, after a preprocessing phase, one matrix multiplication per random element; but in addition we also keep track of the expressions for the random elements in the given generators X .

Note that it is possible to obtain a suitable multiple of $o(R)$ from the degrees of the factors of the minimal polynomial of R , but as we want short expressions in the given generators for our transvection, we use the precise order of R , see [1].

We now estimate the proportion of elements in $GL(d, q)$ of this form. The first lemma counts the elements in $GL(d, q)$ having a given eigenvalue, the second lemma counts the semisimple elements.

Lemma 2.1. *Let α be an element of F_q^* and let M_α be the set of elements in $GL(d, q)$ having eigenvalue α . Then*

$$\frac{1}{q-1} - \left(\frac{1}{q-1}\right)^2 \leq \frac{|M_\alpha|}{|GL(d, q)|} \leq \frac{1}{q-1}.$$

Proof: In order to get the stated upper bound we use a counting argument similar to that in [4]: For each non-zero vector v choose a vector

space complement W_v such that $F_q^d = \langle v \rangle \oplus W_v$. Let \mathcal{E} be the set of triples $\{(v, \beta, \tau) \mid v \in F_q^d, \beta \in GL(W_v), \tau \in \text{Hom}(W_v, \langle v \rangle)\}$. Now if $A \in GL(d, q)$ has an eigenvalue α , and we choose a corresponding eigenvector v , then the pair (A, v) corresponds to exactly one triple in \mathcal{E} . On one hand

$$|\mathcal{E}| = (q^d - 1) \cdot |GL(d-1, q)| \cdot q^{d-1} = (q^d - 1)q^{d-1} \prod_{i=0}^{d-2} (q^{d-1} - q^i) = |GL(d, q)|.$$

On the other hand each A with eigenvalue α has at least $q-1$ eigenvectors. Therefore the number of matrices in $GL(d, q)$ with eigenvalue α is at most $|GL(d, q)|/(q-1)$.

In order to get the stated lower bound we count the matrices whose characteristic polynomials have α as root with multiplicity one; that is, the matrices conjugate to $\text{diag}(\alpha, R)$ for $R \in \mathcal{R}_\alpha \subset GL(d-1, q)$ where \mathcal{R}_α is the set of matrices which do not have eigenvalue α . Let \mathcal{C}_α be a set of representatives of the conjugacy classes in \mathcal{R}_α under the action of $GL(d-1, q)$. Then $\sum_{R \in \mathcal{C}_\alpha} [GL(d, q) : C_{GL(d, q)}(\text{diag}(\alpha, R))]$ matrices in $GL(d, q)$ are conjugate to $\text{diag}(\alpha, R)$. The centraliser of $\text{diag}(\alpha, R)$ is isomorphic to $F_q^* \times C_{GL(d-1, q)}(R)$; so that the above sum gives

$$\begin{aligned} |M_\alpha| &\geq \sum_{R \in \mathcal{C}_\alpha} \frac{|GL(d, q)|}{(q-1) \cdot |C_{GL(d-1, q)}(R)|} \\ &= \frac{|GL(d, q)|}{(q-1) \cdot |GL(d-1, q)|} \sum_{R \in \mathcal{C}_\alpha} [GL(d-1, q) : C_{GL(d-1, q)}(R)] \\ &= \frac{|GL(d, q)|}{q-1} \cdot \frac{|\mathcal{R}_\alpha|}{|GL(d-1, q)|} \\ &\geq \frac{1}{q-1} \cdot \left(1 - \frac{1}{q-1}\right) \cdot |GL(d, q)|, \end{aligned}$$

since from the first part of the proof we already know that at least $(q-2)|GL(d-1, q)|/(q-1)$ matrices do not have eigenvalue α , so that $|\mathcal{R}_\alpha| \geq (q-2)|GL(d-1, q)|/(q-1)$. $\ddagger\ddagger$

Lemma 2.2. *Let S be the set of semisimple, regular elements in $GL(d, q)$; that is to say, the set of elements whose characteristic polynomial is square-free. Then*

$$\left(\frac{q-1}{q}\right)^2 |GL(d, q)| \leq |S|.$$

Proof: Let \mathcal{P} be the set of all square-free polynomials of degree d with leading coefficient one and non-zero constant term. The set S is a union of

$GL(d, q)$ -conjugacy classes. We can choose as a representative for each class the companion matrix C_f of a polynomial f in \mathcal{P} .

