Recognising tensor products of matrix groups

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

Abstract

As a contribution to the project for recognising matrix groups defined over finite fields, we describe an algorithm for deciding whether or not the natural module for such a matrix group can be decomposed into a non-trivial tensor product. In the affirmative case, a tensor decomposition is returned. As one component, we develop algorithms to compute p-local subgroups of a matrix group.

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

1 Introduction

In Leedham-Green & O'Brien (1996), we give an internal description of tensor decompositions of finite-dimensional vector spaces. We do this by constructing a family of projective geometries whose flats are certain subspaces of a vector space V, and showing that there is a one-to-one correspondence between this family of projective geometries and the set of equivalence classes of tensor decompositions of V.

The object of this paper is to exploit the geometrical approach presented there, together with some other ideas, to provide a practical method for finding a tensor decomposition of a finite-dimensional module over a finite group algebra, or proving that no non-trivial tensor decomposition of this module exists.

Let G be a subgroup of $GL(d, K)$, where $K = GF(q)$ and $q = \chi^e$ for some prime χ, and let V be the natural KG-module. (We use "χ" as the characteristic of K since we reserve "p" for p-local subgroups.) We assume that d has a proper factorisation as uw and seek to decide whether or not G preserves a tensor decomposition of V as $U \otimes W$, where U has dimension u and W has dimension w. In more detail, the central product $GL(U) \circ GL(W)$ acts naturally on $U \otimes W$, and we say that G preserves a tensor decomposition of V as $U \otimes W$ if there is an isomorphism of V onto $U \otimes W$ such that the induced image of G in $GL(U \otimes W)$ lies in $GL(U) \circ GL(W)$. Thus the corresponding action of G on the tensor factors U and W may be a projective rather than a linear action.

Our algorithm does *not* require that G act irreducibly on V. Instead, we require that G act irreducibly on *one* of the factors in the decomposition. For notational convenience, we denote this factor by W. Note however that the components of the

algorithm presented in Sections 2 and 4, which seek only to rule out the existence of a tensor decomposition, do not require that G act irreducibly on either of the factors; again for notational convenience, in those sections we assume that $u \leq w$.

One motivation for this work lies in its application to the on-going matrix group "recognition" project. Aschbacher (1984) classified the subgroups of $GL(d, q)$ into nine categories. One possible 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. More recently, Niemeyer & Praeger (1996a, 1996b) and Celler & Leedham-Green (1996a) propose recognition algorithms for the classical groups in their natural representations over finite fields. The MEATAXE algorithm (Parker, 1984; Holt & Rees, 1994) can decide if a KG-module is irreducible, or absolutely irreducible. Irreducible matrix groups which act imprimitively can be recognised using an algorithm of Holt, Leedham-Green, O'Brien & Rees (1996). Here we consider the category where a matrix group preserves a tensor decomposition of its underlying vector space.

Given a matrix group G of degree d, we begin by attempting to rule out that G preserves a tensor decomposition of V as $U \otimes W$, where U has dimension u and W has dimension w and $d = uw$, by considering the restrictions placed on tensor decompositions of V by the *projective orders* of elements of G; that is, the orders of their images in $PGL(d, q)$. To this end, we present in Section 2 an algorithm to find the least k such that $PGL(k, q)$ has an element of given order n. In the order test, we randomly select elements from G and use their projective orders to rule out certain pairs (u, w).

We refine this test by using the characteristic polynomial $f(x)$ of $g \in G$ to decide if the action of g on V is compatible with a tensor decomposition of V as the tensor product of spaces of dimensions u and w. If $f_1(x) = \prod_i (x - \lambda_i)$ and $f_2(x) = \prod_j (x - \mu_j)$ are monic polynomials over some field K, where the λ_i and μ_j lie in some extension field of K, their *tensor product* $f_1 \otimes f_2$ is $\prod_{ij}(x - \lambda_i \mu_j)$. The coefficients of $f_1 \otimes f_2$ obviously lie in K. Clearly a necessary condition for the desired factorisation to take place is that $f(x)$ should be the tensor product of a monic polynomial of degree u and a monic polynomial of degree w. In Section 4, we present an algorithm to test whether a given polynomial has such a tensor factorisation. In the polynomial factorisation test, we randomly select elements from G and use their characteristic polynomials to rule out certain (of the remaining) pairs (u, w).

If, for enough randomly selected elements of G, the order test and the polynomial factorisation test fail to rule out certain pairs (u, w), we are reasonably confident that G preserves such a tensor decomposition of V and we now use our geometrical approach to resolve the matter.

In Leedham-Green & O'Brien (1996), we show that there is a one-to-one correspondence between the set of G-invariant projective geometries and the set of equivalence classes of tensor decompositions of V as G-module. In particular, a tensor decomposition of V as $U \otimes W$ 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. (Naturally there is a dual w-projective geometry.) Thus a flat in this projective geometry has dimension, as K-space, a multiple of u, and if the decomposition is a decomposition into a tensor product of G-spaces, then the set of flats is invariant under the action of G. We also present an algorithm which, given a subspace F of V, determines whether or not 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.

How do we obtain a flat in our putative u-projective geometry? A *u-projectivity* of V onto itself is a non-scalar linear map $g : V \to V$ that acts as a scalar on U, where V is equated with $U \otimes W$. In Leedham-Green & O'Brien (1996, §4) we show how a flat can be constructed from such a projectivity. It may be that we find an element of G whose order dictates that if V has a u-projective geometry, then g must act as a u-projectivity. If so, we construct a flat in this geometry, or find a tensor decomposition over an extension field of K, or decide that there is no such decomposition. We discuss this projectivity test in Section 3.

An alternative approach to finding a flat is the following. Recall that we assume that G acts irreducibly on W. Let H be a subgroup of G that acts reducibly on W. Then at least one of the H-invariant subspaces of V is a non-trivial flat in the corresponding u-projective geometry. One natural class of such subgroups are p-local subgroups: namely, groups contained in the normaliser of some non-trivial p-group, for p a prime. In Section 6, we present two algorithms for constructing such subgroups. Both use as a central component a method presented in Section 5 for computing the stabiliser in G of a subspace of a vector space. Ideally we would like to construct maximal p-local subgroups, particularly for p being the characteristic of K. In practice, for given u and w, we only need the p-local subgroup H to have the following properties: it acts reducibly on the w-dimensional tensor factor in every tensor decomposition of the given module for these dimensions, and the number of H-submodules of V is "reasonably" small. In Section 6, we also consider whether the p-local subgroups constructed satisfy these properties and mention some problems encountered.

