

COMPUTING A CHIEF SERIES AND THE SOLUBLE RADICAL OF A MATRIX GROUP OVER A FINITE FIELD

DEREK F. HOLT AND MARK J. STATHER

Abstract

We describe an algorithm for computing a chief series, the soluble radical, and two other characteristic subgroups of a matrix group over a finite field, which is intended for matrix groups that are too large for the use of base and strong generating set methods. The algorithm has been implemented in MAGMA by the second author.

1. *Introduction*

One of the major projects in computational group theory during the past 15 years has been the development of effective algorithms for analysing the structure of linear groups defined by generating matrices over a finite field. The methods that are currently used by default in the GAP and MAGMA systems are based on an extension of the base and strong generating set (BSGS) techniques that have proved so effective for computing in finite permutation groups. These are unfortunately impractical for large groups of matrices, particularly those that involve the classical groups in their natural representations. (See Chapter 4 and Section 7.8.1 of [17] for definitions and details of the BSGS approach to computing in permutation and matrix groups.)

Leedham-Green and O'Brien [23] have implemented procedures in MAGMA that use methods based on a theorem of Aschbacher, which we shall state in Subsection 1.2, to construct a *composition tree* in a matrix group over a finite field, which effectively identifies the composition factors of the group. See also [29] for a recent survey of this project.

For further structural computations in matrix groups, such as finding Sylow subgroups, a chief series is more useful. As has been demonstrated for various types of computations in finite permutation groups (the calculation of maximal subgroups [9], of automorphism groups [8], and of conjugacy classes of elements [10], for example), a chief series that passes through the characteristic subgroups $O_\infty(G)$, $\text{soc}^*(G)$, and $\text{Pker}(G)$ of G is particularly expedient. These subgroups are defined as follows.

- (i) $O_\infty(G)$ is the soluble radical – that is, the largest normal soluble subgroup – of G .
- (ii) $\text{soc}^*(G)$ is the complete inverse image in G of the nonabelian socle of $G/O_\infty(G)$. So $\text{soc}^*(G)/O_\infty(G)$ is a direct product of nonabelian simple groups, which are permuted under the conjugation action of $G/O_\infty(G)$.

Received 13 July 2007, revised 28 January 2008; *published* 4 August 2008.

2000 Mathematics Subject Classification 20-04, 20H30

© 2008, Derek F. Holt and Mark J. Stather

- (iii) $\text{Pker}(G)$ is the complete inverse image in G of the kernel of the permutation action of $G/O_\infty(G)$ on the simple factors of $\text{soc}^*(G)/O_\infty(G)$ described in (ii). So $\text{soc}^*(G) \leq \text{Pker}(G)$.

Note that $\text{Pker}(G)/\text{soc}^*(G)$ is isomorphic to a subgroup of the direct product of the outer automorphism groups of the simple direct factors of $\text{soc}^*(G)/O_\infty(G)$ which, by the Schreier conjecture, is soluble. Note also that the top factor $G/\text{Pker}(G)$ is isomorphic to a permutation group on the factors of $\text{soc}^*(G)/O_\infty(G)$, which in practice will be of moderately low degree.

It is the object of this paper to describe algorithms to construct such a series. They have been implemented in MAGMA by the second author, and a more detailed description of these methods and of applications to further structural computations in matrix groups can be found in his PhD thesis [31]. See also [32] for an application to the computation of Sylow subgroups. For an input group $G \leq \text{GL}(n, q)$, it is also possible to use the data structures associated with the chief series computed for G to test whether an arbitrary element g of $\text{GL}(n, q)$ lies in G , and if so to write g as a word in the generators of G .

The computations are carried out in three stages. The first step is to compute a series of normal subgroups of the given group G in which the quotients are either soluble groups for which a polycyclic presentation (see, for example, [17, Chapter 8]) is known, or direct products of isomorphic nonabelian simple groups. The second step, which is comparatively straightforward, is to refine the series to a chief series of the group. The final step is to compute a new chief series passing through $O_\infty(G)$, $\text{soc}^*(G)$ and $\text{Pker}(G)$. This is achieved by a sequence of operations, each of which interchanges two adjacent chief factors in the series.

We believe our computation of a chief series of G to be about equally efficient on average, in terms of use of both time and space resources, as the computation of a composition tree using the algorithm of Leedham-Green and O'Brien. As we shall see later, there are some places in our algorithm where we are forced to carry out computations in larger groups than should be necessary, and here we will lose ground in the comparison with the composition tree. In other places, however, we save time by avoiding the unnecessary repetition of certain computations in subgroups that are conjugate to other subgroups in which these computations have already been carried out. We shall indicate instances of these two phenomena in the examples described in Section 4.

In the remainder of this introductory section we shall describe some relevant basic algorithms, and state the Aschbacher Theorem. In Section 2 we describe the computation of a chief series of a matrix group G , in Section 3 we explain how to rearrange the chief series to make it pass through the three characteristic subgroups of G defined above, and then in Section 4 we present some timings of the second author's MAGMA implementation.

1.1. *Some underlying algorithms*

An algorithm to compute the order of an invertible $d \times d$ matrix is described in [12]. Many of the algorithms of the Matrix Group Recognition Project, and indeed of computational group theory in general, require the construction of random elements of a group. An algorithm to produce uniformly distributed random elements of a finite group is given by Babai [5]. Unfortunately this is too slow

to be of practical use. For the purposes of an implementation we use the *Product Replacement Algorithm* [11]. See also [17, Section 3.2.2] for a brief description.

We shall assume throughout that we are able to perform standard computations, including finding chief series, etc., in permutation groups and in soluble groups defined by a polycyclic presentation (see, for example [17, Chapter 8]). The most sophisticated of the algorithms that we shall assume to be available are the *Constructive Recognition* algorithms for the finite nonabelian simple groups.

We formulate the definition of a constructive recognition algorithm following that of Seress [30, Chapter 8]. However we shall adapt the definition slightly to give emphasis on matrix groups. Recall that G is said to be *quasisimple* if G is perfect and $G/Z(G)$ is simple, where $Z(G)$ is the centre of G . We are interested in the case where a quasisimple group G acts absolutely irreducibly, and so $Z(G)$ is the group of scalar matrices $Z(\text{GL}(n, q)) \cap G$ of G .

