Recognising tensor-induced matrix groups

C.R. Leedham-Green E.A. O'Brien

Abstract

We present an algorithm to decide whether or not a matrix group defined over a finite field is tensor-induced.

1991 *Mathematics Subject Classification* (*Amer. Math. Soc.*): 20C20, 20C40.

1 Introduction

We present an algorithm to decide whether or not a matrix group defined over a finite field is tensor-induced.

More precisely, let G be a subgroup of $\mathrm{GL}(d, F)$, where $F = \mathrm{GF}(q)$ and $q = p^e$ for some prime p, and let V be the natural FG-module. We assume that d has a proper factorisation as u^r and seek to answer the following question: does G preserve a decomposition of V as

$$U_1 \otimes U_2 \otimes \cdots \otimes U_r$$

where each U_i has dimension $u > 1$ and $r > 1$, and the set of U_i is permuted by G?

Assume such a decomposition exists, and that v is a vector in V that can be expressed as $u_1 \otimes u_2 \otimes \cdots \otimes u_r$ where $u_i \in U_i$ for $1 \leq i \leq r$; if $g \in G$, then vg can be expressed as $v_1 \otimes v_2 \otimes \cdots \otimes v_r$, where, for some permutation σ_g of $\{1, \ldots, r\}$ independent of v, the map $u_i \mapsto v_{i\sigma_g}$ is a projective linear map of U_i onto $U_{i\sigma_g}$. The map $g \mapsto \sigma_g$ is then a homomorphism of G into the symmetric group S_r, and so $GZ/Z \leq \mathrm{PGL}(u, q) \wr S_r$, where Z is the group of scalar matrices of $\mathrm{GL}(d, q)$.

If the map $g \mapsto \sigma_g$ has an intransitive image, then V is tensor decomposable; hence, we may assume that the image of this homomorphism is *transitive.* We also assume that G acts irreducibly on V. For a general treatment of tensor-induced groups, see Kovács (1990).

The algorithm presented here relies heavily on the theoretical frame-work and algorithm developed in Leedham-Green & O'Brien (1997a, 1997b) for finding a tensor decomposition of a finite-dimensional module over a finite field, or proving that no non-trivial tensor decomposition of this module exists. We first recall the concept of equivalence of tensor products.

Definition 1.1 *A u-tensor decomposition of V is a linear isomorphism from $U \otimes W$ onto V, where U and W are fixed vector spaces, with U of dimension u. If α and β are u-tensor decompositions of V, they are equivalent if there are linear automorphisms ϕ and ψ of U and W respectively such that $\alpha = \beta(\phi \otimes \psi)$.*

Leedham-Green & O'Brien (1997a) showed that there is a one-to-one correspondence between the set of G-invariant projective geometries on V and the set of equivalence classes of tensor decompositions of V as G-module. In particular, a tensor decomposition of V as $U \otimes W$, where U has dimension u, corresponds to a u-projective geometry on V, whose *flats* are defined to be the subspaces of V corresponding to spaces of the form $U \otimes X$ for X a subspace of W. Thus a flat in this projective geometry has dimension, as F-space, a multiple of u, and if the decomposition is a tensor product of G-spaces, then the set of flats is invariant under the action of G. We also presented an algorithm which, given a subspace $\mathcal{F}$ of V, determines whether or not $\mathcal{F}$ is a flat in a G-invariant u-projective geometry on V, and in the affirmative case, returns the corresponding tensor decomposition of V. We then exploited this geometrical approach and some other ideas to provide a practical algorithm to decide tensor decomposability.

In Section 2, we investigate the kernel of the action of a tensor-induced group on the set of tensor factors.

The first step of our tensor-induced algorithm attempts to rule out the possibility that G preserves a tensor-induced decomposition of V by considering the restrictions imposed by the *projective orders* of elements of G; that is, the orders of their images in $\mathrm{PGL}(d, q)$. This test is discussed in Section 3.

Random elements of a group are a central component of our algorithm. In Section 4, we discuss a variation of the product replacement algorithm of Celler *et al.* (1995) to obtain random elements of a group G. Further we present an algorithm to obtain random elements of the normal closure in G of a subset of elements.

If a tensor-induced decomposition exists, then, as we observed above, we assume that G acts transitively on the set of r tensor factors. Hence there exists a homomorphism from G onto a transitive subgroup of S_r. By considering first smaller values of r (and so larger values of u) we reduce to the case where G acts primitively on the set of factors. In summary, we consider homomorphisms from G onto a *primitive* subgroup of S_r, and construct such mappings, or prove that none exists. The construction of such homomorphisms is discussed in Section 5, where we present a low-index subgroup algorithm for *black-box* groups having an *order oracle.*

For each subgroup of G having index r, we now decide whether or not it preserves a tensor decomposition of V as $U \otimes W$, where U has dimension u and W has dimension u^{r-1}; if so, we decide whether or not W can be decomposed into $r - 1$ tensor factors of dimension u in such a way that the resulting set of r u-dimensional tensor factors is permuted by G. This is described in Section 6.

The algorithm is described in Section 7. Finally, we report on the performance of an implementation of the algorithm in Magma (Bosma, Cannon and Playoust, 1997).