We provide an overview of the algorithm in Section 7, and consider its complexity.

Finally, we report on the performance of an implementation. In summary, our implementation performs reasonably well for matrix groups of "moderate" degree. A first version of our implementation is distributed with the computational algebra systems GAP (Schönert *et al.*, 1994) and MAGMA (Bosma & Cannon, 1994).

2 The order test

We present an algorithm to determine the smallest value of d such that $PGL(d, q)$ contains an element of order n.

We first consider the semi-simple case, where n is prime to q. As a preliminary step, we compute the least integer d such that $GL(d, q)$ has an element g of order n. Let V be the natural module for $GL(d, q)$. Then V is a direct sum of indecomposable

$\langle g \rangle$-modules, so $V = V_1 \oplus \cdots \oplus V_t$. Let n_i be the order of g restricted to V_i, hence n is the least common multiple of the n_i. By the minimality of d, the dimension d_i of V_i is minimal subject to $GL(d_i, q)$ having an element of order n_i. Since the set of element orders of a group is closed under taking factors, we may assume that the n_i are pairwise coprime and hence n is the product of the n_i. Let π be a coprime factorisation of n, say $n = u_1 u_2 \ldots u_{t_\pi}$, and define d_i to be the least integer such that $GL(d_i, q)$ has an element of order u_i which acts irreducibly. Now d_i is simply the multiplicative order of q modulo u_i. Hence for each coprime factorisation π of n, we obtain a sum $d_1 + \cdots + d_{t_\pi}$. Then d is the least of these sums. We record, for reasons explained below, the largest value of t_π corresponding to a factorisation π of n that gives rise to d.

To determine the smallest value of d in the semi-simple case, it suffices to find the least d such that $GL(d, q)$ has an element of order n, and then to decide whether $GL(d, q)$ has an element of projective order n. For if we construct an element of $GL(d, q)$ of order n acting on the direct sum of two or more indecomposable blocks, then clearly $PGL(d, q)$ has an element of order n. Otherwise, the only possibility for $GL(d, q)$ to have an element of projective order n is for the element to act irreducibly, and this will require n to divide $(q^d - 1)/(q - 1)$ rather than just $q^d - 1$, and if this condition fails, then the minimum degree required is $d + 1$. This is why we needed to record whether or not it was possible to achieve the required order with more than one irreducible block.

We now consider the case where n is a multiple of the characteristic χ of $GF(q)$. Write $n = \chi^\alpha m$, where $\chi \not| m$. In this case, a composition factor of the natural module must occur with multiplicity at least $\chi^{\alpha-1} + 1$. The most economical way to achieve this is to have a composition factor of size one repeated this number of times. So if $GL(d, q)$ has an element of order m, then $GL(d + \chi^{\alpha-1} + 1, q)$ has an element of order n. However, if it is possible to find an element of $GL(d, q)$ of order m with one composition factor of dimension one, then we can repeat this composition factor $\chi^{\alpha-1} + 1$ times, and save one in the dimension; that is, $GL(d + \chi^{\alpha-1}, q)$ has an element of order n. If d is minimal subject to $GL(d, q)$ having an element of order m, and if an element of $GL(d, q)$ of order m has a composition factor of dimension 1, then $d = 1$.

Clearly a necessary condition for $PGL(d, q)$ to have an element of order n is that $GL(d, q)$ should have such an element, g say. If g can be chosen so as to act on the natural module as a direct sum of more than one indecomposable, then clearly $PGL(d, q)$ has an element of order n, since the set of orders of elements of an abelian group is closed under the formation of least common multiples. Thus $GL(d, q)$ has, but $PGL(d, q)$ does not have, an element of order n, for n a multiple of χ, if and only if $n = \chi^\alpha m$ for some $\alpha > 0$, where $m \neq 1$ and m divides $q - 1$.

In summary, we define the following algorithms.

Algorithm 1 *LeastLinearSemiSimple* (n, q)

/* Given n prime to q, it returns the least d such that $GL(d, q)$ has an element of order n, and the largest t such that this can be achieved with g acting on the natural module as the direct sum of t non-zero submodules. */

for each factorisation π of n as $\prod_{i=1}^{t_\pi} u_i$, where the u_i are relatively prime
define $d_{i,\pi}$ to be the order of q modulo u_i;
put $d_\pi = \sum_{i=1}^{t_\pi} d_{i,\pi}$;
put $d = \min_\pi d_\pi$;
put $t = \max_{d_\pi = d} t_\pi$;
return (d, t);

end

Algorithm 2 *LeastProjectiveSemiSimple* (n, q)

/* Given n prime to q, it returns the least d such that $PGL(d, q)$ has an element of order n. */

if $n = 1$ return 1;
put $(d, t) = LeastLinearSemiSimple\ (n, q)$;
if $t > 1$ return d;
if $(q^d - 1)/(q - 1) \bmod n = 0$ return d;
return $d + 1$;

end

Algorithm 3 *LeastProjective* (n, q)

/* It returns the least d such that $PGL(d, q)$ has an element of order n. */

put $n = \chi^\alpha m$, where χ is the characteristic of $GF(q)$ and $\chi \not| m$;
if $\alpha = 0$ return $LeastProjectiveSemiSimple\ (n, q)$;
put $factor = \chi^{\alpha-1} + 1$;
if $m = 1$ return $factor$;
return $factor + LeastLinearSemiSimple\ (m, q)$;

end

Suppose now that we find $g \in G$ of projective order n. What does this tell us about possible tensor factorisations? There are three possible outcomes. If $d = uw$, with $1 < u \leq w$, then a tensor decomposition of the natural module V as $U \otimes W$, with U and W of dimensions u and w, is clearly possible if $\textsc{LeastProjective}(n, q) \leq u$, and is impossible if n has no coprime factorisation as $n_1 n_2$ with $\textsc{LeastProjective}(n_1, q) \leq u$ and $\textsc{LeastProjective}(n_2, q) \leq w$. In the remaining case, the decomposition is possible, but only if some non-scalar power of g acts as a scalar on one of the tensor factors; we now use the projectivity test to resolve the question.