DEFINITION 1.1. *Let $G = \langle X \rangle$ be a group such that either G is a simple permutation group, or else G is a quasisimple absolutely irreducible matrix group over a finite field. Then we define a constructive recognition algorithm for G to be one that is able to do the following:*

- (i) *Find the standard name of the simple group $G/Z(G)$.*
- (ii) *Find a new generating set Y of size $O(\log |G|)$ for G , along with words over X for each $y \in Y$, and a presentation of length $O(\log^2 |G|)$ of $G/Z(G)$ on Y . (By a word over X , we mean a word in $(X \cup X^{-1})^*$.)*
- (iii) *Compute an epimorphism ϕ from G to S where S is the standard copy of $G/Z(G)$, with the property that images and inverse images of elements under ϕ can be computed efficiently. (We explain the term standard copy below.)*
- (iv) *Given g in the full symmetric or general linear group of which G is a subgroup, determine whether $g \in G$ and, if so, write g as a word over Y .*

A *non-constructive recognition algorithm* is one that can solve part (i) only. A polynomial time Monte-Carlo algorithm for part (i) has been implemented in MAGMA by Malle and O'Brien. This works roughly by calculating orders of random elements of the group and finding the finite simple group having the same distribution of element orders. See [29, Section 6] for a fuller account.

Observe that, although we assumed at the outset that G is quasisimple, after carrying out this process, we have proved that $G/Z(G)$ is isomorphic to the simple group named in (i) above, and hence we have verified that our assumption was correct. (If we need to verify that G is perfect, then we can do so by choosing the generators in Y to be words lying in $[G, G]$.) This is important because, in our applications to be described later, we are sometimes highly confident but not absolutely certain that $G/Z(G)$ is simple and so we need to verify this assumption.

In (iii), the *standard copy* of a nonabelian simple group G is a specific group H that depends only on the isomorphism type of G , where H is either a primitive permutation group with $H \cong G$, or a quasisimple absolutely irreducible matrix group with $H/Z(H) \cong G$. For example, for $G \cong \text{Alt}(n)$ we choose $H = \text{Alt}(n)$, and for $G \cong \text{PSL}(n, q)$ we choose $H = \text{SL}(n, q)$. We assume also (as in the condition (iv) for G) that we can test arbitrary elements of the group $\text{Sym}(n)$ or $\text{GL}(n, q)$ that contains H for membership of H .

The algorithm referred to in part (iv) is known as a solution to the *rewriting problem* in G on Y . In practice we do not store the words that arise as ordinary words over Y , which would be impractically long in many cases, but use *straight line programs* (see, for example [17, Section 3.1.3]) as a more compact storage method. Note that, since from (ii) we can express the elements of Y as words over the original generators X , we can also express arbitrary group elements as words (or rather straight line programs) over X .

Constructive recognition algorithms for the simple groups is currently an active area of research. The alternating groups can be dealt with by the algorithm of Bratus and Pak [6], which has been implemented in MAGMA by Holt. Methods for the classical groups are described in [21]. However, these algorithms have a factor of q in their complexity, making them impractical over large fields. Alternative algorithms are currently under development by Leedham-Green and O'Brien. They rely on the special cases $SL(2, q)$ and $SL(3, q)$ which have been dealt with in [13] and [27] respectively. Algorithms for the Suzuki and Ree groups have been developed by Bäärnhielm in [3] and [2] respectively. Many sporadic groups can be dealt with using the *Ryba algorithm* described in [16].

1.2. Aschbacher's Theorem

The major result upon which the Matrix Group Recognition Project is based is a theorem by Aschbacher [1] on the subgroup structure of the general linear group, which we shall now paraphrase.

THEOREM 1.2 ([1]). *Let V be the vector space of row vectors on which $GL(n, q)$ acts, let G be a subgroup of $GL(n, q)$, and let Z be the group of scalar matrices of G . Then one of the following is true.*

- C1. G acts reducibly.
- C2. G acts imprimitively: G preserves a decomposition of V as a direct sum $V_1 \oplus V_2 \oplus \dots \oplus V_r$ of $r > 1$ subspaces of dimension s , which are permuted transitively by G , and so $G \leq GL(s, q) \wr \text{Sym}(r)$.
- C3. G acts on V as a group of semilinear automorphisms of a space of dimension n/e over the extension field \mathbb{F}_{q^e} for some $e > 1$, and so G embeds in $\Gamma L(\frac{n}{e}, q^e)$. (This covers the class of 'absolutely reducible' matrix groups where G embeds in $GL(\frac{n}{e}, q^e)$.)
- C4. G preserves a decomposition of V as a tensor product $U \otimes W$ of spaces of dimensions $n_1, n_2 > 1$ over \mathbb{F}_q . Then G is a subgroup of the central product of $GL(n_1, q)$ and $GL(n_2, q)$.
- C5. G is definable modulo scalars over a subfield: for some proper subfield $\mathbb{F}_{q'}$ of \mathbb{F}_q , $G^g \leq GL(n, q').Z$ for some $g \in GL(n, q)$.
- C6. For some prime r , $n = r^m$ and G is contained in the normaliser of an extraspecial group of order r^{2m+1} or of a group of order 2^{2m+2} and of symplectic type.
- C7. G is tensor-induced : it preserves a decomposition of V as $V_1 \otimes V_2 \otimes \dots \otimes V_m$, where each V_i has dimension $r > 1$ and the set of V_i is permuted transitively by G , and so $G/Z \leq PGL(r, q) \wr \text{Sym}(m)$.
- C8. G contains and normalises a classical group in its natural representation.
- C9. G is almost simple modulo scalars: for some nonabelian simple group T we have $T \leq G/Z \leq \text{Aut}(T)$.

In short this theorem states that either G is almost simple modulo scalars (in C8 or C9) or else preserves some natural geometric structure. If G lies in C1–C7 then we obtain a natural geometric *directly computable homomorphism* from G to some “smaller” group. We define a directly computable homomorphism to be one that can be evaluated without solving the rewriting problem in G ; the action of a matrix on an invariant subspace, for example.

Many papers have now been published that describe algorithms and their implementations that recognise groups that lie in one or other of the Aschbacher classes. We shall cite these when we come to describe our procedures for handling groups that lie in these classes in Subsection 2.2.

2. Computing a chief series

2.1. Sequences of homomorphisms

As we explained in the introduction, our first aim is to construct a normal series in our input group $G \leq \text{GL}(n, q)$ in which the quotients are either soluble groups or direct products of isomorphic nonabelian simple groups. This series will be represented by a sequence of homomorphisms.

DEFINITION 2.1. *We define a preliminary sequence for $G \leq \text{GL}(n, q)$ to be a sequence (ϕ_1, \dots, ϕ_k) of homomorphisms and a descending sequence $G = K_0 \geq K_1 \geq K_2 \cdots \geq K_k \geq 1$ of normal subgroups of G that satisfy the following properties:*

- (i) $K_{i-1} \trianglelefteq \text{Domain}(\phi_i) \leq \text{GL}(n, q)$ for $1 \leq i \leq k$.
- (ii) $\ker(\phi_i) \cap K_{i-1} = K_i$ for $1 \leq i \leq k$.
- (iii) For each i , $\text{im}(\phi_i)$ is either a soluble group defined by a polycyclic presentation or a direct product of isomorphic nonabelian simple groups.
- (iv) $\text{im}(\phi_k)$ is abelian and $Z_G \leq K_{k-1}$, where Z_G is the scalar subgroup of G .
- (v) $K_k = O_p(G)$, where p is the defining characteristic of G .