Let $f = \prod_{i=1}^r f_i \in \mathcal{P}$, where f_i is irreducible of degree d_i . Then the centraliser of C_f is isomorphic to $C_{GL(d_1, q)}(C_{f_1}) \times \cdots \times C_{GL(d_r, q)}(C_{f_r})$, so that $|C_{GL(d, q)}(C_f)| = \prod_i (q^{d_i} - 1) \leq q^d$. Therefore we get

$$\begin{aligned} |S| &= \sum_{f \in \mathcal{P}} [GL(d, q) : C_{GL(d, q)}(C_f)] \\ &\geq \sum_{f \in \mathcal{P}} \frac{|GL(d, q)|}{q^d} \\ &= \frac{|\mathcal{P}|}{q^d} |GL(d, q)|. \end{aligned}$$

Now let \mathcal{N} be the set of all polynomials of degree d with leading coefficient one and non-zero constant term that do have a non-trivial square factor. In order to get an upper bound on $|\mathcal{N}|$ we use the following counting argument. As $f \in \mathcal{N}$ is not square-free it contains a non-trivial factor h of degree $k \leq d/2$ at least twice. Counting all possibilities for k , the factor h , and the co-factor f/h^2 , we count each polynomial with repeated factors at least once.

$$\begin{aligned} |\mathcal{N}| &\leq \sum_{k=1}^{\lfloor d/2 \rfloor} (q^{k-1}(q-1)) \cdot (q^{d-2k-1}(q-1)) \\ &= \frac{(q-1)^2}{q^2} \sum_{k=1}^{\lfloor d/2 \rfloor} q^{d-k} \\ &= \frac{(q-1)}{q^2} (q^d - q^{\lfloor d/2 \rfloor}) \\ &< \frac{(q-1)}{q^2} q^d. \end{aligned}$$

Hence we have $|\mathcal{P}| = (q-1)q^{d-1} - |\mathcal{N}| > (q-1)q^{d-1}(1 - \frac{1}{q})$ and therefore $|S| > \left(\frac{q-1}{q}\right)^2 \cdot |GL(d, q)|$. ‡‡

Lemma 2.3. *Let $q > 3$. Let N_α be the set of elements A of $GL(d, q)$ such that both the characteristic polynomial $c(x)$ and the minimal polynomial of A have α as a root of multiplicity two and $c(x)/(x-\alpha)^2$ is square-free. Then*

$$\frac{1}{q} \left(\left(\frac{q-1}{q} \right)^2 - \frac{1}{q-1} \right) \leq \frac{|\cup_{\alpha \in F_q^*} N_\alpha|}{|GL(d, q)|}.$$

Proof: Let A be an element of N_α . Then A is conjugate to $\text{diag}(J_2(\alpha), R)$, where $R \in GL(d-2, q)$, α is not an eigenvalue of R and R is semisimple. The set \mathcal{R}_α of all such R is a union of $GL(d-2, q)$ -conjugacy classes; let \mathcal{C}_α be a system of representatives for these classes. As α is not an eigenvalue of R the centraliser of $\text{diag}(J_2(\alpha), R)$ is isomorphic to $C_{GL(2,q)}(J_2(\alpha)) \times C_{GL(d-2,q)}(R)$, therefore its order is $q \cdot (q-1) \cdot |C_{GL(d-2,q)}(R)|$. Taking the sum $[GL(d, q) : C(\text{diag}(J_2(\alpha), R))]$ for all $R \in \mathcal{C}_\alpha$, we get

$$|N_\alpha| = \frac{1}{q(q-1)} \cdot |GL(d, q)| \cdot |\mathcal{R}_\alpha| / |GL(d-2, q)|.$$

According to Lemma 2.2 at least $\frac{(q-1)^2}{q^2} \cdot |GL(d-2, q)|$ matrices are semisimple, and according to Lemma 2.1 at most 1 in $q-1$ matrices have eigenvalue α ; therefore $|\mathcal{R}_\alpha| / |GL(d-2, q)|$ is at least $\frac{(q-1)^2}{q^2} - \frac{1}{q-1}$. As R is semisimple it does not contain any other 2 dimensional Jordan block, so $N_\alpha \cap N_\beta = \emptyset$ for $\alpha \neq \beta$. Therefore the total proportion of matrices of the required form is at least $\frac{1}{q} \cdot (\frac{(q-1)^2}{q^2} - \frac{1}{q-1})$. $\ddagger\ddagger$

We can now analyse an algorithm for finding a transvection if $\langle X \rangle = GL(d, q)$. The general case $SL(d, q) \leq \langle X \rangle$ is discussed at the end of this section.