3 The projectivity test

The algorithm described in Leedham-Green & O'Brien (1996, §4) for constructing a flat from a projectivity takes as input a putative u-projectivity, g, in some unspecified G-invariant u-projective geometry on V. It constructs a flat in this projective geometry and hence finds a tensor decomposition over K, or it finds a tensor decomposition over an extension field of K, or it decides that g is not a projectivity. It is encoded as the procedure IsProjectivity.

Assume that, by applying the order test, we find $g \in G$ whose projective order is not the order of an element of $PGL(u, q)$, where $d = uw$ and $u \leq w$. In this case, a tensor decomposition is only possible if some power of g acts as a scalar on U – that is, as a projectivity. So for some prime p dividing n, we send $g^{n/p}$ to IsProjectivity. If no tensor decomposition results, then the projective order of the action of g on both U and W must be a multiple of p^α, where α is the p-adic value of n. If there is no factorisation of $p^\alpha n$ as $n_1 n_2$ where the greatest common divisor of n_1 and n_2 is p^α and LeastProjective$(n_1, q) \leq u$ and LeastProjective$(n_2, q) \leq w$, then the desired tensor factorisation of V does not exist. Otherwise, we take another prime ℓ dividing n, send $g^{n/\ell}$ to IsProjectivity, and either find a tensor decomposition or another prime power that must divide n_1 and n_2. Repeating, we either find a tensor decomposition, or prove that no tensor decomposition of the required dimensions exists.

4 The polynomial factorisation test

We are now in the situation where we have at least one factorisation of d as uw where $u \leq w$, and the projective order n of every selected random element g of G satisfies the condition LeastProjective$(n, q) \leq u$. To decide if the action of g on V is compatible with a tensor decomposition of V as the tensor product of spaces of dimensions u and w, we now need a more sensitive invariant than the order of g and consider the characteristic polynomial, say $f(x)$, of g. As we observed earlier, a necessary condition for the required tensor decomposition to exist is that $f(x)$ should be the tensor product of a monic polynomial of degree u and a monic polynomial of degree w. We now show how to decide if $f(x)$ has such a factorisation.

Although not needed in the present context, the formula for forming the tensor product of two polynomials with indeterminate coefficients can be readily computed. If f is a monic polynomial, let $S_i(f)$ denote the sum of the i-th powers of the zeros of f, including multiplicities. The Newton and Waring formulae provide polynomial expressions for $S_i(f)$ in terms of the coefficients of f, and also for the coefficients of f as polynomials in $S_i(f)$ for $1 \leq i \leq d$, where d is the degree of f. Details of these formulae can be found, for example, in Macdonald (1995). Now observe that $S_i(f) = S_i(f_1)S_i(f_2)$ if $f = f_1 \otimes f_2$. Combining this information solves the problem.

In practice, to compute the tensor product of two polynomials when the coefficients are given members of the ground field, it is more efficient to write down the companion matrices of the two factors, form their tensor product as matrices, and compute the

characteristic polynomial of this matrix.

A discussion of the uniqueness of tensor factorisation and related topics is best conducted in the context of ring theory. In order to construct a ring with the tensor product of two polynomials being their product in the ring, we need to extend the definition of tensor product to formal power series. For this to be possible, a slight change in the construction is necessary. We now take polynomials with constant term 1 rather than monic polynomials. If f and g are polynomials over the field K with constant term 1, and zeros $\alpha_1, \ldots, \alpha_m$ and $\beta_1, \ldots, \beta_n$ respectively, including multiplicities, define $f \odot g$ to be the polynomial with constant term 1, and zeros $\{\alpha_i \beta_j\}$, including multiplicities. The advantage of this definition is that it extends by continuity to the set $K_1[[x]]$ of formal power series over K with constant term 1. This now becomes a ring, where the addition operator is the usual multiplication of polynomials, and the multiplication operator is $\odot$. The zero of this ring is 1 and the identity is $1 - x$.

It is easy to see that the ring is not an integral domain. For example, let a be a positive integer, and consider the polynomial $f(x) = (1-x+x^2/2) \odot (1-ax^2)$ in $\mathbb{Q}_1[[x]]$. This polynomial has as its zeros $\{\pm b \pm ib\}$, where b is a square root of a^{-1}. Replacing a by $-a$ does not change this set, so $f(x)$ is also equal to $(1 - x + x^2/2) \odot (1 + ax^2)$. It follows that $1 - x + x^2/2$ is a zero divisor in $K_1[[x]]$ for every field K of characteristic not equal to 2.

The relationship between the products $\otimes$ and $\odot$ may be explained as follows. If f and g are polynomials of degrees m and n respectively, with non-zero constant term, and leading coefficients a and b respectively, define $f \otimes g$ to be $a^n b^m (f/a) \otimes (g/b)$. Then $f \odot g = (-1)^{mn} f \otimes g$.

The ring $K_1[[x]]$ has long been of interest to algebraic geometers; more generally, they consider Λ-rings, where analogues of exterior powers and tensor products arise. The semi-group of polynomials under tensor products and the uniqueness of tensor factorisation – which fails in general – are considered in Brawley & Carlitz (1987) and Glasby (1995).

We now revert to the question of determining whether or not the characteristic polynomial $f(x)$ of g has a tensor factorisation of the required form. This we do by successively trying all possible factors of degree u and testing if one of these tensor-divides $f(x)$. Since $u \leq w$, this should be more efficient than testing all possible values for the factor of degree w. We now require an algorithm that will determine whether or not a given monic polynomial with non-zero constant term will tensor-divide another such polynomial. One possible approach is to compute the formula for the tensor product of two polynomials of the given degrees, and use this as a means of obtaining quotients, if they exist. However, in our case a faster method is available.