In addition, we shall say that the sequence is *correct on scalars* if $Z_G O_p(G) = K_{k-1}$ in (iv).

We denote the sequence (ϕ_1, \dots, ϕ_k) by $[\phi]^{(k)}$. Condition (ii) simply defines the subgroups K_i from the maps ϕ_i , so we can refer to $[\phi]^{(k)}$ as a preliminary sequence for G if the remaining conditions are satisfied. In fact we will almost always have $G = \text{Domain}(\phi_1)$ – the only exception is when we adjoin some scalars to G in ALMOSTSIMPLEMAPS below – but the remaining ϕ_i will frequently be defined on a larger domain than K_{i-1} . (This is a potential source of inefficiency in our overall procedure, because we are only interested in the restriction of ϕ_i to K_{i-1} , but we may on occasion waste effort in analysing the action of ϕ_i on a larger domain.)

Note that Condition (v) says that we insist that $O_p(G)$ occurs at the bottom of the normal series for G . This subgroup can only be nontrivial for reducible groups. Condition (iv) says that the scalar subgroup Z_G of G is also pushed to the bottom of the series, but above $O_p(G)$. We allow the final map ϕ_k to be trivial if $Z_G = 1$.

The first step in the algorithm is to construct the maps $[\phi]^{(k)}$. This is done by means of a recursive algorithm based on Aschbacher decompositions. The maps ϕ_i in the sequence will all be directly computable homomorphisms, as defined in Subsection 1.2. They will either be maps associated with Aschbacher decompositions, or will involve other straightforward computations, such as determinants.

The calculation and identification of the kernels and images, which will be discussed in Subsection 2.3, is only carried out after all of the maps ϕ_i have been defined. Note that some of the layers K_{i-1}/K_i in the series could turn out to be trivial.

The properties (i) – (v) listed will be satisfied by $[\phi]^{(k)}$ provided that all of our procedures work correctly. As we shall see, some of them have a small probability of returning incorrect answers, but these will be detected either when we identify the images and kernels, or when we verify the correctness of the complete series. It is the restriction of ϕ_i to K_{i-1} that will be used to define and work with the quotient K_{i-1}/K_i of the series. Condition (i) ensures that the domain of ϕ_i is large enough to ensure that we can evaluate it on K_{i-1} .

Another point to note is that we often represent insoluble images of the ϕ_i as direct products of quasisimple groups and work modulo scalars. For example, an image that is really isomorphic to $\text{PSL}(n, q)^d$ for some n, q, d would typically be represented as $\text{SL}(n, q)^d$ in its natural representation as a subgroup of $\text{SL}(dn, q)$. In such cases, ϕ_i is represented in the implementation by a map $\tilde{\phi}_i$ with codomain $\text{SL}(dn, q)$, for which $\phi_i(g) = \tilde{\phi}_i(g)Z(\text{SL}(dn, q))$ for $g \in \text{Domain}(\phi_i)$.

In general, a map $\psi : G \rightarrow \text{GL}(n, q)$ for which the induced composite map $\overline{\psi} : G \rightarrow \text{PGL}(n, q)$ is a homomorphism is known as a *projective homomorphism* or a *homomorphism mod scalars*. We define the kernel $\ker(\psi)$ of a projective homomorphism to be the kernel of the induced homomorphism to $\text{PGL}(n, q)$, and the image $\text{im}(\psi)$ to be the complete inverse image in $\text{GL}(n, q)$ of the image in $\text{PGL}(n, q)$ of $\overline{\psi}$.

We fix some general notation regarding arbitrary sequences of homomorphisms. Given two sequences $[\phi]^{(k)}$ and $[\psi]^{(m)}$, we shall denote the concatenation of the two sequences by $[\phi]^{(k)} \text{ cat } [\psi]^{(m)}$. We may also append a single homomorphism ζ to a sequence and denote the result by $[\phi]^{(k)} \text{ cat } \zeta$. We denote the pruned sequence $(\phi_1, \dots, \phi_{k-1})$ by $[\phi]^{(k-1)}$.

Let $\psi : G \rightarrow H$ be an epimorphism for a matrix group H , and let $[\phi]^{(k)}$ be a preliminary sequence for H . Then we define the *pullback* $\text{PULLBACK}([\phi]^{(k)}, \psi)$ of $[\phi]^{(k)}$ through ψ to be the sequence $[\zeta]^{(k)}$ with $\zeta_i = \phi_i \circ \psi$ for each i . Then $\text{PULLBACK}([\phi]^{(k-1)}, \psi)$ represents a normal series for the quotient $G/\ker(\psi) \cong H$ of G .

We shall also use the pullback construction for projective homomorphisms $\psi : G \rightarrow \text{GL}(d, r)$. In that case, let $H = \text{im}(\psi)$ and let $[\phi]^{(k)}$ be a preliminary sequence for H . Then, for $1 \leq i \leq k-1$, we have $Z_H \leq \ker(\phi_i)$, and so the composites $\overline{\phi_i} \circ \psi$ ($1 \leq i \leq k-1$) are (genuine) homomorphisms. We shall denote this sequence of composites by $\text{PULLBACK}([\phi]^{(k-1)}, \psi)$. If, in addition, H acts irreducibly (so $O_p(H) = 1$) and the preliminary sequence $[\phi]^{(k)}$ for H is correct on scalars, then $\text{PULLBACK}([\phi]^{(k-1)}, \psi)$ will represent a normal series for the quotient $G/\ker(\psi)$ of G .

If $\phi_i : G \rightarrow H_i$ are homomorphisms for $1 \leq i \leq k$, then we denote by $\phi_1 \times \dots \times \phi_k$ the homomorphism from G to $H_1 \times \dots \times H_k$ with $g \mapsto (\phi_1(g), \dots, \phi_k(g))$.

Our algorithm to compute a preliminary sequence $[\phi]^{(k)}$ for G is described in Subsection 2.2. In Subsection 2.3, we explain how the images and kernels of these maps are computed using a Monte Carlo algorithm that may underestimate their orders. This series is verified and (if required) corrected by constructing a presentation of G , turning the whole process into a Las Vegas algorithm. The verification

process is described in Subsection 2.4, where we also explain how to test membership of arbitrary elements of $\text{GL}(n, q)$ in the subgroups K_i in the series. So, in particular, this gives us a membership test for $K_0 = G$.

The remainder of the normal series from $O_p(G)$ to 1 is computed by exploiting the module structure of G on its lower triangular blocks (Subsection 2.5). The normal series for G is then refined into a chief series (Subsection 2.6).

2.2. *Constructing the preliminary sequence of maps*

The main procedure to be described in this subsection is `NORMALSERIESMAPS`, which constructs a preliminary sequence for a matrix group G . This is the part of the procedure for finding a chief series that differs most from the composition tree method of Leedham-Green and O'Brien, so we shall treat this topic in more detail than the other parts of the procedure, and provide pseudocode for the analysis of the groups in each of the nine Aschbacher classes.