Theorem 2.4. *It is possible to find a transvection in $GL(d, q) = \langle X \rangle$ as a word in X using Las Vegas $O(qd^3)$ finite field operations if $O(q)$ random elements as words in X are given.*

Proof: If $d = 2$ then 1 in $q+1$ matrices in $GL(d, q)$ are conjugate to $J_2(\alpha)$ for some α .

Now assume $d > 2$ and $q > 2$. Lemma 2.3 shows that the proportion of suitable elements is at least $1/(5q)$. Therefore the probability of failure is less than e^{-1} if we look at $5q$ random elements.

Now assume $d > 2$ and q arbitrary. Counting the matrices conjugate to $\text{diag}(J_2(\alpha), R)$, where the characteristic polynomial of R is square-free, in the same way as in Lemma 2.3, shows that there are at least $\frac{(q-1)^2}{q^5} \cdot |GL(d, q)|$ such matrices. Hence the proportion $\frac{(q-1)^2}{q^5}$ of suitable elements is independent of d for $q = 2$ and 3 .

It follows that the proportion of suitable elements is at least $1/32$ for $q = 2$, at least $4/243$ for $q = 3$, and at least $1/(5q)$ for $q > 3$. Hence the probability of failure will be less than e^{-1} for any q if we choose $21q$ elements.

Computing each characteristic polynomial $c(x)$ requires $O(d^3)$ finite field operations [1]. By looking at the greatest common divisor of $c(x)$ and its derivative it is possible to check if $c(x)$ has a root of multiplicity 2 and square-free co-factor without quantifying over the field. Computing the gcd requires

$O(d^2)$ finite field operations. We do not need to compute the minimal polynomial because the condition on the minimal polynomial given in Lemma 2.3 can be checked by looking at the dimension of the eigenspace for the eigenvalue α . This again requires $O(d^3)$ finite field operations. Therefore we can check if we have found a suitable element using $O(d^3)$ finite field operations.

We have to look at $21q$ elements to get a probability of failure of less than e^{-1} ; computing and checking one element requires $O(d^3)$ finite field operations. Having found a suitable element $\text{diag}(J_2(\alpha), R)$ we can compute the order of R using $O(d^3 \log q \log t)$ finite field operations, where t is the maximal number of prime factors in $q^i - 1$ for $i \leq d$ (see [1]). $\ddagger\ddagger$

Remark. After a preprocessing phase, finding a random element using the method described in [3] requires one matrix multiplication.

Remark. Note that we do not have to raise the element to the power $\text{lcm}(o(R), o(\alpha))$ explicitly, which would require $O(d^4 \log q)$ finite field operations. However, although we need only $O(q)$ matrix multiplications to find the transvection, evaluating the corresponding straight line program requires $O(q + d \log q)$ multiplications because of this last powering step.

We now consider the general case when $SL(d, q) \leq \langle X \rangle \leq GL(d, q)$.

If $d > 3$, the determinant will simply partition the set of suitable elements into subsets of approximate equal cardinality. If $d > 1$ the chance that a random element has an eigenvalue α is still about one in $q - 1$ even if we impose a condition on the determinant of R .

If $d = 3$ and $G = SL(3, q)$, however, the determinant condition forces $R \in GL(1, q)$ to be (α^{-2}) . Hence, if F_q^* has elements of order 3 we get a slightly worse chance, with the exception of the rogue group $SL(3, 4)$ which has no elements of the required form.

To catch these exceptions we use the following variation of the algorithm. We try to find a random element A with minimal polynomial m_A and characteristic polynomial c_A , such that m_A has a root α of multiplicity two, $m_A/(x - \alpha)^2$ is square-free and c_A has α as root of multiplicity two or three. If q is even and we find an element where α is a root of multiplicity three of m_A , we try A^2 instead.

3 Constructing a transvection basis

We assume in this section that $\langle X \rangle$ contains $SL(d, q)$.

We now describe an algorithm to find a set of transvections that generate $SL(d, q)$. We first find a set of transvections that centralise a common subspace M of co-dimension one. We then look for conjugating elements g_j such that $M \cap \bigcap_j M g_j$ is trivial.