Suppose that g is of projective order n, and that g^n is the scalar ϕ, so that the irreducible factors of $f(x)$ are irreducible factors of $x^n - \phi$; say $f(x) = \prod_{i=1}^{r} f_i(x)^{\alpha_i}$. Now g^n must act as some scalar, say θ, on U, and as $\phi\theta^{-1}$ on W. Let $u(x) = \prod_{j=1}^{s} u_j(x)^{\beta_j}$, where the $u_j(x)$ are the irreducible factors of $x^n - \theta$, and let $w(x) = \prod_{k=1}^{t} w_k(x)^{\gamma_k}$, where the $w_k(x)$ are the irreducible factors of $x^n - \phi\theta^{-1}$, and $u(x)$ and $w(x)$ are the characteristic polynomials of g acting on U and W respectively. Now $u_j(x) \otimes w_k(x) = \prod_i f_i(x)^{c_{ijk}}$

where $f_i(x)$ runs through the irreducible factors of $x^n - \phi$, and we compute these c_{ijk}. We observe that

$$\alpha_i = \sum_{j,k} \beta_j \gamma_k c_{ijk},$$

and these equations in the γ_k have a solution over $\mathbb{N}$ if and only if the required tensor factorisation of $f(x)$ exists. Our method is, then, to try all possible values for the β_j subject to the condition that $\sum_j \beta_j \deg(u_j(x)) = \deg(u(x))$ where $\deg(u_j(x))$ is the degree of $u_j(x)$, stopping if we find a solution over $\mathbb{N}$ for the γ_k. If we find no solution over $\mathbb{N}$, then the required tensor factorisation does not exist. A useful improvement to the above method may be obtained by observing that we may multiply the action of $u(x)$ on U by some scalar λ and divide its action on W by λ, without harm. Thus θ need only vary over the representatives of $K^*/(K^*)^n$. Also, if $\nu \in K$ is a maximal m-th root of 1 where m divides the order of g, then we prune the possible choices for $u(x)$ so that no two differ by a tensor factor $(x - \nu)$.

How do we decide whether a system of linear equations has a solution over $\mathbb{N}$? This is, of course, a well-known problem in integer programming.

Consider the matrix, A say, of the system of linear equations which determine the γ_k. If A has zero nullity (for example, if $u(x)$ is not a zero divisor), then the equations either have a unique solution over $\mathbb{Z}$, or they are inconsistent over $\mathbb{Z}$. But sometimes we find an integral solution that has negative terms, and A has non-zero nullity.

One approach to deciding whether there is a solution over $\mathbb{N}$ to the system of equations is outlined in Adams & Loustaunau (1994, pp. 105–107). In summary, they translate the integer programming problem into an equivalent problem about polynomials, by letting the unknown non-negative variables be exponents of variables in a polynomial ring. They then compute a Gröbner Basis for the ideal generated, and use this basis to solve the polynomial problem. We have used an implementation of this technique in MAGMA to test its feasibility. The time taken to compute the Gröbner Basis grows exponentially with the number of variables and this computation is sometimes too expensive a component of our overall algorithm.

On the other hand, Gaussian Elimination allows us to decide quickly whether there is a solution to the system over $\mathbb{Z}$. Hence, on performance grounds, we solve the system first over $\mathbb{Z}$. There are three possible outcomes. First, if there is no solution, then we know that the polynomial has no tensor factorisation for this choice of $u(x)$. Second, our solution may be unique or it may already be over $\mathbb{N}$. Finally, if the solution obtained is over $\mathbb{Z}$ and it is not unique, we have various options: decide whether there is a solution for this $u(x)$ over $\mathbb{N}$ by using the Gröbner Basis (or some other) approach; search for a conclusive factorisation of $f(x)$ involving a different $u(x)$; or terminate the test and select a new element.

The alternating group of degree 8 in dimension 28 over $GF(11)$ is one example where the non-negative test is effective in deciding that there is no tensor product as 4×7. There is an element of order 15 where, for some possible $u(x)$, the system of linear equations determining the γ_k has a solution over $\mathbb{Z}$, but for no possible $u(x)$, is there a solution over $\mathbb{N}$. The solution over $\mathbb{Z}$ is always unique.

We thus have a test that may reject, but cannot prove, the possibility of a tensor decomposition. The test is quite powerful in practice; Table 1 gives some indication of its performance. If q is large, cases can arise in which the test would be too slow for some group elements. However, it is easy to see in advance if the test will be slow for a particular element, the critical limiter being the number of irreducible factors of $x^n - \phi$, and we can simply decline to carry out the test in such cases.

As a final theoretical point, it may be worth noting that there is a marked difference between the problems which arise with tensor factorisation and standard factorisation of polynomials. If $f(x) \in K[x]$ has degree mn, then $f(x)$ can only be the tensor product of a polynomial of degree m and one of degree n if the coefficients of $f(x)$ satisfy certain integral polynomial equations independent of K. If $x^{mn} - c_1 x^{mn-1} + \cdots + (-1)^{mn} c_{mn}$ has such a tensor factorisation, then $(c_1, \ldots, c_{mn})$ must lie in a graded ideal in $\mathbb{Z}[x_1, \ldots, x_{mn}]$, where x_i has weight i. The reader can verify that $x^4 - c_1 x^3 + c_2 x^2 - c_3 x + c_4$ can only be the tensor product of two polynomials, each of degree 2, if $c_3^2 = c_1^2 c_4$. For $m = 3$ and $n = 2$, it is less easy to verify that a non-zero element of least weight in this ideal is a polynomial of weight 19 and involves 28 distinct monomials. We thank Ruth Schwingel for this discouraging information; see Schwingel (1996) for further details.

5 Computing the stabiliser of a space

The algorithms presented in Section 6 for computing p-local subgroups of a matrix group G use as a central component the following method for computing the stabiliser in G of a subspace U of a vector space V.

A naïve approach to obtaining elements of the stabiliser of U is to construct the image of U under the action of a large number of random elements of G and to record both the image and the element used. If two images coincide, that is $Ug_1 = Ug_2$, then $g_1 g_2^{-1}$ stabilises U. Since we cannot afford to store either a large number of images or random group elements, we modify this approach by initially storing instead a hash address for each image and noting only the number of the random element giving rise to this image. In a second stage, we now resolve all coincidences in the hash table. We reset the random element seed and repeat the random element generation. For each recorded coincidence, we now decide whether the random elements g_1 and g_2 that give rise to this coincidence satisfy $Ug_1 = Ug_2$.