One motivation for this work lies in its application to the on-going matrix group "recognition" project. Aschbacher (1984) classified the subgroups of $\mathrm{GL}(d,q)$ into nine categories. Tensor-induced groups constitute one category. A potentially useful first step in studying a matrix group is to determine at least one of its categories in this classification. Much of the recent work on this topic was stimulated by the algorithm of Neumann & Praeger (1992) to recognise the special linear group in its natural representation over finite fields.

It may be worth considering the qualitative differences that arise between proving that a matrix group is tensor-induced as opposed to induced (or equivalently imprimitive). An algorithm to decide membership in the latter Aschbacher category is presented in Holt, Leedham-Green, O'Brien & Rees (1996).

One difference is that the degree of the base representation for tensor-induced groups is at most $\log_2 d$, whereas for induced groups it is at most $d/2$. Further, it is impossible, as we shall see in Lemma 2.3, for a group to act faithfully modulo scalars on the set of tensor factors, whereas a group can act faithfully modulo scalars on the set of blocks. On the other hand, we have no easy means of deciding whether or not the putative tensor decomposition exists on the strength of a given non-scalar element of the group that would have to preserve all of the tensor factors, whereas in the induced case it is easy to complete the calculation given a non-scalar element that would have to preserve all the blocks. This reflects the fact in the induced case one can construct the set of blocks explicitly, whereas in the tensor-induced case what is being permuted is a set of projective geometries, and a projective geometry in this context is a set of subspaces of V that is in general too large to enumerate.

2 The kernel of the action

In Section 6.2, we shall assume that the following hypothesis is satisfied.

Hypothesis 2.1 *Let G be an irreducible tensor-induced subgroup of $\mathrm{GL}(V)$, preserving the tensor decomposition $V = \otimes_1^r U_i$ and permuting the set of tensor factors U_i primitively. If G_S is the subgroup of G that fixes each element of a subset S of $\{1, \ldots, r\}$ in the induced permutation representation of G, then G_S acts irreducibly in its induced action on $\otimes_{i \in S} U_i$.*

We now show why it is reasonable to assume this hypothesis. We start with the case when S is a singleton.

Lemma 2.2 *Let G be an irreducible tensor-induced subgroup of $\mathrm{GL}(V)$, preserving* *the tensor decomposition $V = \otimes_1^r U_i$ and permuting the U_i transitively. Let G_1 be the subgroup of G that fixes the tensor factor U_1. Then G_1 acts irreducibly on U_1.*

Proof. Let W_1 be a G_1-invariant subspace of U_1. For $1 \leq i \leq r$, let g_i be an element of G that takes U_1 to U_i. Let $W_i \leq U_i$ be the image of W_1 under g_i. Clearly W_i depends only on i, and $\otimes_1^r W_i$ is a G-invariant subspace of V. It follows that $W_1 = 0$ or $W_1 = U_1$ as required. □

We now move towards the case of general S.

Lemma 2.3 *Let G be an absolutely irreducible tensor-induced subgroup of* $\mathrm{GL}(V)$*, acting primitively on the set of tensor factors. Let K be the kernel of the action on the set of tensor factors. Then K contains a non-scalar matrix.*

Proof. Assume that K only contains scalar matrices, and hence is the group of scalars in G. Let the tensor decomposition in question be $V = \otimes_1^r U_i$. Then G/K is a primitive subgroup of S_r, and hence is either A_r or S_r or has order less than 4^r (Babai (1981); Cameron (1981); Praeger & Saxl (1980)). Now V has dimension $d = u^r$ for some $u > 1$. But G acts absolutely irreducibly on V, and hence spans the whole of $M_d(q)$, which has dimension d^2; so G has order at least 4^r. Hence G/K is either A_r or S_r.

It now follows that the inverse image in G of the stabiliser of a point in the action of G on the set of tensor factors is a covering group of A_{r-1} or of S_{r-1}. If $r \geq 9$, these groups do not have faithful representations of degree less than $r-2$ (Kleidman & Liebeck (1980, p. 186, Prop. 5.3.7)), so V must have dimension at least $(r-2)^r$, which again violates the condition that the order of G cannot be less than the square of the dimension of V.

This same condition shows that, for a counter-example, we need to have $2(r!) \geq 2^{2r}$, which is equivalent to having $r \geq 8$, and this assumes that U_1 has dimension 2. But neither A_8 nor S_8 has a faithful linear or projective representation of degree less than 4, and since $2(r!) < 4^{2r}$ for $r = 8$ the lemma is proved. □

Now K as above is a normal subgroup of G, containing a non-scalar matrix. It follows from Clifford's theorem (Huppert, 1967, p. 565) that if K acts reducibly on V, then G is either reducible, or imprimitive, or tensor indecomposable, or acts semilinearly with respect to the action of an extension field of $\mathrm{GF}(q)$. It is not unreasonable to exclude these possibilities, since they would give rise to more elementary descriptions of G. But if K acts irreducibly on V, then *a fortiori* G_S acts irreducibly on $\otimes_{i \in S} U_i$.

3 An element order test

As a first step, we compute the projective orders of some random elements of $G \leq \mathrm{GL}(d, q)$. This may rule out the possibility that G is tensor-induced, or produce some constructive information.

Assume that $d = u^r$, and that we are examining the possibility that G is tensor-induced from a subgroup of index r.