We shall assume that a version of this algorithm is available for permutation groups. This is dealt with by the BSGS techniques described by Cannon and Holt in [7] and implemented in `MAGMA`. Since a permutation group has no defining characteristic or scalar subgroup, we define the final map ϕ_k in a preliminary sequence for a permutation group to have trivial domain and image, and we do not insist that $\text{im}(\phi_{k-1})$ is abelian.

We shall not write out the pseudocode for `NORMALSERIESMAPS` itself, since it simply uses the methods discussed in Section 1 to find an Aschbacher decomposition, and then calls the appropriate subroutine for that type of decomposition. As we explained in Section 1, there are programs available that test G for membership in each of the Aschbacher classes C1–C8. When they find such a decomposition they return any associated directly computable homomorphisms, such as the action of a matrix on a proper subspace in the reducible case. If the tests fail to find a decomposition in any of these classes, then we assume that the group is in Class C9, in which case it is almost simple modulo scalars and is not equal to a classical group in its natural representation.

Unfortunately, several of these tests, including the tests for classes C2, C4, C5, C6 and C7, may occasionally fail to reach a decision within a reasonable time. In that case, the group is almost always in Class C9 even if it is in one of the other classes as well, so our policy is to abort the test and, provided that none of the other tests returns a positive answer, to assume that the group is in C9. This means that there is a small danger that this assumption is false, and then the group might even be falsely identified as being a specific almost simple group. The error would however be discovered when we come to perform constructive recognition of the (nonabelian) simple composition factors, which will be discussed in Subsection 2.4.

We remark also that, as we shall see later, the correctness of the complete procedure requires us to test for classes C1, C2 and C3 before C4, C5 and C7, and to test for C4 before C6. Errors in which, for example, we test for and find that G lies in C4 after having failed to detect that it also lies in C2 are extremely rare (in fact we have never known them to occur), but they would be detected when we applied `NORMALSERIESMAPS` recursively to the actions on the tensor factors.

We shall now describe each of the procedures for the individual Aschbacher classes that are called by `NORMALSERIESMAPS`. These procedures generally call `NORMALSERIESMAPS` recursively on a permutation group, or on a matrix group of

smaller dimension or over a smaller field than G . In fact all of these recursive calls are to permutation groups or to irreducible matrix groups, and so the final map ϕ_k in the preliminary sequence returned by any of these calls will have trivial kernel. Indeed, the only procedure that can return a final map ϕ_k with nontrivial kernel $O_p(G)$ is REDUCIBLEMAPS.

We claim that, provided that no errors are made in the execution of these procedures (as explained above, certain types of errors may occur with small probability, and these would be detected later), the sequence of homomorphisms returned by each of the procedures is a preliminary sequence for G . Furthermore we claim that the procedures that process Aschbacher decompositions in the classes C4–C9 will return preliminary sequences that are correct on scalars. These properties need to be proved as part of the correctness proofs of the procedures, and it may be assumed by induction that they are true for all recursive calls made within the procedures. Having said that, we shall not in fact write out formal correctness proofs, but we shall of course point out any aspects of them that may not be clear.

Throughout, we refer to the natural $\mathbb{F}_q G$ -module of row vectors by M_G , and the vector space of row vectors by V_G . Given any G -module (or vector space) M we define ACTIONGROUP(M) to be the group generated by the action of the generators of G on M and ACTION(M) to be the homomorphism that maps $g \in G$ to its action on M . We also use PROJECTIVEACTION(M) to denote an induced projective homomorphism; this arises for groups that preserve tensor product and induced tensor product decompositions.

We shall denote the scalar subgroup of a matrix group G by Z_G and we define the function SCALARMAP(n, q) to be the isomorphism from $Z_{\text{GL}(n, q)} \rightarrow \mathbb{F}_q^\times$ with \mathbb{F}_q^\times represented by a polycyclic presentation. In the procedures to calculate a preliminary sequence for groups in the Aschbacher classes C4–C9, we shall define the final map in the sequence to be SCALARMAP(n, q). As we shall see, this will ensure that the returned preliminary sequence is correct on scalars. This is necessary for the correctness of the complete procedure. The preliminary sequences defined for groups in classes C1, C2, and C3 will not necessarily be correct on scalars.

All of the procedures take as input a subgroup G of $\text{GL}(n, q)$ and output a preliminary sequence for G . The input group is assumed to lie in the Aschbacher class for that procedure, and in some cases it is assumed not to lie in various other Aschbacher classes. As we have already mentioned, for all procedures other than REDUCIBLEMAPS, we assume that G is not in Class C1. Other assumptions of this type will be justified as they arise.

REDUCIBLEMAPS(G)

Assume: G lies in Class C1.

- 1 Construct (using the Meataxe) a composition series
 $0 = E_m < \dots < E_1 < E_0 = M_G$ for M_G ;
- 2 **for** $i \in [1..m]$
- 3 **do** $A_i := \text{ACTIONGROUP}(E_{i-1}/E_i)$; $\psi_i := \text{ACTION}(E_{i-1}/E_i)$;
- 4 $[\phi^{(i)}]^{(k_i)} := \text{NORMALSERIESMAPS}(A_i)$;
- 5 $[\zeta^{(i)}]^{(k_i)} := \text{PULLBACK}([\phi^{(i)}]^{(k_i)}, \psi_i)$;
- 6 **return** $[\zeta^{(1)}]^{(k_1-1)} \text{ cat } \dots \text{ cat } [\zeta^{(m)}]^{(k_m-1)} \text{ cat } (\zeta_{k_1}^{(1)} \times \dots \times \zeta_{k_m}^{(m)})$;

Notice that we push the scalars coming from each of the irreducible actions of G on E_{i-1}/E_i to the bottom of the preliminary sequence returned for G . So the image of the final map in the sequence returned will be abelian, and its domain will contain Z_G , possibly as a proper subgroup.

Our policy of analysing, via the recursive call to `NORMALSERIESMAPS`, each of the action groups of E_{i-1}/E_i separately appears to be necessary in order to obtain a normal series for G , but in some examples it can result in our repeating the analyses of the same chief factors of G . This will happen, roughly speaking, when the action of G on the E_{i-1}/E_i is diagonal rather than a full direct product. We could, in principle, avoid such repetition if some of the E_{i-1}/E_i were isomorphic as G -modules, but we have not yet attempted this in the implementation.

We turn now to groups in Class C2. Notice at Line 7 in the procedure below, we are applying `NORMALSERIESMAPS` recursively only to the action of the stabilizer of the first block in the decomposition on that block. The corresponding actions on the other blocks are computed by conjugation in Line 12. This is a potentially significant gain in efficiency compared with the composition tree program of Leedham-Green and O'Brien [23], in which the kernels of the actions on all of the blocks are analysed separately.