This method can be obviously generalised to find elements of the set-wise stabiliser of a set of subspaces of V.

The *product replacement* algorithm, to construct random elements of a finite group, only requires one multiplication to compute each random element, after an initialisation stage. See Celler, Leedham-Green, Murray, Niemeyer & O'Brien (1995) for details; further analysis of this algorithm can be found in Diaconis & Saloff-Coste (1996).

If the stabiliser of U has index n in G, the number of field operations required to find, with given probability, an element of the stabiliser of U is proportional to $d^3\sqrt{n}$. For we expect to construct about $O(\sqrt{n})$ images of U before finding a coincidence, it

takes $O(d^3)$ field operations to produce a random element of G, and the number of field operations required to compute the image of a basis of U and to reduce this basis to echelon form is $O(ud^2)$, where u is the dimension of U. In practice, we allow n to take values up to about 1 000 000.

As presented, this algorithm can only produce random elements of the stabiliser S of U in G. However, if we have a test for deciding whether or not an element of G lies in a given subgroup of S, then we can modify the algorithm to produce a generating set of S with arbitrarily high confidence. Namely, if H_k is the subgroup of S generated by the first k random elements of S that we find, and if $H_k = H_{k+t}$, then we can assert, after the $(k+t)$-th random element of S has been constructed, that $S = H_k$ with confidence at least $1 - (1/2^t)$. In specific examples, an easy membership test is often available; if a precise test is not practical, then a statistical profile of, for example, the orders of elements may lead to a reasonably effective test. We may also be in a situation – for example, if S is almost simple – where S has very few subgroups of small index, and so we also expect that every subset of S with significantly more elements than are contained in a minimal generating set for S will generate the whole of S.

6 Constructing p-local subgroups

We present algorithms for constructing p-local subgroups of a matrix group G, consider when these subgroups act reducibly on at least one factor in a putative tensor decomposition of the underlying vector space, and mention some problems associated with our approach. These subgroups are used as input to the reducible subgroup component of our algorithm.

While our primary interest here is in producing large p-local subgroups of G, we believe that modified versions of these algorithms, combined with a membership test similar to that discussed in Section 5, would allow us to construct, with a high level of probability, a generating set for the normaliser of a given p-subgroup of G.

6.1 p-local subgroups for $p = \chi(K)$

Given a p-subgroup P of G, where p is the characteristic of the ground field K, we wish to construct a subgroup H of G with a normal p-subgroup that is generated, as normal subgroup, by P.

The theoretical basis for the algorithm is the following easy result.

Lemma 6.1.1 *Let P be a p-subgroup of $G \leq GL(d, p^e)$. Then P^G is a p-group if and only if V has a series, as G-module, whose factors are centralised by P.*

Proof. If P^G is a p-group, define $V_0 = \langle 0 \rangle$, and define V_i for $i > 0$ so that V_i/V_{i-1} is the subspace of V/V_{i-1} that is centralised by P^G. Since P^G is a p-group, $V_t = V$ for some t. Since P^G is normal in G, the V_i are G-submodules of V, as required.

Conversely, if V has a series $\{V_i\}$ of the required form, then P^G centralises V_i/V_{i-1} for all i, and hence is a p-group. ‖

Hence, to find a p-local subgroup where $p = \chi(K)$, we proceed as follows. Given a non-scalar p-group P, we compute a series of P-submodules of V whose factors are centralised by P, and take the p-local subgroup to be the subgroup H of G that stabilises the terms in this series. In practice, we use the series $\{U_i\}$, where $U_0 = \langle 0 \rangle$, and for $i > 0$ we define U_i so that U_i/U_{i-1} is the subspace of V/U_{i-1} that is centralised by P.

To compute H, we first compute the stabiliser H_1 of U_1 in G, then the stabiliser H_2 of U_2 in H_1, *etc.* Elements of the stabiliser S of a subspace of V may be constructed as described in Section 5. We can obtain elements of S in other ways. First, we may have an element of the normaliser of P given in advance, and this will certainly stabilise each of the subspaces constructed. Second, we are likely to find, when computing the stabiliser H_i of U_i in H_{i-1}, that the whole of H_{i-1} stabilises U_i; this is easy to check. This is bound to occur if, as we hope, H is a maximal subgroup of G.

We exploit these ideas as follows. We first find a random element of G of order p. This is done by constructing random elements until we find an element k whose minimum polynomial has a repeated factor. This is equivalent to k having order a multiple of p. We then compute the order, pm say, of k, and take $g = k^m$. We now have a p-group $P = \langle g \rangle$, and an element k of its centraliser, and we construct the p-local subgroup H as described above.

Recall that we construct H because we wish to search among the H-submodules of V for potential flats in a G-invariant geometry. Hence, we want a p-local subgroup H where the lattice of H-submodules of V is relatively small. In practice, finding a p-local subgroup, when one exists, is faster than computing the relevant lattice. Thus we look for a subgroup H where the H-submodule lattice of V has relatively short composition length, in the expectation that such a lattice has a small number of submodules.

If H is such that the H-submodule lattice of V has composition length greater than some pre-assigned value (or is otherwise unsuitable), we can search in H for another element, h say, of order p that commutes with g but which does not centralise the same H-invariant series as g, and hence is not conjugate in H to g. Then taking $P = \langle g, h \rangle$, we can start again to find a new p-local subgroup. Alternatively, we look (in H) for another element g' of order p that is not conjugate in $GL(d, q)$ to g, and start again with g'. Our experience is that we tend, for obvious reasons, to find larger rather than smaller p-local subgroups, provided that they exist.

6.2 p-local subgroups for $p \neq \chi(K)$

Here we consider the task of constructing p-local subgroups of G for primes p not equal to the characteristic of K.

Our starting point is a random element, h say, of projective order a multiple of p. A suitable power, g say, of h gives us an element of projective order p, and order some power of p. We find a Jordan decomposition of V under the action of g. That is, we express V as a direct sum of $\langle g \rangle$-invariant subspaces $J_1, \ldots, J_t$ such that the minimum polynomial of g acting on J_i is f_i, where the f_i are distinct and irreducible. We take the direct decomposition of V in which each direct summand is the direct sum of all

the J_i for which the corresponding f_i have a given degree. We construct as large a subgroup S of the subgroup of G that stabilises each of these direct summands as we can find. The method used is that described in Section 5.