Assume that we have found a transvection using Theorem 2.4. Hence we know an element $A \in \langle X \rangle$, where A has order pm , $t_1 = A^m$ is a transvection, and A is conjugate to $\text{diag}(J_2(\alpha), R)$, where R is semisimple and regular.

Let M be the subspace of co-dimension 1 that is centralised by t_1 , and let b be a vector not in M . For a transvection t centralising M define $\pi(t)$ to be $b - b^t \in M$. Let $M = S \oplus T$ be the A -invariant decomposition of M into a one-dimensional subspace S and a $(d - 2)$ -dimensional subspace T .

Our goal is to find a set of $(d - 1) \cdot k$ transvections $\{t_i\}$ with centralised subspace M , such that $\{\pi(t_i)\}$ is a basis for $M < F_q^d$ over the prime field of F_q . Using these transvections it is possible to write any transvection t centralising M as word in the transvections t_i ; if $\pi(t) = \sum \alpha_i \pi(t_i)$ for $0 \leq \alpha_i < p$ then $t = \prod t_i^{\alpha_i}$.

We find such a set of transvections in two stages. In the first stage we find another transvection by taking conjugates of t_1 with random elements of $\langle X \rangle$ until we find a transvection that also normalises M . In the second stage we conjugate our transvections with A . We then iterate until we find a basis.

The next lemma estimates the chances of finding a conjugate of t_1 that will normalise M or fix b .

Lemma 3.1. *Let t be a transvection, M its centralised subspace and b a vector outside M . Let g be a random element of $GL(d, q)$.*

- (1) *The probability that t^g normalises M is at least 1 in $q + 1$.*
- (2) *The probability that t^g fixes b is at least 1 in $q + 1$.*

Proof: It is clear that t^g normalises M if and only if t^g either centralises M or $\pi(t)g$ lies in M .

If $d = 2$ and t^g normalises M it must also centralise this one-dimensional subspace. We have a probability of 1 in $q + 1$ that Mg and M are equal as we can assume that g is a random invertible matrix, and so Mg must be a random, one-dimensional subspace.

If $d > 2$ the probability that t^g centralises M is one in $(q^d - 1)/(q - 1)$ and we ignore this possibility for the analysis. Now g is a random element of $GL(d, q)$, so $\pi(t)g$ is a random, non-zero vector. The probability that a random, non-zero vector of F_q^d lies in a given hyperplane is $q^{d-1} - 1$ in $q^d - 1$.

Part (2) is proved in the same way. ‡‡

Using a transvection normalising M we can now construct a conjugate of t centralising M .

Lemma 3.2. *Let t be transvection with centralised subspace M . It is possible to find a conjugate t' of t centralising the same subspace M using Las Vegas $O(d^3 q)$ finite field operations if $O(q)$ random elements of $GL(d, q)$ are given.*

Proof: Lemma 3.1 states that a conjugate s of t has a chance of at least

1 in $q + 1$ of normalising M .

If $d = 2$ then a transvection normalising M already centralises M and we can choose $t' = s$.

If $d > 2$, then either s centralises M and we choose $t' = s$ or it does not. In the latter case we choose $t' = t^s$ which does centralise M . $\ddagger\ddagger$

For the second stage we assume that we have found another transvection t_2 fixing M such that T -component of $\pi(t_2)$ is non-trivial. The conjugate t_2^A still centralises M and A acts as R on the T -component of $\pi(t_2)$.

We now form $t_3 = t_2^A, t_4 = t_3^A, \dots$, until some linear relation occurs between the $\pi(t_i)$, $1 < i$. We are looking for a basis over the prime field of F_q for T ; if a linear relation occurs before we have such a basis, we add in a new transvection constructed in the same way as t_2 . Iterating this process, we get a basis for T . We shall see later that we expect to construct this basis with $O(k)$ iterations.

Remark. Using [3] for finding random elements, we need $O(q)$ multiplications to evaluate the straight line program for A , $O(d \log q)$ additional multiplications to evaluate the straight line program for t_1 . We expect to need $O(k)$ iterations, therefore evaluating the straight line programs for all t_i simultaneously requires $O(qk + d \log q)$ multiplications.

After we have found the $\{t_i\}$, our next goal is to find $d - 1$ elements $g_j \in \langle X \rangle$ such that $M \cap \bigcap_j M g_j = \{0\}$.