The maps ζ_i in Line 12 will not necessarily be surjective even when ψ_i is surjective, but the images will be subdirect products of the images of the ψ_i and so, in the insoluble case, will still be direct products of isomorphic simple groups. (See Proposition 2.9 and the associated discussion in Subsection 2.3 below.)

`IMPRIMITIVEMAPS(G)`

Assume: G lies in Class C2 but not in Class C1.

- 1 Construct (using methods described in [19]) a set of blocks of imprimitivity $\Omega = \{V_1, \dots, V_r\}$ for G , along with a homomorphism $\rho : G \rightarrow \text{Sym}(\Omega)$;
- 2 $\Sigma := \text{im}(\rho)$;
- 3 $[\tau]^{(k)} := \text{PULLBACK}(\text{NORMALSERIESMAPS}(\Sigma), \rho)$;
- 4 $G_1 := G_{V_1}$ the stabiliser of V_1 in G ;
- 5 Let M_1 be V_1 considered as an $\mathbb{F}_q G_1$ -module;
- 6 $A := \text{ACTIONGROUP}(M_1)$; $\theta := \text{ACTION}(M_1)$;
(* Note that $\text{Domain}(\theta) = G_1 *$)
- 7 $[\psi]^{(m)} := \text{NORMALSERIESMAPS}(A)$;
- 8 **for** $i \in [1..r]$
- 9 **do** Choose $e_i \in G$ with $V_i^{\rho(e_i)} = V_1$;
- 10 Let $\alpha_{e_i} : G \rightarrow G$ be the map $\alpha_{e_i}(g) = e_i^{-1} g e_i$;
 (* So $\alpha_{e_i}(G_{V_i}) = G_1$ and $\text{Domain}(\theta \circ \alpha_{e_i}) = G_{V_i} *$)
- 11 **for** $i \in [1..m]$
- 12 **do** $\zeta_i := \psi_i \circ \theta \circ \alpha_{e_1} \times \dots \times \psi_i \circ \theta \circ \alpha_{e_r}$;
- 13 **return** $[\tau]^{(k)}$ **cat** $[\zeta]^{(m)}$;

Groups in Class C3 will be subdivided into those that act absolutely irreducibly and those that do not. Absolutely irreducible subgroups of $\text{GL}(n, q)$ in Class C3 that are isomorphic to subgroups of $\Gamma\text{L}(n/e, q^e)$ containing a field automorphism of order e become imprimitive with e blocks of imprimitivity when regarded as

subgroups of $\mathrm{GL}(n, q^e)$, so our method in that case is to extend the ground field of G and call `IMPRIMITIVEMAPS`. Groups that are not absolutely irreducible are handled by the following procedure.

`ABSOLUTELYREDUCIBLEMAPS(G)`

Assume: G acts irreducibly but not absolutely irreducibly on M_G .

- 1 Construct (using the methods outlined in [20]) an isomorphism $\psi : G \rightarrow H \leq \mathrm{GL}(\frac{n}{e}, q^e)$ where H is the irreducible reduced-degree representation of G ;
- 2 $[\phi]^{(k)} := \text{NORMALSERIESMAPS}(H)$;
- 3 **return** `PULLBACK`($[\phi]^{(k)}, \psi$);

For groups in Class C4, which preserve a tensor product decomposition $V_G = U \otimes V$, the induced actions of G on U and V are only projective actions. In order to use the pullback construction, we therefore require that the preliminary sequence computed on the images of these actions are correct on scalars. Recall that we are claiming that the procedures for processing groups in Classes C4–C9 return preliminary sequences that are correct on scalars, and that we are assuming by induction that this is true for recursive calls of these procedures. So we need to assume that the induced actions of G on U and V do not lie in any of the classes C1, C2, C3. To justify this, we shall now show that if either of the actions were in C1, C2 or C3 then so would G be, in which case we would have already called one of the earlier procedures on G .

THEOREM 2.2. *Let G preserve a tensor decomposition of V_G , as $V_G = U \otimes W$.*

1. *If G acts reducibly on U then G acts reducibly on V_G .*
2. *If G acts imprimitively on U then G acts imprimitively on V_G .*
3. *If the action of G on U is not absolutely irreducible then G is not absolutely irreducible on V_G .*
4. *If the action of G on U is semilinear then G is semilinear on V_G .*

Proof. The proofs of 1 and 2 are straightforward. For 3, assume that the action of G on U is not absolutely irreducible. Then there exists a non-scalar $C \in \mathrm{GL}(U)$ that centralises the action of g on U for all $g \in G$. Now let D be a matrix in $\mathrm{GL}(n, q)$ that preserves the decomposition $U \otimes W$ of V_G and acts as C on U and as I_W on W . Then D is a non-scalar matrix that centralises G .

For 4, we know from 3 that G has a normal irreducible subgroup N that does not act absolutely irreducibly. Let D be the non-scalar matrix that centralises N and acts as I_W on W as constructed above. Let $g \in G$, let g_1 be the induced action of g on U and let C be the induced action of D on U . Then since G is semilinear on U there exists $i = i(g)$ such that $Cg_1 = g_1C^{q^i}$, but the action of D on W is trivial so $Dg = gD^{q^i}$ and G is semilinear. \square

TENSORMAPS(G)

Assume: G lies in Class C4 but not in Classes C1, C2, C3.

(* We use the algorithm of [24] to find U and W with $V_G = U \otimes W$ *)

- 1 $\psi_1 := \text{PROJECTIVEACTION}(U)$; $\psi_2 := \text{PROJECTIVEACTION}(W)$;
- 2 $H_1 := \text{im}(\psi_1)$; $H_2 := \text{im}(\psi_2)$;
- 3 $[\alpha]^{(k_1)} := \text{NORMALSERIESMAPS}(H_1)$; $[\beta]^{(k_2)} := \text{NORMALSERIESMAPS}(H_2)$;
 (* By Theorem 2.2 and induction, $[\alpha]^{(k_1)}$ and $[\beta]^{(k_2)}$ are correct on scalars *)
- 4 $[\phi]^{(k_1+k_2-1)} := \text{PULLBACK}([\alpha]^{(k_1-1)}, \psi_1)$ **cat** $\text{PULLBACK}([\beta]^{(k_2-1)}, \psi_2)$
cat $\text{SCALARMAP}(n, q)$;
- 5 **return** $[\phi]^{(k_1+k_2-1)}$;

Groups in Class C5 are conjugates of subgroups of $\langle \text{GL}(n, \mathbb{K}), Z \rangle$, where Z is the scalar subgroup of $\text{GL}(n, q)$ and \mathbb{K} is a proper subfield of \mathbb{F}_q . They therefore give rise to a projective homomorphism $\psi : G \rightarrow H \leq \text{GL}(n, \mathbb{K})$ with kernel Z_G .