We now construct as large a subgroup T of the subgroup of S that preserves the set of subspaces $\{J_i\}$ as we can find. This is done by looping over the degrees of the f_i: for each degree ℓ, replace S by as large a subgroup as we can find of the set-wise stabiliser in S of the set of subspaces J_i for which f_i has degree ℓ.

Suppose that all the f_i have degree 1; that is, if g has order p^ℓ, then K contains a primitive p^ℓ-th root of 1. Then g and all its T-conjugates are diagonal when written with respect to a basis that exhibits the Jordan decomposition of V. Hence, $\langle g \rangle^T$ is an abelian p-group, and T is p-local as required.

If some f_i has degree greater than 1, we write g and the generating matrices for T over a splitting field for the f_i, and apply the above steps again, with T playing the role of G. This involves conjugating the matrices with matrices whose coefficients do not lie in K, but we can then conjugate the new T by the inverse of these matrices to obtain the matrices of the new T in terms of the earlier basis; the resulting matrices have entries in K.

In general, we expect all but one or two of these steps to be trivial, because the complete parent group will already stabilise the space. In particular, all but one of the steps is clearly trivial if we find a p-local subgroup that is maximal in G.

6.3 Testing reducibility of p-local subgroups

Having constructed a p-local subgroup H, with normal p-subgroup P, we use the H-submodules of V of suitable dimension to search for flats giving rise to a tensor decomposition of V. Since our earlier tests have failed to disprove the existence of a tensor decomposition, we are quite hopeful that one exists, in which case it is again likely that H will act reducibly on one, or indeed on both, of the tensor factors. However, we may find that no H-submodule of V is a flat in a G-invariant u-projective geometry on V. If this happens, we wish to prove that if such a tensor decomposition did exist, then H would have to act reducibly on at least one of the factors. If p is the characteristic of K, then this condition is automatic. If not, we can reduce to the case in which P is a minimal normal subgroup of H. If P is cyclic, then the following consequence of Clifford's theorem provides a necessary criterion for H to act irreducibly on W. A proof can be found in Leedham-Green & O'Brien (1996).

Theorem 6.3.1 *If H is a subgroup of G that acts irreducibly on W, and normalises the cyclic group generated by g, where g has projective order p, then the characteristic polynomial of g acting on W is of the form $\prod_i f_i(x)^t$, where f_1 is an irreducible factor of $x^p - \lambda$ for some scalar λ, and the f_i are the conjugates of f_1 under the action of H. That is, if $f_1(x) = \prod_j (x - \theta_j)$, and $h \in H$ conjugates g to g^s for some integer s, then h conjugates $f_1(x)$ to $\prod_j (x - \theta_j^s)$.*

Hence, if H acts irreducibly on W, then the characteristic polynomial of g must have a tensor factorisation involving such a factor. We have developed an effective imple-

mentation of this criterion, which uses a modified version of the algorithm presented in Section 4.

If P is non-cyclic, the structure of the characteristic polynomial of g could perhaps be used to provide similar (if more complex) necessary criteria for H to act irreducibly in the appropriate dimension. In practice, we can generally settle the question by either taking p to be the characteristic of K, or by taking P to be cyclic, or having some other theoretical way of deciding that H would have to act reducibly.

6.4 Problems with this approach

Three central problems arise with the p-local subgroup test. First, every proper subgroup of G may be of such large index that our stabiliser technique for finding a p-local subgroup is ineffective. Second, the lattice of H-submodules of V may be too large. Third, we may have problems in proving that our p-local subgroup acts reducibly on a putative tensor factor.

A typical case in which we cannot hope in practice to find a large p-local subgroup using the stabiliser approach is when G is a large subgroup of a general linear group. For example, if G is $GL(d, q)$, and V is the tensor square of the natural module for G, then the p-local subgroup we need to find is the natural module extended by $GL(d-1, q)$. In an attempt to address this problem, we have developed a prototype algorithm that constructs the relevant subgroup whenever G is isomorphic to a group lying between $SL(d, q)$ and $GL(d, q)$.

Clearly, there are groups whose p-local subgroups are not large enough to provide a suitable submodule lattice. For example, we have frequently encountered problems when G is isomorphic to an alternating or symmetric group since, in this case, a p-local subgroup is likely to leave far too many subspaces of V invariant. We need algorithms to find alternating subgroups on one (or two) fewer points, and prove that these must act reducibly on a putative tensor factor. More generally, we hope to develop algorithms to construct other reducible subgroups.

7 An overview of the algorithm

The input to the algorithm is a set of matrices which generate a subgroup G of $GL(d, q)$. We first compute all proper factorisations of d as uw and store these as a list $\mathcal{L}$.

The steps of the algorithm are the following:

1. Order test;
2. Projectivity test;
3. Polynomial factorisation test;
4. Reducible subgroup test.

The first three steps are each applied to a small number of random elements. In applying these, we either rule out certain of the possible factorisations of d as uw, or in applying the projectivity test find a tensor decomposition. It is possible that the order test generates no elements which are potential projectivities. If this is the case, we call IsProjectivity with a small number of random elements of G which have prime projective order.

As input to the final stage, we now have a (hopefully shorter) list, $\mathcal{L}$, of possible factorisations of d as uw. We construct a small number of reducible subgroups looking for subgroups H which satisfy the following requirements: the composition length of V as an H-module is small and H acts reducibly in at least one of the dimensions that occur in a factorisation in $\mathcal{L}$. Usually the subgroups constructed are p-local for various primes p. Among the resulting collection, we then select the smallest number of subgroups which both permit us to decide all factorisations in $\mathcal{L}$ and give the smallest composition length.

We now process these subgroups in order of increasing composition length. The algorithm for finding a point in our projective geometry from a flat, described in Leedham-Green & O'Brien (1996, §3), is encoded as the procedure FindPoint. For the selected reducible subgroup H, we first compute the lattice of H-submodules of V; each subspace of dimension a multiple of an outstanding u is then handed to FindPoint. If we do not find a tensor decomposition, then we eliminate another possible factorisation from $\mathcal{L}$.