If $g \in G$ has projective order n, then $\mathrm{GL}(u, q) \wr S_r$ contains an element h of projective order n. Further, the image of h in S_r has order k for some k that both divides n and is the order of an element of S_r, and n/k must be the least common multiple of the projective orders of r elements of $\mathrm{GL}(u, q)$. If no such k can be found, then G is not tensor-induced from a subgroup of index r. If no k can be found for which n/k satisfies the stronger property of being the projective order of an element of $\mathrm{GL}(u, q)$, then some power of g would have to act as a scalar on one tensor factor and as a non-scalar on another tensor factor. Such an element is called a *projectivity* (Leedham-Green & O'Brien, 1997a). The characteristic polynomial of a projectivity that acts as a scalar on a tensor factor of dimension u is a u-th power. Hence we obtain a powerful negative test.

Given generators for a subgroup K of $\mathrm{GL}(d, q)$ that preserves a tensor decomposition of the natural module, and a projectivity in K, it is easy to find the corresponding tensor decomposition if K acts irreducibly on the tensor factor on which it does not act as scalars. However, at this stage in the computation we do not have a candidate for K and hence cannot employ this observation.

Leedham-Green & O'Brien (1997b) present an algorithm to decide whether or not $\mathrm{PGL}(u, q)$ has an element of a given order.

4 Random elements

Given a generating set X for a group G, we wish to construct random elements of G which are close to the uniform distribution and nearly independent. Similarly if Y is a subset of G, we wish to construct random elements of $N = \langle Y \rangle^G$.

4.1 Random elements of a group

Babai (1991) proposed a general theoretical solution to the problem of constructing random elements of a group G. Let n be an upper bound for the order of G. His algorithm constructs a sequence of $O(\log n)$ elements in $O(\log^5 n)$ multiplications. By taking random subproducts of this sequence, nearly uniformly distributed random elements can now be obtained in $O(\log n)$ multiplications for each element.

We use a variation of the product replacement algorithm presented in Celler *et al.* (1995) to generate random elements of G. This variation was developed in part to address the poor performance of the product replacement algorithm for a certain family of examples. For a discussion of these cases and a report on other aspects of the algorithm, see Pak (2000); for a comparison of the performance of this variation with the original, see Baddeley *et al.* (2000).

The algorithm is designed for the case where G is described by a generating set X, and we have no convenient canonical form for the elements of G.

The algorithm is the following. We initialise an array S of length M of elements of G, where $M \geq |X|+1$. Initially S contains the elements of X and is padded out with copies of the identity element 1_G. An additional element of G is stored in a variable T, the *accumulator*; initially T is set to 1_G.

The *basic operation* of the algorithm picks $i \neq j$ in the range $[1, \ldots, M]$; it now carries out step A and then step B, or, with equal probability step B and then step A, where these steps are defined as follows.

(A) Replace $S[i]$ by $S[i]S[j]^{\pm 1}$.

(B) Replace T by $TS[i]^{\pm 1}$.

The two choices of exponent are random and independent. There are other obvious minor variations; for example, multiplying at random on the right or left.

The algorithm as defined is symmetrical: the probability of moving from one given state to a second in a single step is equal to the probability of moving from the second to the first in a single step.

We now use classical Markov process theory to prove that T converges exponentially fast to the uniform distribution on G. That is to say, if $p_t(g)$ is the probability that $T = g$ for some $g \in G$ after t iterations of the basic operation, then $|p_t(g) - 1/|G|| < e^{-\alpha t}$ for some $\alpha > 0$.

A Markov process is *homogeneous* if the probability of moving from one state to another in one step is time independent. We now restrict our attention to homogeneous Markov processes having finitely many states. Such a process is *irreducible* if the probability of moving from any one state to any other state in t steps is positive for some t. It is *aperiodic* if given two states, when the process has arrived at the first state, the set of values of t for which there is a positive probability that it will arrive at the second state after t further steps has greatest common divisor 1. It is *doubly stochastic* if it satisfies the following condition. If $p(i,j)$ is the probability that the homogeneous process will move in one step from state i to state j, then $\sum_i p(i,j) = 1$, where the summation is over all states i. By the very nature of a Markov process, $\sum_j p(i,j) = 1$, where again the summation is over all states.

The Perron-Frobenius theorem (see, for example, Grimmett & Stirzaker (1982, p. 134)), applies to Markov processes having finitely many states: If a Markov process is homogeneous, irreducible, aperiodic and doubly stochastic, then it converges exponentially fast to the uniform distribution.

A state for our Markov process is a value of the array S and the variable T that can be reached simultaneously from the original configuration. Clearly a necessary condition that the values taken by (S, T) constitute a state of the process is that the values of S generate G. It is an intriguing open question, under what circumstances this condition is also sufficient (see Pak (2000)).

The fact that our process is homogeneous and irreducible is now clear. It is doubly stochastic since it is symmetric. To prove that it is aperiodic, we argue as follows. It is sufficient to prove that it is possible to get from some state to another by two different paths of lengths differing by 1. To do this, it suffices to find two states such that we can move from the first to the second by two paths, one of length 1 and one of length 2. We may assume that the first state is $S = [g_1, g_2, \ldots, g_M]$ and $T = h$ where $g_j = 1_G$ for some j, and the second state has S unaltered, but $T = hg_i$ for some $i \neq j$. The process can move from the first of these states to the second in two ways as follows. The first path has length one:

- $S[i] := S[i]S[j] = g_i; \quad T := TS[i] = hg_i;$

The second has length 2:

- $S[j] := S[j]S[i] = g_i; \quad T := TS[j] = hg_i.$
  $S[j] := S[j]S[i]^{-1} = 1_G; \quad T := TS[j] = hg_i.$

It follows, from the Perron-Frobenius theorem, that the process converges exponentially fast to the uniform distribution.