It is easy to see that if the image H of ψ is reducible or imprimitive then so is G . Furthermore, we leave it to the reader to prove that if H is reducible but not absolutely irreducible, then G is either reducible or is not absolutely irreducible, and if H is absolutely irreducible but semilinear, then G is either semilinear or imprimitive. So we may assume that H is not in Class C1, C2 or C3, and hence that the preliminary sequence computed for H is correct on scalars.

SMALLERFIELDMAPS(G)

Assume: G lies in Class C5 but not in Classes C1, C2, C3.

- 1 Construct (using the methods given in [14]) a projective homomorphism $\psi : G \rightarrow H \leq \text{GL}(n, \mathbb{K})$ for a proper subfield \mathbb{K} of \mathbb{F}_q ;
- 2 $[\phi]^{(k)} := \text{NORMALSERIESMAPS}(H)$;
 (* $[\phi]^{(k)}$ is correct on scalars *)
- 3 $[\alpha]^{(k-1)} := \text{PULLBACK}([\phi]^{(k-1)}, \psi)$;
- 4 **return** $[\alpha]^{(k-1)}$ **cat** $\text{SCALARMAP}(n, q)$;

Groups G in Class C6 have a normal subgroup E that is either extraspecial of order r^{2m+1} or a 2-group of symplectic type of order 2^{2m+2} for a prime r and integer $m \geq 1$, where $|Z(E)| = r$ or 4, respectively. (A 2-group of symplectic type is a central product of an extraspecial 2-group with a cyclic group of order 4.) So we have an action of G/E on $E/Z(E)$ giving rise to an embedding $\psi : G/E \rightarrow \text{GL}(2m, r)$. We want to apply NORMALSERIESMAPS recursively to $\text{im}(\psi)$ and then to pull the resulting preliminary series back through ψ and, for this to work properly, we need to be able to assume that $\text{im}(\psi)$ is irreducible, thereby ensuring that $O_r(\text{im}(\psi)) = 1$. In the next few lemmas we show that, if $\text{im}(\psi)$ is reducible, then G lies in one of the classes C2, C3, C4, which we can assume is not the case. We shall prove the lemmas only for the extraspecial case.

LEMMA 2.3. *Let r be a prime and let $E \leq \text{GL}(r^m, q)$ be an extraspecial r -group of order r^{2m+1} , with $r \nmid q$. Then E is the central product of m extraspecial groups of order r^3 . Furthermore, E preserves a tensor decomposition of M_E as $M_E = M_1 \otimes \cdots \otimes M_m$, where the restriction of E to each M_i is a representation of an extraspecial group of order r^3 .*

Proof. See [15, Theorem 5.5.5]. □

LEMMA 2.4. *Let $E \leq \text{GL}(2^m, q)$ be a 2-group of symplectic type of order 2^{2m+2} , with $q \equiv 1 \pmod{4}$. Then E preserves a tensor decomposition of M_E as $M_E = M_1 \otimes \cdots \otimes M_m$.*

LEMMA 2.5. *Let r be a prime and let $E \leq \text{GL}(r^m, q)$ be an extraspecial r -group of order r^{2m+1} or a 2-group of symplectic type of order 2^{2m+2} . Let H be a proper subgroup of E that contains $Z(E)$. Then H acts reducibly.*

Proof. It is enough to show H acts reducibly when $|G : H| = r$. Assume first that $m = 1$, so $|E| = r^3$. Then H has order r^2 , so is abelian. By [15, Theorem 3.2.3], H has no faithful irreducible representations, so H must act reducibly.

In general E is a central product of m extraspecial groups of order r^3 . As H has index r in E , it can be seen that H is a central product of $r - 1$ extraspecial groups of order r^3 and an abelian group of order r^2 . Hence one of the tensor factors of the representation of H in $\text{GL}(r^m, q)$ is reducible and H acts reducibly by Theorem 2.2. □

THEOREM 2.6. *Let $G \leq \text{GL}(r^m, q)$ act irreducibly and have a normal subgroup E that is either an extraspecial r -group or, when $r = 2$, a 2-group of symplectic type. Let L be the $\mathbb{F}_r G$ -module defined by the conjugation action of G on $E/Z(E)$. If G acts reducibly on L then either G acts imprimitively on V_G , or G acts as a group of semilinear automorphisms on V_G , or G preserves a tensor decomposition of V_G .*

Proof. This follows from the general theory of Smash described in [18]. Briefly, Clifford's Theorem [15, Theorem 3.4.1] states that, if $N \trianglelefteq G$, then V_G splits as a direct sum $W_1 \oplus \cdots \oplus W_k$ of irreducible $\mathbb{F}_q N$ -modules, all of the same dimension. For some $t, s \geq 1$, with $ts = k$, the W_i 's partition into t sets containing s pairwise isomorphic $\mathbb{F}_q N$ -modules each, and if V_1, \dots, V_t are each the sum of pairwise isomorphic W_i , so that $V = V_1 \oplus \cdots \oplus V_t$, then G permutes the V_i transitively.

If G acts reducibly on L then G normalises some proper subgroup, N say, of E . By the previous lemmas, N acts reducibly on V_G so, in the context of Clifford's Theorem, we must have $k > 1$.

If $t > 1$, then $\{V_1, \dots, V_t\}$ forms a block system for G and G is imprimitive. Otherwise V_G decomposes as a direct sum of k irreducible pairwise isomorphic $\mathbb{F}_q N$ -modules, $W = W_1, W_2, \dots, W_k$. From the descriptions given in [18] we see that if each W_i acts absolutely irreducibly then G preserves a tensor decomposition of V_G as $U \otimes W$, where N acts as scalars on U . If each W_i does not act absolutely irreducibly then G is semilinear. □

Let $G \leq \text{GL}(r^m, q)$ act irreducibly, and suppose that G has an extraspecial normal r -subgroup or a normal 2-subgroup of symplectic type E . Then methods described in [18] provide us with the action of G on the $\mathbb{F}_r G$ -module $E/Z(E)$, where $Z(E)$ is a group of scalar matrices of order r , or of order 4 if E is a 2-group of symplectic type. The kernel of this action is EZ_G . We also obtain elements $e_1, \dots, e_{2m} \in E$ whose images in $E/Z(E)$ are a free basis for the elementary abelian group. We wish to construct a homomorphism $\alpha : EZ_G \rightarrow E/Z(E)$ that has kernel Z_G .