7.1 Complexity of the algorithm

The complexity of the algorithm is a delicate issue. If G does not preserve a tensor decomposition of V, we expect in "most" 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 G 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.

If these simple algorithms fail, the worst case analysis becomes meaningless. In the best case, if n is the least index of a p-local subgroup H of G, 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. If the composition length of V as an H-module is "too large" for a particular H, we do not attempt to compute this lattice.

To justify and clarify our claims on the complexity of the simpler parts of the algorithm, an informal discussion follows. We assume throughout that the time required for a field operation is bounded. The complexity of the order test is the complexity of constructing a random element of G, plus the complexity of computing the order of an element of G. To assert that these have complexity $O(d^3)$ and $O(d^3 \log q)$ respectively are (slightly optimistic) interpretations of the results in Celler *et al.* (1995) and Celler & Leedham-Green (1996b). To compute the characteristic polynomial $f(x)$ of an element g of G has complexity $O(d^3)$. To compute the minimum polynomial has Las Vegas complexity $O(d^3)$; that is, the algorithm will, with probability greater

than $1-(1/2^t)$, terminate in time $O(d^3t)$. Hence, the procedures IsProjectivity and FindPoint each have Las Vegas complexity $O(d^3)$. To decide whether or not a polynomial of degree $d = uw$, with $u \leq w$, is the tensor product of polynomials $u(x)$ and $w(x)$ of degrees u and w respectively requires considering all possible candidates for $u(x)$ where $u(x)$ must be a product of powers of certain irreducible polynomials. If there are k irreducible polynomials in the factorisation of $u(x)$, the exponents of $k-1$ of these determine the exponent of the k-th. Hence, if $k \leq 7$, then the number of candidates for $u(x)$ is at most $O(u^6) = O(d^3)$. The amount of work required to determine whether or not a given candidate for $u(x)$ tensor divides $f(x)$ is also a function of k, as it involves solving approximately k integer linear equations in approximately k unknowns. The numbers here are only approximate, as they are the number of irreducible factors of $x^n - \theta$ for various values of θ, where n is the projective order of g. This estimate also ignores the problem of negative exponents. If the matrix defining the equations has positive nullity, then we may use the Gröbner Basis approach, which has exponential cost in the number of variables, to decide if there is a solution over $\mathbb{N}$. However, since such cases are rare, the cost can "usually" be ignored. So we can run a number of these tests on elements of projective order bounded in theory by 7 in $O(d^3)$ field operations. In practice, we are prepared to consider much larger orders, the critical limiter being the number of irreducible factors of $x^n - \theta$.

8 Implementation and performance

In developing an implementation of our algorithm, we have extensively used existing implementations of other algorithms.

We use the product replacement algorithm to generate random elements of G. For details, see Celler *et al.* (1995).

We compute the order of an element of G using an algorithm of Celler & Leedham-Green (1996b). They have also developed effective algorithms for computing characteristic and minimal polynomials of matrices defined over finite fields.

An algorithm to obtain the lattice of submodules of a KG-module is described by Lux, Müller & Ringe (1994), and an implementation of their algorithm is distributed as a package with GAP. Another algorithm has been developed by Cannon and Leedham-Green and is available in Magma.

Implementations of our algorithm are available in both GAP and Magma. The computations reported in Table 1 were carried out using Magma Version 1.3 on a HP 9000/730, and all CPU times are given in (rounded) seconds. 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, and the finite field it is defined over. If we found a tensor decomposition, then 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 tensor decomposition for this group or proved that it preserves no tensor decomposition.

Since the algorithm has a random component, the times listed should be viewed

Group	Dimension	q	Factorisation	Time	Notes
$A_5 \times A_5$	5^2	7	5×5	1.5	Projectivity test
$6.A_6$	12	7^2	3×4	160.0	3-local subgroup
A_6	16	2	4×4	6.8	2-local subgroup
A_7	20	2	–	0.6	Polynomial factorisation
$6.A_7$	24	11^2	4×6	220.1	3-local subgroup
A_8	20	11	–	1.0	Polynomial factorisation
A_8	28	11	–	2.9	Polynomial factorisation
A_9	21	17	–	0.4	Order test
A_9	28	49	–	3.8	Polynomial factorisation
A_9	168	17	8×21	818.2	A_8 as reducible subgroup
$(3^2 : 4 \times A_6) \cdot 2$	18	7	3×6	0.8	Projectivity test
$(3^2 : 4 \times A_6) \cdot 2$	27	7	3×9	1.3	Projectivity test
$SL(2,5)$	6	7^2	2×3	4.9	3-local subgroup
$SL(2,25)$	16	5^2	4×4	4.9	5-local subgroup
$SL(6,49)$	36	7^2	6×6	34.9	$SL(5,49)$ as reducible subgroup
$SL(7,25)$	49	5^2	7×7	59.1	$SL(6,25)$ as reducible subgroup
$U_4(2)$	30	5^2	5×6	87.1	2-local subgroup
$U_4(2)$	30	7^2	–	4.4	Polynomial factorisation
$U_4(2)$	58	5	–	2.0	Order test
$L_2(13)$	14	7	–	0.1	Order test
$L_2(17)$	18	41	–	0.2	Order test
$L_2(31)$	32	2^4	–	1.1	Order test
$L_2(81)$	82	41	–	10.1	Order test
M_{11}	55	7	–	1.3	Order test
M_{12}	55	7	–	2.0	Order test
M_{22}	54	7	–	1.7	Order test
Co_2	22	2	–	0.1	Order test
Co_3	22	3	–	0.2	Order test
Ru	28	5	–	0.7	Order test
J_1	14	7	–	0.2	Order test
J_1	27	11	–	1.9	Polynomial factorisation
$He.2$	50	7	–	2.7	Order test
$He.2$	102	2	–	18.1	Order test
Ly	111	5	–	4.5	Order test
J_4	112	2	–	2.1	Order test
$Z_4 \times 2.HS$	112	17	2×56	37.8	Projectivity test

Table 1: Performance of implementation for a sample of groups

only as a general guideline. In an attempt to provide a realistic guide to performance, we report the CPU time averaged over three consecutive executions.