To use this result, we need to know something about the set of states.

Lemma 4.1 *If a state is represented by a given set of values for S and T, then there is another state in which S is unchanged, but T is replaced by an arbitrary element of G.*

Proof. It is clearly sufficient to consider the case in which some $S[i]$ is 1_G. But then T can be multiplied by $S[j]$ for any $j \neq i$ without altering S. □

Theorem 4.2 *The value of T converges exponentially fast to the uniform distribution on G.*

Proof. The probability that after time t the process has arrived at the state $S[i] = g_i$ for all i, and $T = h$, converges exponentially to the uniform distribution. But we have just seen that the set of possible values for $[g_1, g_2, \ldots, g_M]$ is independent of h. □

In practice, we carry out a *preprocessing step* by executing the basic operation a number of times. Whenever a random element of G is required we now execute the basic operation again and return the resulting value of T as the random element of G. Hence, we assume much more than is proved. One obvious disadvantage of the technique is that the elements returned are not independent of each other. In particular, ensuring that the algorithm is symmetric assists the analysis, but may impact negatively on performance, especially independence.

4.2 Random elements of a normal subgroup

Let Y be a subset of $G = \langle X \rangle$ and assume we wish to construct random elements of $\langle Y \rangle^G$.

Our algorithm is the following. We initialise array S and accumulator T as in Section 4.1, where S is now required to have length M, where $M \geq |X| + 2$. We also have a second array U of length O, where $O \geq |Y| + 2$, and another accumulator V; initially U contains the elements of Y padded out with copies of 1_G and V is set to 1_G.

The *basic operation* is to pick $i \neq j$ in the range $[1, \ldots, M]$ and $k \neq \ell$ in the range $[1, \ldots, O]$, and to perform each of the steps (A), (B), (C), (D) in some order, where (A) and (B) are defined in Section 4.1, and (C) and (D) are as follows.

(C) $U[k] := U[k](U[\ell]^T)^{\pm 1}$;

(D) $V := VU[k]^{\pm 1}$.

In each case the exponent ± 1 is chosen independently at random.

The order in which these steps are taken is chosen with equal probability from 8 possibilities: first A and B in either order, and then C and D in either order; or C and D in either order, and then A and B in either order.

Defining the states of the process to be the values that S, T, U, V can reach simultaneously from the initial configuration, one proves, as in Section 4.1, that this process is homogeneous, irreducible, aperiodic and doubly stochastic.

We need an analogue of Lemma 4.1.

Lemma 4.3 *If a state is represented by a given set of values for S, T, U, V, then there is another state in which S and U are unchanged, but T and V are replaced by arbitrary elements of G and $\langle Y \rangle^G$, respectively.*

Proof. It is sufficient to prove this when both S and U have two values set equal to 1_G, and in this case the result is clear. □

It follows as before that the value of V converges exponentially fast to the uniform distribution on $\langle Y \rangle^G$. In practice, after performing the basic operation a number of times as a preprocessing step, we take successive values of V as random elements of $\langle Y \rangle^G$.

5 Subgroups of low index in black-box groups

The concept of a black-box group was introduced in Babai & Szemerédi (1984). In this model, group elements are represented by bit-strings of uniform length; the only group operations permissible are multiplication, inversion, and checking for equality with the identity element.

We assume that an *order oracle* is available – namely, we can determine efficiently the order of an element. Matrix groups defined over finite fields are covered by this model; in this case, the order of an element can be computed using the algorithm of Celler & Leedham-Green (1997).

If G is tensor-induced, then there exists a homomorphism from G onto a transitive subgroup of S_r, and so G has a subgroup of index r. We want to construct all such homomorphisms or equivalently construct representatives of all conjugacy classes of subgroups of index r.

For our application, r is a *small* integer; a realistic upper bound for r is 5 for $u \geq 3$, and 8 for $u = 2$.

5.1 The general strategy

We assume that we are given a black-box group $G = \langle X \rangle$, a small positive integer r, and an order oracle for elements of G. The aim is to provide a set of subsets of G in one-to-one correspondence with the set of conjugacy classes of subgroups of G of index r, each subset generating a group in the corresponding class.

The standard low-index subgroup algorithm described in Sims (1994) achieves this aim when G is given as a finitely-presented group. For each conjugacy class representative H having index r in G, it returns a homomorphism of G into S_r, which is defined in terms of the image of each element of X in S_r, and H is the inverse image in G of a point stabiliser.

If the low-index subgroup algorithm is used with a set of relations for G that is not defining, then the output will contain generating sets for representatives of all conjugacy classes of subgroups of G of index r, possibly including repetitions, and subgroups of larger index.

An important observation is that the relations used in the low-index subgroup algorithm to obtain subgroups of index at most r do not need to be satisfied by G, but rather by G/K where K is a normal subgroup contained in the intersection of the kernels of all homomorphisms of G into S_r.

Since the relations we construct are not in general defining, we may obtain both some subgroups of larger index and some repetitions.

We now discuss the individual components of this strategy in more detail and in Section 5.4 summarise the resulting algorithm.