Taking random elements we expect to find such a set $\{g_j\}$ after $O(d)$ tries. However, as we assume that constructing a random element already requires $O(d^3)$ finite field operations, this would require $O(d^4)$ finite field operations altogether. So instead of taking $O(d)$ random elements, we start with just one random element g_1 . We now form $g_2 = g_1 A, g_3 = g_2 A$, and so on. If we do not get $M \cap \bigcap_j M g_j = \{0\}$, we add in a new random element. As R is semisimple and regular, we can choose a basis such that A is sparse. Hence multiplication by A can be done in $O(d^2)$ finite field operations.

We now prove that, with positive probability independent of d and q , two random vectors of the underlying vector space will generate the vector space as A -module.

Lemma 3.3. *Let R be a semisimple, regular element of $GL(d, q)$. Then the underlying row space F_q^d is generated as R -module by Las Vegas $O(1)$ random vectors.*

Proof: We can assume that R is $\text{diag}(R_1, \dots, R_l)$, where R_i has an irreducible characteristic polynomial. Let $F_q^d = \bigoplus_{i=1}^l V_i$ be the corresponding decomposition of F_q^d into irreducible A -modules V_i . Therefore, each non-trivial vector in V_i generates V_i as R -module.

Let $a(j)$ be the number of j dimensional subspaces occurring in the above

decomposition. First of all we assume that there are only blocks of dimension j . Taking two random non-trivial vectors we have a chance of one in q^{2j} that both vectors have a trivial V_i component for a fixed i . Hence we have a chance of $(1 - 1/q^{2j})^{a(j)}$ to generate the whole vector space with two random vectors. It is clear that $a(j) < q^j$. Hence

$$\begin{aligned} (1 - 1/q^{2j})^{a(j)} &\geq (1 - 1/q^{2j})^{q^j} \\ &\geq (1 - q^{-2})^{q^{-j}} \\ &\geq 4^{-q^{-j}}. \end{aligned}$$

Now, if there are blocks of various dimensions, we have a chance of success of at least $\prod_{j=1}^d 4^{-q^{-j}} \geq 4^{-1/(q-1)}$. $\ddagger\ddagger$

4 Recognising SL constructively

Putting the algorithms of section 2 and 3 together we have now proved

Theorem 4.1. *Given $\langle X \rangle = GL(d, q)$, we can construct straight line programs for a set of $(d - 1)k$ transvections $\{t_i\}$ with common centralised subspace M , such that $\{\pi(t_i)\}$ is linear independent over the prime field of F_q using Las Vegas $O(qkd^3)$ finite field operations provided that $O(qk)$ random elements as words in X are given.*

Using an additional Las Vegas $O(d^3)$ finite field operations we can construct a set of elements $\{g_j\}$ as straight line programs such that $\bigcap_j Mg_j = \{0\}$.

Using the transvections $\{t_i\}$ and the elements $\{g_j\}$ we can now write any element of $G = \langle X \rangle$ as word in X . This word is represented as straight line program. If $G > SL(d, q)$, it is a triviality to find an element Z of G such that $G = \langle Z, SL(d, q) \rangle$, and to give Z as word in X .

As $\bigcap Mg_j$ is trivial, we can choose a basis $\mathcal{B} = \{b_1, \dots, b_d\}$ for F_q^d such that $b_j \notin Mg_j$ and $b_j \in Mg_{j'}$ for $j' \neq j$. From now on assume that \mathcal{B} is the standard basis of F_q^d .

If we have a transvection t centralising M and $\pi(t) = \sum \alpha_i \pi(t_i)$ then we know that $t = \prod t_i^{\alpha_i}$. As we know straight line programs for the t_i we can produce a straight line program for t . If we have a transvection t centralising Mg_j , then $t^{g_j^{-1}}$ centralises M , and we can therefore produce a straight line program for t using the straight line programs for the $\{t_i\}$ and g_j as before.

In order to write an element A of G as word in X , we proceed as follows. First set $A' = Z^n A$, where n is the integer of smallest modulus such that $\det(A') = 1$. The algorithm then uses a Gaussian Elimination, that is biased

to prefer column operations, to write A' as product of transvections T_k , where each transvection T_k centralises M_{i_k} for some i_k . By solving a system of linear equations the algorithm is now able to write each T_k as a product of $t_{i_k}^{g_j}$. This enables one to write A as a straight line program in $X \cup Z$.