THEOREM 2.7. *Let E and e_1, \dots, e_{2m} be as defined above and let $\bar{e}_1, \dots, \bar{e}_{2m}$ be the images of the e_i in $E/Z(E)$. Let z be a scalar element of G of order r . Define $\alpha : EZ_G \rightarrow E/Z(E)$ by $\alpha(g) = \bar{e}_1^{i_1} \cdots \bar{e}_{2m}^{i_{2m}}$, where $[g, e_j] = z^{i_j}$ for $1 \leq j \leq 2m$. Then α is a homomorphism with kernel Z_G .*

Proof. If $j \in \{1, \dots, 2m\}$ and $g \in EZ_G$, then $[g, e_j] \in Z_E$, since e_i and g commute modulo Z_G and $[gy, e_j] = [g, e_j]$ for any scalar y . Also, $[gh, e_j] = [g, e_j]^h [h, e_j] = [g, e_j][h, e_j]$ for any $g, h \in EZ_G$ and $j \in \{1, \dots, 2m\}$, so α is a homomorphism. Finally, $[g, e_j] = 1$ for all $1 \leq j \leq 2m$ if and only if $g \in Z(EZ_G) = Z_G$, so $\ker(\alpha) = Z_G$ and α is surjective. \square

EXTRASPECIALNORMALISERMAPS(G)

Assume: G lies in Class C6 but not in Classes C1, C2, C3, C4.

(* We find E and the module $L := E/Z(E)$ as described in [18] *)

- 1 $\psi := \text{ACTION}(L)$; $H := \text{ACTIONGROUP}(L)$;
- 2 $[\phi]^{(k)} := \text{NORMALSERIESMAPS}(H)$;
(* $\ker(\phi_k) = O_r(G/EZ_G) = 1$ by Theorem 2.6 *)
- 3 Let $\alpha : EZ_G \rightarrow E/Z(E)$ be the homomorphism defined in Theorem 2.7.
- 4 **return** $\text{PULLBACK}([\phi]^{(k)}, \psi)$ **cat** α **cat** $\text{SCALARMAP}(n, q)$;

We require results for the tensor induced case similar to those in the tensor case.

THEOREM 2.8. *Let G be a tensor induced group. So $V_G = V_1 \otimes \cdots \otimes V_m$. For $1 \leq i \leq m$ let G_i be the subgroup of G that stabilises V_i and let A_i be the restriction of G_i to V_i . Note that the A_i are all isomorphic.*

1. If A_1 acts reducibly on V_1 then G acts reducibly on V_G .
2. If A_1 acts imprimitively on V_1 then G acts imprimitively on V_G .
3. If the action of A_1 on V_1 is not absolutely irreducible then G acts reducibly on V_G .
4. If the action of A_1 on V_1 is semilinear then G acts imprimitively on V_G .

Proof. The proofs of 1 and 2 are straightforward. For 3, if \mathbb{F}_{q^e} is the splitting field for V_1 under A_1 , then \mathbb{F}_{q^e} is the splitting field for V_i under A_i for $1 \leq i \leq m$. Over \mathbb{F}_{q^e} we have $V_i = L_i \oplus L_i^\sigma \oplus \cdots \oplus L_i^{\sigma^{e-1}}$, where L_i is absolutely irreducible and σ is the field automorphism. Then the subspace

$$(L_1 \otimes \cdots \otimes L_m) \oplus (L_1^\sigma \otimes \cdots \otimes L_m^\sigma) \oplus \cdots \oplus (L_1^{\sigma^{e-1}} \otimes \cdots \otimes L_m^{\sigma^{e-1}})$$

is fixed by the field automorphism σ . Hence it can be written over \mathbb{F}_q . It is also fixed by all permutations of the tensor factors, hence by elements of G . So G is reducible. The proof of 4 is similar and is left to the reader. \square

The code for `TENSORINDUCEDMAPS` is similar to that for `IMPRIMITIVEMAPS`.

TENSORINDUCEDMAPS(G)

Assume: G lies in Class C7, but not in Classes C1, C2, C3, C4.

- 1 Construct (using methods described in [25]) a set of tensor factors $\Omega = \{V_1, \dots, V_r\}$ permuted by G , and a homomorphism $\rho : G \rightarrow \text{Sym}(r)$;
- 2 $\Sigma := \text{im}(\rho)$;
- 3 $[\tau]^{(k)} := \text{PULLBACK}(\text{NORMALSERIESMAPS}(\Sigma), \rho)$;
- 4 $G_1 := G_{V_1}$, the stabiliser of V_1 in G ;
- 5 Let M_1 be V_1 considered as an $\mathbb{F}_q G_1$ -module;
- 6 $\theta := \text{PROJECTIVEACTION}(V_1)$; $A := \text{im}(\theta)$;
- 7 $[\psi]^{(m)} = \text{NORMALSERIESMAPS}(A)$;
 (* $[\psi]^{(m)}$ is correct on scalars by Theorem 2.8 and induction *)
- 8 **for** $i \in [1..r]$
- 9 **do** Choose $e_i \in G$ with $V_i^{\rho(e_i)} = V_1$;
- 10 Let $\alpha_{e_i} : G \rightarrow G$ be the map $\alpha_{e_i}(g) = e_i^{-1} g e_i$;
- 11 **for** $i \in [1..m-1]$
- 12 **do** $\zeta_i := \psi_i \circ \theta \circ \alpha_{e_1} \times \dots \times \psi_i \circ \theta \circ \alpha_{e_r}$;
 (* $Z_A = \ker(\psi_{m-1})$ and the ζ_i are normal homomorphisms *)
- 13 **return** $[\tau]^{(k)} \text{ cat } [\zeta]^{(m-1)} \text{ cat } \text{SCALARMAP}(n, q)$;

If G is in Class C8 then the algorithms of [28] will return the type of classical group normalised by G and, if G does not contain $\text{SL}(n, q)$, they will also return a classical form Φ that is fixed by G modulo scalars. We denote the action of $g \in G$ on the form Φ by Φ^g . So, for $g \in G$, $\Phi^g = w\Phi$ for some non-zero scalar $w \in \mathbb{F}_q$.

The code for handling these groups is complicated by the fact that the structure of the normaliser Δ of the quasisimple classical group Ω in $\text{GL}(n, q)$ varies from case to case, and also by our need to push the scalar subgroup Z_G of G to the bottom of the normal series. We therefore present separate code for the four cases.

Here is a brief summary of the structure of Δ ; we refer the reader to [22, Chapter 2] for further details. There are subgroups S and I of Δ with $\Omega \leq S \leq I \leq \Delta$. In the linear case, $I = \Delta = \text{GL}(n, q)$. Otherwise, Δ (sometimes known as the *conformal* unitary, symplectic or orthogonal group) is the subgroup of $\text{GL}(n, q)$ that fixes the form modulo scalars, and I is the subgroup that fixes the form; that is, $\Phi^g = \Phi$ for $g \in I$. In the unitary groups and in all groups in even characteristic or odd degree, Δ is generated by I and scalar matrices, but in the symplectic and orthogonal groups in odd characteristic and even degree, the subgroup generated by I and the scalars has index two in Δ . In all cases, S is the ‘special’ subgroup of matrices of determinant 1. In the symplectic case $S = I$, and in the orthogonal groups $|I : S| = 1$ or 2 , respectively, in even and odd characteristics. The quasisimple group Ω is the derived subgroup of S and has index 2 in S in the orthogonal groups, and is equal to S otherwise.