5.2 Laws in S_r

How do we construct a normal subgroup of G contained in the intersection of the kernels of all homomorphisms of G into S_r? In practice, we construct a generating set for a subgroup K of K_r, the verbal subgroup of G corresponding to the variety generated by S_r, by evaluating instances of some known laws of the variety.

Cossey, Macdonald & Street (1970) present bases for the defining laws for the varieties generated by S_4, S_5 and A_6. Laws (not defining) for S_6 may be deduced from those for A_6. For $r \geq 7$, we take two laws: namely x^e, where e is the exponent of S_r, and

$$[x_1^{a_1}, (x_1^{a_2})^{x_2}, \ldots, (x_1^{a_{t+1}})^{x_{t+1}}]$$

where $a_1, \ldots, a_{t+1}$ are the multiplicatively maximal orders of elements of S_r.

The laws for S_4 are short and hence evaluating instances of these is efficient. Those for S_5 and A_6 consist of eleven short laws, and a "u-law" introduced in Kovács & Newman (1966). This last law is too long to be of computational value, and appears to be needed solely to exclude certain infinite simple groups. We observe that law (10) in the basis for S_5 given by Cossey *et al.* (1970) is inaccurate; for example, it fails for $x = (1, 5, 4, 3)$ and $y = (1, 3, 4)(2, 5)$.

5.3 Estimating element orders in a quotient group

Let G be a black-box group having an order oracle, and let N be a normal subgroup of G. We now discuss how to estimate the order of an element of G/N.

Our algorithm returns a multiplicative upper bound to the order of an element of G/N. Let g be an element of G and let m be its order in G. We wish to estimate the order of the image of g in G/N. The algorithm iterates the following operation for some preassigned number of times.

- $a :=$ random element of N;
- $m := \gcd(m, |ga|)$;

It then returns m as the estimate of the order of the image of g in G/N.

Clearly, m is a multiple of the order of g modulo N. If g is an element of N then m should eventually become 1, if only by waiting until a becomes g^{-1}. On the other hand, if G is the quaternion group of order 8, and N is the centre of G, and $a \in G \setminus N$ then a has order 2 modulo N, but the algorithm will return 4.

To overcome the latter problem, we refine as follows. If the algorithm returns $m > 1$, then for every prime p dividing m, apply the algorithm to $g^{m/p}$. If the algorithm returns 1 or any number prime to p as the order of the image of $g^{m/p}$, then the order of the image of g divides m/p; now repeat this refinement with m replaced by m/p.

Babai & Shalev (2000, §4.4) prove that this algorithm, with high probability, returns the order of g modulo N as 1 if $g \in N$, a simple normal subgroup of G. Hence it can be used to decide membership of a simple normal subgroup. Their result immediately implies the correctness of the order returned by the refined algorithm when N is a simple normal subgroup of G.

Refinements introduced by Celler & Leedham-Green (1997) can also be exploited here.

5.4 The low-index subgroup algorithm

Given $G = \langle X \rangle$ and an integer $r > 1$, we construct a set S of subsets of G such that every subgroup of G of index r is conjugate to the subgroup of G generated by an element of S. Some subsets in S may generate subgroups of G of index larger than r, and some pairs of subsets may generate conjugate subgroups. The algorithm is the following.

1. Construct a set Y of elements of the verbal subgroup of G corresponding to the variety generated by S_r. This we do by evaluating instances of known laws for the variety in random elements of G.

2. For each w in a random subset T of G, find a multiplicative upper bound m_w to the order of w modulo K, where $K = \langle Y \rangle^G$. We use the algorithm of Section 4.2 to obtain random elements of K, and the order algorithm of Section 5.3.

3. Hence we obtain a presentation of a group Q with generating set $\bar{X}$, the image of X in Q, and relators $\{w^{m_w} : w \in T\}$, where every $w \in T$ is expressed as a word in $\bar{X}$. Now Q is a preimage of G/K. Apply the low index algorithm to obtain homomorphisms from Q to S_r, defined by the images of $\bar{X}$ in S_r. Lift those having primitive images to maps from X to S_r, thus obtaining putative homomorphisms from G to S_r.

4. We now decide the validity of these putative homomorphisms. For each map from X to S_r constructed as above, we first compute a subset R of G whose normal closure in G generates the kernel of the corresponding homomorphism if this exists. We then estimate the orders of various "short" random words in X in $G/\langle R \rangle^G$. If the order of the image of the random element does not divide the estimated order of the element in $G/\langle R \rangle^G$, then we do not have a homomorphism.

5. For each putative homomorphism which passes this test, we now obtain a generating set for the inverse image in G of the stabiliser of a point.

Hence we construct a set S of subsets of G such that every subgroup of G of index r is conjugate to the subgroup of G generated by an element of S.

In practice, we introduce one important refinement to the algorithm presented here: we apply the low-index subgroup algorithm to Q with a bound to the number of subgroups we are prepared to construct. If the number of subgroups constructed exceeds this bound, we abort the computation, determine additional relations and reapply the low-index subgroup algorithm. We hope to impose sufficient relations to ensure that the number of subgroups constructed is less than this bound. If adding further relations for a specified number of iterations does not reduce the number of subgroups, then we terminate.

6 Constructing the decomposition

6.1 The first step

We discuss how to decide whether or not G preserves a tensor decomposition of V with factors of dimensions u and u^{r-1}, for some $r > 1$.