5 Variations

5.1 Computing a transvection, q large

If $q > d + 1 > 3$, the following algorithm can be used to find a transvection in G .

First we try to find an element A that is almost irreducible; that is to say, that acts irreducibly on a subspace of co-dimension 1. About 1 in d of the elements of G have this property if G contains $SL(d, q)$, see [6]. After raising A to the power $o(A)/\gcd(o(A), q - 1)$, this power B will either be a scalar matrix, in which case we try again, or it will be a diagonal matrix with two eigenvalues, say α of multiplicity 1 and β of multiplicity $d - 1$.

If G is $SL(d, q)$, we know that $\alpha \cdot \beta^{d-1} = 1$; since q is larger than $d + 1$, the worst case is $q = 2d + 1$, in which case $\beta^{d-1} = \pm\beta^{-1}$. If $\beta^{d-1} = \beta^{-1}$ then B is a scalar matrix and we have to try again.

When we find such a B , we conjugate B by a random element of G . Since q is large, the eigenspaces of B and C with eigenvalue β will almost certainly be distinct. These spaces then intersect in a subspace W of co-dimension 2 which is invariant under the action of B and C . Let \overline{B} and \overline{C} be the images of B and C acting on V/W , and \overline{G} the group they generate. We shall now search for a transvection in \overline{G} using the algorithm described in Section 2 hoping that \overline{G} contains $SL(2, q)$ and try to pull it back. In an attempt to avoid the situation where $SL(d, q) \not\leq \overline{G}$ we demand that the squares of \overline{B} , \overline{C} , \overline{BC} do not commute. Hence G is neither abelian nor imprimitive.

Assume that we have found a transvection in $GL(2, q)$. Its preimage S in $GL(d, q)$, after using a suitable basis transformation, is of the form

$$S = \begin{pmatrix} 1 & \delta & t_1 \\ & 1 & t_2 \\ & & \beta I_{d-2} \end{pmatrix},$$

where $\delta \neq 0$ and t_i are row vectors of length $d - 2$.

First assume that $\beta \neq 1$ and let n be the order of β . Using the same basis as above we see that

$$S^n = \begin{pmatrix} 1 & n\delta & (\sum_{i=0}^{n-1} \begin{pmatrix} 1 & \delta \\ 0 & 1 \end{pmatrix}^i \beta^{n-1-i}) \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \\ 0 & 1 & I \end{pmatrix}.$$

Note that $n\delta \neq 0$ because n and p are coprime. Now we need to look closer at the upper right corner. Expanding the sum gives

$$\sum_{i=0}^{n-1} \begin{pmatrix} 1 & \delta \\ 0 & 1 \end{pmatrix}^i \beta^{n-1-i} = \sum_{i=0}^{n-1} \begin{pmatrix} \beta^{n-1-i} & i\delta\beta^{n-1-i} \\ 0 & \beta^{n-1-i} \end{pmatrix}.$$

Let $x = \sum_{i=0}^{n-1} \beta^i$; then $x\beta = x$ because n is the order of β . As we assume $\beta \neq 1$, x must be zero, and therefore we get $S^n = \begin{pmatrix} 1 & n\delta & * \\ & 1 & \\ & & I \end{pmatrix}$, which is a transvection.

Now assume that $\beta = 1$ and q is even. In this case it follows that

$$S^2 = \begin{pmatrix} 1 & 0 & \delta t_2 \\ 0 & 1 & 0 \\ 0 & 0 & I \end{pmatrix}.$$

If t_2 is trivial then $S = \begin{pmatrix} 1 & \delta & t_1 \\ 0 & 1 & 0 \\ 0 & 0 & I \end{pmatrix}$ is already a transvection or $t_2 \neq 0$ and S^2 is a transvection.

If $\beta = 1$ but q is odd then the order of S is p and S is a transvection if and only if $t_2 = 0$. So the only bad case is if we find an element S with $\beta = 1$, $t_2 \neq 0$ and q is odd. We avoid this case from the beginning by using only diagonal elements with $\beta \neq 1$ if $p \neq 2$. As β is the power of an element from a field with q^{d-1} elements lying in the subfield with q elements we have a chance of one in $\varphi(l)$ of getting a bad β , where l is the π -part of $q^{d-1} - 1$, for π the set of primes occurring in $q - 1$, and φ is the Euler phi function.

5.2 Computing a transvection basis, k large

If k is large the following can reduce the iteration required in section 3. We use the same notation as in section 3.