We now report in a little more detail on three examples. The first two illustrate that a representation may support more than one distinct tensor decomposition of the same degrees. We thank Stephen Glasby and L.G. Kovács who suggested them.

Example 8.1 The alternating group A_9 has a representation of dimension 168 over various finite fields. This representation is primitive and can be constructed as the tensor product of the representation of degree 8 with either of the two distinct representations of dimension 21. We construct A_8 as a subgroup of A_9 and use A_8 as a reducible subgroup to find both tensor decompositions.

Example 8.2 Let G be the split extension of $GL(2,3)$ acting on its natural module. Then G has exactly 3 irreducibles of dimension 2, and exactly 2 of dimension 8 over $GF(11)$. The tensor product of each 2-dimensional irreducible with each 8-dimensional gives the same irreducible imprimitive 16-dimensional representation. The natural module acts trivially on all three 2-dimensional modules over $GF(11)$. However, $GL(2,3)$ acts faithfully on two of the 2-dimensionals, but as S_3 on the third. Again, we find these distinct factorisations.

Our final example illustrates the tensor factorisation over an extension field discussed in Leedham-Green & O'Brien (1996, §3).

Example 8.3 There is a 45-dimensional representation of $3.O'N$ over $GF(7)$; in fact, there are two classes of such representations, conjugate by an outer automorphism. Restricting to the preimage G of the maximal subgroup $(3^2 : 4 \times A_6) \cdot 2$ of $O'N$ the representation becomes reducible, with a submodule V of dimension 18. If we send a suitable involution to IsProjectivity, we find a tensor decomposition of V as 3×6. However, if we send an element g of order 5 to IsProjectivity and search for a G-invariant 3-geometry, an eigenspace F of dimension 6 of g is sent to FindPoint. The matrices written with respect to the geometric basis constructed for the set of G-images of F in general position are not proportional and all matrices in the algebra of dimension 6 generated by the geometric transformations between flats are zero or non-singular. That is, V is isomorphic to $U \otimes W_0 \otimes GF(49)$ for some 3-dimensional module W_0, where G acts non-trivially on $GF(49)$ and hence acts semi-linearly on V as a $GF(49)$-space.

To conclude, we observe that those representations which pose practical difficulties for the algorithm share certain common features. They are representations of almost simple groups which we can easily decide (from the Atlas or other sources) do not preserve tensor decompositions. Usually we are attempting to eliminate one possible factorisation $d = uw$, where u are w are approximately equal. Every random element considered potentially supports a tensor decomposition, there are no potential u-projectivities, and the p-local subgroups constructed do not provide sufficiently small submodule lattices. As indicated earlier, we plan to address such examples by developing new techniques for constructing other reducible subgroups.

Acknowledgements

O'Brien is supported by the Alexander von Humboldt Foundation, Bonn, which also partially funded a visit by him to the University of London via a European Research Fellowship. Both authors were supported by the European Union HCM Grant for Computational Group Theory. We thank the School of Mathematical Sciences at the Australian National University and the Department of Mathematics, University of Western Australia, for their hospitality while part of this work was carried out. We are grateful to Peter Cameron, Stephen Donkin, Stephen Glasby, L.G. Kovács, I.G. Macdonald, and Allan Steel for helpful discussions.

References

William W. Adams and Philippe Loustaunau (1994), *An Introduction to Gröbner Bases*. Graduate Studies in Math. **3**, Amer. Math. Soc., New York.

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

Wieb Bosma and John Cannon (1994), *Handbook of* Magma *functions*. School of Mathematics and Statistics, Sydney University.

J.V. Brawley and L. Carlitz (1987), "Irreducibles and the composed product for polynomials over a finite field", *Discrete Math.* **65**, 115–139.

Frank Celler and C.R. Leedham-Green (1996a), "A Non-Constructive Recognition Algorithm for the Special Linear and Other Classical Groups", *Groups and Computation* (DIMACS, 1995), Amer. Math. Soc. DIMACS Series.

Frank Celler and C.R. Leedham-Green (1996b), "Calculating the order of an invertible matrix", *Groups and Computation* (DIMACS, 1995), Amer. Math. Soc. DIMACS Series.

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.

P. Diaconis and L. Saloff-Coste (1996), "Walks on generating sets of abelian groups", *Probab. Theory Relat. Fields* **105**, 393–421.

S.P. Glasby (1995), "Tensor products of polynomials", Sydney University School of Mathematics and Statistics Research Report **95-4**.

Derek F. Holt, C.R. Leedham-Green, E.A. O'Brien and Sarah Rees (1996), "Testing matrix groups for primitivity ", *J. Algebra* **184**.

Derek F. Holt and Sarah Rees (1994), "Testing modules for irreducibility", *J. Austral. Math. Soc. Ser. A*, **57**, 1–16, 1994.

C.R. Leedham-Green and E.A. O'Brien (1996), "Tensor Products are Projective Geometries", to appear *J. Algebra*.

Klaus Lux, Jürgen Müller and Michael Ringe (1994), "Peakword Condensation and Submodule Lattices: An application of the Meat-Axe", *J. Symbolic Comput.*, **17**, 529–544.

I.G. Macdonald (1995), *Symmetric functions and Hall Polynomials*, Oxford Mathematical Monographs, Clarendon Press, New York. 2nd edition.

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

Alice C. Niemeyer and Cheryl E. Praeger (1996a), "A Recognition Algorithm for Classical Groups over Finite Fields", preprint.

Alice C. Niemeyer and Cheryl E. Praeger (1996b), "Implementing a recognition algorithm for classical groups", *Groups and Computation* (DIMACS, 1995), Amer. Math. Soc. DIMACS Series.

R.A. Parker (1984), "The computer calculation of modular characters (the Meat-Axe)", M.D. Atkinson (Ed.), *Computational Group Theory*, (Durham, 1982), pp. 267–274. Academic Press, London, New York.

Ruth Schwingel (1996), "The tensor product of polynomials", preprint.

Martin Schönert *et al.* (1994), GAP – *Groups, Algorithms and Programming.* Lehrstuhl D für Mathematik, RWTH Aachen.

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

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

Lehrstuhl D für Mathematik
RWTH
Templergraben 64
52062 Aachen
Germany

obrien@math.rwth-aachen.de