Given $H \leq \mathrm{GL}(V)$, our algorithm for determining whether or not H preserves a tensor decomposition of V as the tensor product of two spaces only requires in the first instance a supply of random elements of H. If the algorithm succeeds, it proves that no such tensor decomposition of V exists; or we construct a change-of-basis matrix $x \in \mathrm{GL}(V)$ and constructively demonstrate that it corresponds to a suitable tensor factorisation of V by demonstrating that conjugating each element of the given generating set of H by x reduces the element to a Krönecker product of the required shape.

If H is defined as the normal closure of a given subset of $G \leq \mathrm{GL}(V)$, we cannot directly apply this last constructive step; instead, we apply this step to some preassigned number of random elements of the normal closure of H; if they have the required shape, we proceed on the assumption that H does preserve the corresponding tensor decomposition; if this assumption is false, it will become clear later.

Step (2) of the algorithm outlined in Section 5.4 constructs a normal generating set Y for a subgroup K of the verbal subgroup of G defined by certain laws of S_r. We try to determine whether or not $K = \langle Y \rangle^G$ preserves a tensor decomposition of V with factors of dimensions u and u^{r-1}. If K does *not* preserve such a tensor decomposition, then G is not tensor-induced and the algorithm terminates.

If K preserves such a tensor decomposition, it remains to decide whether or not G is tensor-induced from a subgroup of index r.

In trying to determine whether or not K preserves a suitable tensor decomposition of V, we apply only the three "fast" tests of the tensor product algorithm (Leedham-Green & O'Brien, 1997b): element order, characteristic polynomial structure, and existence of projectivities. Since K may be small, we do not attempt to construct local subgroups of K. We hope to eliminate elementary cases readily: G is not tensor-induced, or G is tensor-induced, and it is easy to find an element of K that acts as a scalar on precisely one of the tensor factors. Since we apply only "fast" tests, we may be unable to decide whether or not K preserves a suitable tensor decomposition of V.

If we have not reached a definite conclusion, we apply the remaining steps of low-index subgroup algorithm of Section 5.4 to construct subgroups of G having index at least r.

Now, for each putative homomorphism, we obtain a generating set for the inverse image of the stabiliser of a point and supply this generating set to the tensor product algorithm. If none preserves a suitable tensor product of V, then G is not tensor-induced and the algorithm terminates. Otherwise, we obtain a tensor decomposition of V as $U \otimes W$ where U has dimension u and W has dimension u^{r-1}.

6.2 Completing the decomposition

Suppose now that we have a tensor decomposition of V as $U \otimes W$, where U has dimension u and W has dimension u^{r-1} for some $r > 1$. We need to determine whether or not W can be decomposed into $r-1$ tensor factors of dimension u in such a way that the resulting set of r u-dimensional tensor factors of V is permuted by G. The tensor decomposition of V is defined by a point P of the form $\langle \mu \rangle \otimes W$ in the corresponding u-projective geometry.

Let G_1 be the subgroup of G that preserves the decomposition $U \otimes W$; this decomposition is preserved by $H < G$ where H has index at least r in G. Of course, H may be a *proper* subgroup of G_1. It is easy to decide whether or not $g \in G$ is also an element of G_1: we check whether or not the matrix of g written with respect to a basis that exhibits the tensor decomposition is a suitable Krönecker product.

The membership test for G_1 provides the basis for an obvious algorithm to obtain representatives of distinct cosets of G_1 in G. We look for r such coset representatives, $1 = g_1, g_2, \ldots, g_r$. If we cannot find r representatives, then G_1 has index less than r in G (and so H is indeed a proper subgroup of G_1). This seems unlikely to occur in practice.

We may now assume that G_1 has index at least r in G. Hence we can compute the permutation action of an element g of G on these cosets. To find the image of the coset $G_1 g_i$, find, by trial and error, a value of j for which $g_i g g_j^{-1}$ preserves the original tensor decomposition of V as $U \otimes W$. Then g sends $G_1 g_i$ to $G_1 g_j$. If, for some g, no such j exists, then G_1 has index larger than r in G. Conversely, if every element g of the given generating set of G gives rise in this way to a permutation σ_g of $\{1, \ldots, r\}$, then G_1 has index r in G as required.

At this point, we know that G permutes a set of r distinct tensor decompositions of V. We need to determine whether or not there is a tensor decomposition of V as $\otimes_1^r U_i$ giving rise to this situation.

If the tensor decomposition we have found at this stage is $V = U_1 \otimes W_1$, we wish to find recursively tensor decompositions of V as $U_1 \otimes U_2 \otimes \cdots \otimes U_i \otimes W_i$ where U_j has dimension u for $1 \leq j \leq r$. Assume that we have a point P_i of dimension u^{r-i} in a projective geometry defining the tensor decomposition $V = (U_1 \otimes U_2 \otimes \cdots \otimes U_i) \otimes W_i$, where $P_1 = P$. Now define $P_{i+1} := P_i \cap P^{g_{i+1}}$.

For $1 \leq j \leq r$, let H_j be the subgroup of G that maps to the subgroup of S_r that fixes each element of $\{1, \ldots, j\}$. We can readily find a generating set for H_j. We use Hypothesis 2.1 to deduce that H_j acts irreducibly on $U_1 \otimes U_2 \otimes \cdots \otimes U_j$.