After we found the second transvection t_2 , we form $t_3 = t_2^A, \dots$, until some linear relations occurs as in section 3. But now instead of iterating we construct an element B that will leave the subspace M invariant and centralises b . Conjugating our transvection t_i with B will most likely extend our basis.

We look for a conjugate t_1^g of t_1 that fixes b . If a non-trivial linear combination $\sum \alpha_i \pi(t_i)g$ lies in M then the corresponding transvections $B = (\prod t_i^{\alpha_i})^g$ fixes b and normalises M . We now conjugate the given t_i with B .

5.3 Black box recognition

It would not be hard to produce an analogous algorithm that works in a black box group. That is to say, we assume that G is an arbitrary group isomorphic to a group lying between $SL(d, q)$ and $GL(d, q)$. We assume, of course, that we have efficient algorithms to multiply and compare elements of G . We also assume that we have an efficient algorithm to compute the order of any element of G , and that, given an elementary abelian p -subgroup W of G , where p is the characteristic of the field over which G is defined, we can efficiently determine linear relations in W . These conditions would be satisfied if, for example, G was given by some faithful representation over a finite field of characteristic p . It would be possible to adapt our algorithm to work in G , thus getting an isomorphism of some subgroup of G into $GL(d, q)$. The question then arises of computing the image of X in $GL(d, q)$, thus proving that the whole of G is embedded in $GL(d, q)$. We are investigating an elaboration of these ideas into an efficient algorithm.

6 Timings and examples

We have implemented the algorithm in GAP [8] and this implementation is distributed with GAP.

We investigate the group G generated by

$$\begin{pmatrix} 5 & 13 & 4 & 3 \\ 2 & 16 & 7 & \\ & & 1 & 3 \\ & & 16 & 15 \end{pmatrix} \text{ and } \begin{pmatrix} 1 & 14 & 14 & 14 \\ 1 & 15 & 10 & 11 \\ & & 3 & 4 \\ & & & 1 \end{pmatrix}$$

over the field $GF(17)$. Let the matrices act from the right, so the upper left hand corner describes the action on a two-dimensional quotient space. First we investigate the group acting on this quotient space.

```
gap> o:=GF(17).one;
gap> x1:=[[5,13,4,3],[2,16,7,0],[0,0,1,3],[0,0,16,15]]*o;;
gap> x2:=[[1,14,15,14],[1,15,10,11],[0,0,3,4],[0,0,0,1]]*o;;
gap> y1 := [ [ 5, 13 ], [ 2, 16 ] ] * o;;
gap> y2 := [ [ 1, 14 ], [ 1, 15 ] ] * o;;
gap> cr := CRecognizeSL( Group(y1,y2), [y1,y2] );
#I <G> is GL( 2, 17 )
<< constructive SL recognition record >>
```

This show that G acts as $GL(2,17)$ on the quotient space; rewriting two “random” elements we get two elements in the kernel. We now investigate the action of the kernel on the invariant subspace.

```

gap> w1 := Rewrite( cr, y2 );
(t2_1)^4*(t1_1)^3*(t2_1)^7
gap> DisplayMat( (x2) / Value( w1, [x1,x2] ) );
  1 . 7 2
  . 1 14 4
  . . 5 2
  . . 8 6
gap> w2 := Rewrite( cr, y1^2*y2 );
z^2*(t2_1)^4*(t1_1)^3*(t2_1)^7
gap> DisplayMat( (x1^2*x2) / Value( w2, [x1,x2] ) );
  1 . 8 .
  . 1 16 .
  . . 5 10
  . . 5 6
gap> z1 := [ [ 5, 2 ], [ 8, 6 ] ] * Z(17)^0;;
gap> z2 := [ [ 5, 10 ], [ 5, 6 ] ] * Z(17)^0;;
gap> cr := CRecognizeSL( Group(z1,z2), [z1,z2] );
#I <G> is GL( 2, 17 )
<< constructive SL recognition record >>

```

Hence the kernel acts as $GL(2,17)$ on the invariant subspace. Rewriting a “random” element in the kernel shows that G is $17^4 \cdot GL(2,17) \cdot GL(2,17)$ because $GL(2,17)$ acts from both sides on the upper right corner; therefore we either get 17^4 or the trivial subspace in this corner.

```

gap> DisplayMat((xx1)/Value(Rewrite(cr,z1),[xx1,xx2]));
  1 . 14 8
  . 1 11 16
  . . 1 .
  . . . 1

```

The timings in Figure 1 were obtained by running the program on an Intel Pentium P5, 133 Mhz, running FreeBSD 2.1.0. They are an average of 100 runs of running the algorithm using 2 random matrices generating $SL(d, q)$. Note that GAP computes in finite fields using a Zech logarithm table; therefore the time required for one finite field multiplication does not depend on q .

