An Algorithm to Find an Element of SL(d, q) as a Word in its Generators

Elliot Costi

April 2006

1

SL(d, q) is the set of all $d \times d$ matrices over a finite field with $q = p^e$ elements. Elements of a finite field; can be written in two ways. Firstly as powers of a primite element ω (except 0) and secondly as a vector space in *omega* over the prime field with p elements as the following table shows for F_9:

$0- > 0$
$\omega- > \omega$
$\omega^2- > 1 + \omega$
$\omega^3- > 1 + 2\omega$
$\omega^4- > 2$
$\omega^5- > 2\omega$
$\omega^6- > 2 + 2\omega$
$\omega^7- > 2 + \omega$
$\omega^8- > 1$

SL(V) is the set of all linear transformations from the vector space V to itself. If V is ${F_q}^d$, then the natural representation of SL(V) is SL(d, q). Algorithms to find any element A of SL(d, q) as a word in its generators is long established. I produced a similar algorithm that worked in the following way. You take as generators of SL(d, q) the following matrices:

$$t = \begin{pmatrix} 1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

a transvection

$$u = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ -1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

a 2-cycle

$$v = \begin{pmatrix} 0 & -1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & -1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & -1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}$$

an n-cycle

$$\delta = \begin{pmatrix} \omega & 0 & 0 & 0 & \cdots & 0 \\ 0 & \omega^{-1} & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

an element to extend the field to p^e as opposed to just p.

The first step is to continuously multiply A by δ to get 1 in the $(1,1)$ entry. The generators u and v generate the permutation group S_d. With these you can manipulate the matrix A in question to move row i to row 1 and column j to column 1 and then use various combinations of conjugates of t and δ to continuously add multiples of the first row/column to every other entry in first row/column until they are all zero. Working through the matrix A in this way, you will eventually be left with the identity matrix. You will then have $x_1 \dots x_m A x_{m+1} \dots x_n = I$, where the x_i are elements of the generating set. Then you can rearrange the equation to get A in terms of the generators.

However, as has already been said, algorithms to solve this problem have already been found. The idea is to now find a similar algorithm, that will probably also utilize linear algebra to solve this problem, when you are no longer working in the natural representation. So you are still working on $\mathrm{SL}({F_q}^d)$ but the matrices that you have that represent these transformations are of dimension n, where $n > d$.

I have just started to work on this problem and will be attempting to solve it using an idea put forward by my supervisor Charles Leedham-Green. He has given me the general outline of how it should work and it will be left for me to fill in the details. It is this that I will be outlining today.

The first step in order to solve this problem is to look at a specific example. I have taken $n = \binom{d}{2}$

and the representation in question to be the exterior square.

What is the exterior square of a module? You choose a basis $\{v_i\}$ for V, you form the tensor square $V \otimes V$ which is generated by the basis $\{v_i \otimes v_j\}$ and then you quotient out the symmetric elements. That is to say, $v \wedge v = 0$ for all $v \in V$, where $\wedge$ is the symbol you use to denote the product in the exterior square (obviously different from $\otimes$ as $v \wedge v = 0$.

Now consider the subgroup $H \leq \mathrm{SL}(d, q)$. $H = \begin{pmatrix} det^{-1} & 0 & 0 & 0 & 0 & 0 \\ * & & & & & \\ * & & & & & \\ * & & \text{GL(d-1, q)} & & & \\ * & & & & & \end{pmatrix}$

This fixes a 1 dimensional space and is isomorphic to $C_{q^{(d-1)}} \rtimes GL(d-1, q)$. Now we map H from the natural representation to $SL(n, q)$ by a map ϕ. Now, $\phi(H)$ acts reducibly on the underlying vector space ${F_q}^n$ since it has a normal p-subgroup (a theorem from representation theory). The normal p-subgroup in question is $C_{q^{(d-1)}}$. So there is a non-trivial submodule U of ${F_q}^n$. Now, H is maximal and normalises U and so $H = N(U)$. By normaliser we mean $\{g \in \mathrm{SL}(n, q) | gU = U\}$. Now let $W = U^g$. We want to find out the first row of the matrix $g \in SL(n, q)$.

Consider ${g_2}^\alpha, {g_3}^\alpha, \ldots, {g_n}^\alpha \in \mathrm{SL}(d, q)$ and say that these elements are the preimage of $\{I + \alpha\delta_{1i}\} \in \mathrm{SL}(n, q)$, where α is a primitive element of F_q. We want to find $\alpha_2, \alpha_3, \ldots, \alpha_d$ such that $W^{{g_2}^{\alpha_2} \ldots {g_d}^{\alpha_d}} = U$. We then have that $g{g_2}^{\alpha_2} \ldots {g_d}^{\alpha_d} \in H$ and hence we now have the whole problem reduced by a dimension. This process is then repeated on the next dimension down.