Hence, we have a proper subspace P_{i+1} of V, and a generating set for H_i acting irreducibly on V. We now use the algorithm of Leedham-Green & O'Brien (1997a, §3) to decide whether or not V has a u^{i+1}-projective geometry that is preserved by H_i and which has P_{i+1} as a point. If so, we have now constructed the tensor decomposition $V = (U_1 \otimes U_2 \otimes \cdots \otimes U_{i+1}) \otimes W_{i+1}$.

If we obtain a decomposition of $V = \otimes_1^r U_i$, we must now prove that G permutes the set of tensor factors. For each $g \in X$, we compute as described above the permutation action of g on the cosets of G_1, thus obtaining the image σ_g of g in S_r. We now write g with respect to a basis for V that exhibits V as $\otimes_1^r U_i$, and multiply the matrix for g by a matrix which permutes the U_i as σ_g^{-1}; if the resulting matrix is a Krönecker product of the appropriate form, it visibly preserves the tensor factorisation $V = \otimes_1^r U_i$.

7 The tensor-induction algorithm

The input to the algorithm is a set of matrices which generates an irreducible subgroup G of $\mathrm{GL}(d, q)$ and an integer $r > 1$, where d has a factorisation as u^r. A top-level outline of the algorithm is the following.

1. Apply the order test of Section 3 to a small number of random elements of G. This may rule out the possibility that G is tensor-induced.

2. If not, construct a set Y of elements of the verbal subgroup of G defined by certain laws of S_r.

3. Attempt to decide if $K = \langle Y \rangle^G$ preserves a tensor decomposition of V as $U \otimes W$, where U has dimension u. There are three possible outcomes.

   (a) "No": G is not tensor-induced from a subgroup of index r. Return *false*.

   (b) "Yes": Attempt to deduce that G is tensor-induced from a subgroup of index r using the algorithm outlined in Section 6. If this construction succeeds, return the resulting decomposition of V as $\otimes_i^r U_i$. Alternatively, K may preserve some other tensor decomposition of V. Go to Step 4.

   (c) "Unknown": Go to Step 4.

4. Use the low-index subgroup algorithm of Section 5.4 to construct representatives $K_1, \ldots, K_n$ for each conjugacy class of maximal subgroups of index r in G.

5. For each K_i, decide if it preserves a tensor decomposition of V as $U \otimes W$ where U has dimension u. If so, decide if this gives rise to a decomposition of V as $\otimes_i^r U_i$, demonstrating that G is tensor-induced from a subgroup of index r; if it does, return the decomposition of V as $\otimes_i^r U_i$.

   If no K_i preserves such a tensor decomposition of V, or the algorithm of Section 6.2 fails for each K_i which does preserve such a decomposition, then G is not tensor-induced. Return *false*.

7.1 Limitations of the algorithm

The tensor-induced algorithm relies heavily on our algorithm to determine whether or not a subgroup K of $\mathrm{GL}(d,q)$ generated by a given set of matrices preserves a tensor decomposition of the natural module V.

If K does not preserve a tensor decomposition of V, we expect in many cases to prove this in approximately $O(d^3 \log q)$ field operations. If there is a tensor decomposition in which some non-scalar element of K acts as a scalar on one of the tensor factors, we also expect to find the tensor decomposition in $O(d^3 \log q)$ field operations. There are two cases which may pose difficulties.

1. Every element of K preserves some tensor decomposition of V as $U \otimes W$ where U has dimension u, but K preserves no such decomposition.

2. K preserves a tensor decomposition of V as $U \otimes W$ with U of dimension u, but no non-scalar element of K acts as a scalar on U or on W.

In these cases, the tensor product algorithm constructs a p-local subgroup H of K, for various primes p, and looks for a flat among the H-invariant subspaces of V. If n is the least index of a p-local subgroup H of K, finding a p-local subgroup requires at least $O(d^3\sqrt{n})$ field operations. The most expensive part of the algorithm is usually computing the lattice of H-submodules of V.

For example, $Sz(8)$ has an absolutely irreducible representation in dimension 4 over GF(8). Tensor induction gives an absolutely irreducible representation of $Sz(8):3$ acting on a space V of dimension 64 over GF(8). The four maximal subgroups of $Sz(8)$ have orders 14, 20, 52 and $2^6 \cdot 7$. All act on V with composition length 64. The algorithm constructs the 2-local subgroup H of $Sz(8)$ of order $2^6 \cdot 7$, and finds a flat among the H-invariant subspaces of V. This is a particularly hard example, since H is small, and so there are many H-invariant subspaces of V.

8 Implementation and performance

An implementation of our algorithm is publicly available in Magma. The computations reported in Table 1 were carried out using Magma V2.7 on a Sun UltraSPARC Enterprise 4000 server, and all CPU times are given in seconds, averaged over three consecutive executions. Twenty random elements of each group were selected for the order test. For each group, we list its Atlas name (Conway *et al.*, 1985), report its dimension d, and cardinality q of the finite field it is defined over. If the group is tensor-induced, we list the dimensions of the factors; if no decomposition exists, we indicate this by "–". In the final column, we identify the test which either produced the decomposition for this group or proved that it is not tensor-induced.