ACKNOWLEDGEMENTS

We are grateful to M. Geck, K. Lux, W. Nickel, and E. A. O’Brien for helpful conversations.

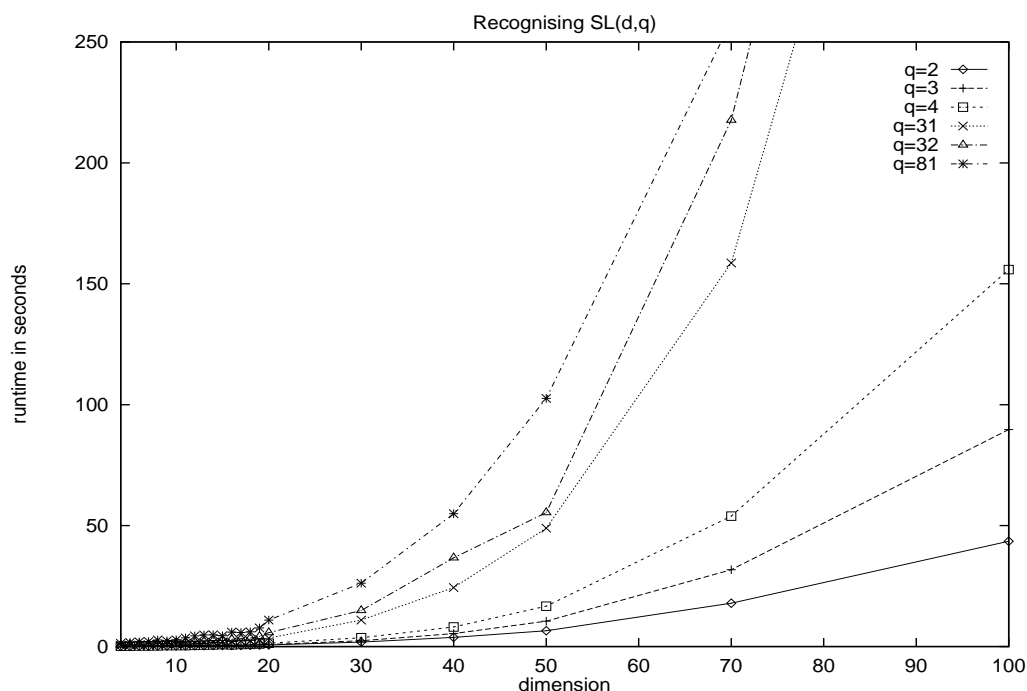


Figure 1: Timings

References

- [1] Frank Celler and C. R. Leedham-Green, Calculating the Order of an Invertible Matrix, DIMACS proceedings, to appear.
- [2] Frank Celler and C. R. Leedham-Green, A Non-Constructive Recognition Algorithm for the Special Linear and Other Classical Groups, DIMACS proceedings, to appear.
- [3] Frank Celler, Charles R. Leedham-Green, Scott H. Murray, Alice C. Niemeyer, and E. A. O'Brien, Generating random elements of a finite group, *Comm. Algebra* **23** (1995), pp. 4931–4948
- [4] Derek F. Holt and Sarah Rees, Testing modules for irreducibility, *J. Austral. Math. Soc. Series A* **57** (1994), pp. 1–16
- [5] Peter Kleidman and Martin Liebeck, *The Subgroup Structure of the Finite Classical Groups*, Cambridge University Press, London Math. Soc. Lecture Note Series **20** (1990)
- [6] Peter M. Neumann and Cheryl E. Praeger, A Recognition Algorithm for Special Linear Groups, *Proc. London Math. Soc. (3)* **65** (1992), pp. 555–603

- [7] Alice C. Niemeyer and Cheryl E. Praeger, A Recognition Algorithm for Classical Groups over Finite Fields, in preparation
- [8] Martin Schönert *et al*, GAP – *Groups, Algorithms and Programming*, Lehrstuhl D für Mathematik, RWTH Aachen, 1994