Group	d	q	Factor	Time	Notes
A_6	16	2	–	0.2	Order test §3
$A_5 \times A_5$	25	7	–	1.5	No subgroup of index 2 §5.4
$3 \cdot J_3 \cdot 2$	36	2^{12}	–	1.5	Characteristic polynomial structure §6.1
$\mathrm{GL}(2,5) \wr S_6$	64	2	2^6	188.1	Low-index §5.4
$\mathrm{GL}(2,2^8) \wr \mathrm{PGL}(2,5)$	64	2^8	2^6	1490.4	Low-index §5.4
$Sz(8):3$	64	2^3	4^3	13148.1	Low-index §5.4

Table 1: Performance of implementation for a sample of groups

Acknowledgements

We are indebted to L.G. Kovács, who has given expert and detailed answers to many technical questions, and significantly influenced the direction of this paper. We are also grateful to László Babai, Peter J. Cameron, Peter M. Neumann, and Cheryl E. Praeger for helpful discussions. We thank the School of Mathematical Sciences of the Australian National University and the Department of Mathematics of the University of Western Australia for their hospitality while part of this work was carried out. This work was supported in part by the Marsden Fund of New Zealand via grant #9144/3368248.

References

M. Aschbacher (1984), "On the maximal subgroups of the finite classical groups", *Invent. Math.*, **76**, 469–514.

László Babai (1981), "On the order of uniprimitive permutation groups", *Ann. Math.*, II. Ser. **113**, 553-568.

László Babai (1991), "Local expansion of vertex-transitive graphs and random generation in finite groups", *Theory of Computing*, (Los Angeles, 1991), pp. 164–174. Association for Computing Machinery, New York.

László Babai and Endre Szemerédi (1984), "On the complexity of matrix group problems, I", *Proc.* 25*th IEEE Sympos. Foundations Comp. Sci.*, pp. 229–240.

László Babai and Aner Shalev (2000), "Recognizing simplicity of black-box groups and the frequency of p-singular elements in affine groups", *Groups and Computation* III, Ohio State Univ. Math. Res. Inst. Publ., (Ohio, 1999). de Gruyter, Berlin.

Adrian Baddeley, C.R. Leedham-Green, Alice C. Niemeyer and Martin Firth (2000), "Measuring the Performance of Random Element Generators in Large Algebraic Structures", in preparation.

Wieb Bosma and John Cannon and Catherine Playoust (1997), "The Magma Algebra System I: The User Language", *J. Symbolic Comput.*, **24**, 235–265.

Peter J. Cameron (1981), "Finite permutation groups and finite simple groups", *Bull. London Math. Soc.* **13**, 1-22.

Frank Celler and C.R. Leedham-Green (1997), "Calculating the order of an invertible matrix", *Groups and Computation II*, Amer. Math. Soc. DIMACS Series, **28**, (DIMACS, 1995), pp. 55–60.

Frank Celler, Charles R. Leedham-Green, Scott H. Murray, Alice C. Niemeyer and E.A. O'Brien (1995), "Generating random elements of a finite group", *Comm. Algebra*, **23**, 4931–4948.

J.H. Conway, R.T. Curtis, S.P. Norton, R.A. Parker and R.A. Wilson (1985), *Atlas of finite groups.* Clarendon Press, Oxford.

John Cossey, Sheila Oates Macdonald and Anne Penfold Street (1970), "On the laws of certain finite groups", *J. Austral. Math. Soc.* **XI**, 441-489.

Geoffrey Grimmett and David Stirzaker (1982), *Probability and Random Processes.* Oxford University Press, London.

Derek F. Holt, C.R. Leedham-Green, E.A. O'Brien and Sarah Rees (1996), "Testing Matrix Groups for Primitivity", *J. Algebra*, **184**, 795–817.

B. Huppert (1967), *Endliche Gruppen I*, Grundlehren Math. Wiss., **134**. Springer-Verlag, Berlin, Heidelberg, New York.

Peter Kleidman and Martin Liebeck (1990), *The Subgroup Structure of the Finite Classical Groups*, London Math. Soc. Lecture Note Ser., **129**. Cambridge University Press, Cambridge.

L.G. Kovács (1990), "On tensor induction of group representations", *J. Austral. Math. Soc. Ser. A*, **49**, 486–501.

L.G. Kovács and M.F. Newman (1966), "On critical groups", *J. Austral. Math. Soc*, **6**, 237–250.

C.R. Leedham-Green and E.A. O'Brien (1997a), "Tensor Products are Projective Geometries", *J. Algebra*, **189**, 514–528.

C.R. Leedham-Green and E.A. O'Brien (1997b), "Recognising tensor products of matrix groups", *Internat. J. Algebra Comput.*, **7**, 541–559.

Peter M. Neumann and Cheryl E. Praeger (1992), "A recognition algorithm for special linear groups", *Proc. London Math. Soc.* (3), **65**, 555–603.

Igor Pak (2000), "What do we know about the product replacement algorithm?", *Groups and Computation* III, Ohio State Univ. Math. Res. Inst. Publ., (Ohio, 1999). de Gruyter, Berlin.

Cheryl E. Praeger and Jan Saxl (1980), "On the orders of primitive permutation groups", *Bull. London Math. Soc.* **12**, 303-307.

Charles C. Sims (1994), *Computation with finitely presented groups.* Cambridge University Press.

School of Mathematical Sciences
Queen Mary and Westfield College
University of London
London E1 4NS
United Kingdom

C.R.Leedham-Green@qmw.ac.uk

Department of Mathematics
University of Auckland
Private Bag 92019
Auckland
New Zealand

obrien@math.auckland.ac.nz

Last revised September 2000