Membership of elements g of Δ in I and in S is readily tested by computing Φ^g and $\det(g)$. Membership of elements g of $S = \text{SO}^\epsilon(n, q)$ in $\Omega = \Omega^\epsilon(n, q)$ can also be easily computed using the function `SPINORNORM`; see [33, p. 163] for a description. The value of `SPINORNORM`(g) is 0 when $g \in \Omega$ and 1 otherwise. It turns out that, if n is even, q is odd and $\det(\Phi)$ is a non-square, then $-I_n$ has spinor norm 1, and so S and Ω are equal mod scalars, and we take the map ϕ_3 in `C8MAPSORTH`(G) to be trivial. In all other cases, $-I_n$ has spinor norm 0 and $|\text{PSO}^\epsilon(n, q) : \text{P}\Omega^\epsilon(n, q)| = 2$. Note that ϕ_3 is well-defined in this situation, because the scalar z_g is unique up to multiplication by $-I_n$.

Chief series and the soluble radical of a matrix group over a finite field

Let $\gamma : \mathbb{F}_q^\times \rightarrow \mathbb{F}_q^\times / N$ with $N := \langle \det(g) \mid g \in Z_G \rangle$ be the natural epimorphism with the image represented as a group with polycyclic presentation. We denote by \det_Z the homomorphism $\gamma \circ \det : \mathrm{GL}(n, q) \rightarrow \mathbb{F}_q^\times / N$. Note that, with I and S defined as above, we have $\ker(\det_Z) \cap I = SZ_G$. Note also that the scalar z_g in the procedures below, which is chosen to make $gz_g^{-1} \in S$ in all cases, can be easily computed from $\det(g)$ and Φ^g .

C8MAPSSL(G)

Assume: $\mathrm{SL}(n, q) \leq G \leq \mathrm{GL}(n, q)$

- 1 $\phi_1 := \det_Z$;
- 2 Define $\phi_2 : \langle \mathrm{SL}(n, q), Z_G \rangle \rightarrow \mathrm{PSL}(n, q)$ from the projective homomorphism $g \mapsto gz_g^{-1}$, where z_g is a scalar with $\det(z_g) = \det(g)$;
- 3 **return** $(\phi_1, \phi_2, \mathrm{SCALARMAP}(n, q))$;

C8MAPSUNITARY(G)

Assume: G contains and normalises $\mathrm{SU}(n, \sqrt{q})$.

(* Let Φ be the unitary form associated to G *)

- 1 $\phi_1 := \det_Z$;
- 2 Define $\phi_2 : \langle \mathrm{SU}(n, q), Z_G \rangle \rightarrow \mathrm{PSU}(n, \sqrt{q})$ from the projective homomorphism $g \mapsto gz_g^{-1}$, where z_g is a scalar with $\det(z_g) = \det(g)$ and $\Phi^g = z_g^{\sqrt{q}+1}\Phi$;
- 3 **return** $(\phi_1, \phi_2, \mathrm{SCALARMAP}(n, q))$;

C8MAPSP(G)

Assume: G contains and normalises $\mathrm{Sp}(n, q)$.

(* Let Φ be the symplectic form associated to G *)

- 1 Define $\phi_1 : G \rightarrow \mathbb{Z}_2 = \{0, 1\}$ as follows:
- 2 Let $\omega \in \mathbb{F}_q$ be such that $\Phi^g = \omega\Phi$;
- 3 **if** ω is a square **then** $\phi_1(g) := 0$ **else** $\phi_1(g) := 1$;
- (* ϕ_1 is always trivial when q is even *)
- 4 Define $\phi_2 : \langle \mathrm{Sp}(n, q), Z_G \rangle \rightarrow \mathrm{PSP}(n, q)$ from the projective homomorphism $g \mapsto gz_g^{-1}$, where z_g is a scalar with $\Phi^g = z_g^2\Phi$;
- 5 **return** $(\phi_1, \phi_2, \mathrm{SCALARMAP}(n, q))$;

C8MAPSORTH(G)

Assume: G contains and normalises $\Omega^\epsilon(n, q)$.

(* Let Φ be the orthogonal or quadratic form associated to G *)

- 1 Define $\phi_1 : G \rightarrow \mathbb{Z}_2 = \{0, 1\}$ as follows:
- 2 Let $\omega \in \mathbb{F}_q$ be such that $\Phi^g = \omega\Phi$;
- 3 **if** ω is a square **then** $\phi_1(g) := 0$ **else** $\phi_1(g) := 1$;
- (* ϕ_1 is always trivial when q is even or n is odd *)
- 4 $\phi_2 := \det_Z$;
- 5 Define $\phi_3 : \langle \mathrm{SO}^\epsilon(n, q), Z_G \rangle \rightarrow \mathbb{Z}_2 = \{0, 1\}$ as follows:
- 6 **if** n is even, q is odd, and $\det(\Phi)$ is a non-square
- 7 **then** $\phi_3(g) = 0$;
- 8 **else** $\phi_3(g) := \mathrm{SPINORNORM}(gz_g^{-1})$
where z_g is a scalar with $\det(z_g) = \det(g)$ and $\Phi^g = z_g^2\Phi$;
- 9 Define $\phi_4 : \langle \Omega^\epsilon(n, q), Z_G \rangle \rightarrow \mathrm{P}\Omega^\epsilon(n, q)$ from the projective homomorphism $g \mapsto gz_g^{-1}$, where z_g is a scalar with $\det(z_g) = \det(g)$ and $\Phi^g = z_g^2\Phi$;
- 10 **return** $(\phi_1, \phi_2, \phi_3, \phi_4, \mathrm{SCALARMAP}(n, q))$;

To deal with groups in Class C9, we make use of the following procedure, developed by Leedham-Green and O'Brien in [25], to perform membership testing in a normal subgroup of a group.

ELEMENTINN(G, N, g)

Input: Group G , normal subgroup N of G , element $g \in G$.

```

1   $m := |g|$ ;
2  for some number  $I$  of times
3      do  $a := \text{RANDOM}(N)$ ;
4           $m := \text{GCD}(m, |ag|)$ ;
5          if  $m = 1$ 
6              then return true;
7  return false;
```

It is easy to see that if ELEMENTINN returns true then $g \in N$, but it could conceivably return a false negative; that is, it could claim that an element $g \in N$ is not in N . We shall only be applying it when N is quasisimple, in which case an unpublished result of Babai, Pálffy and Saxl enables us to prove an upper bound on the probability of a false negative as a function of the chosen number I of iterations of the loop of the procedure. In other words, we can make ELEMENTINN into a Monte-Carlo algorithm in this situation. We omit the details, which can be found in [4].

Using ELEMENTINN, we can test whether a group G is perfect. The commutator subgroup of G is equal to the normal closure of the subgroup H generated by the commutators of the generators of G . Using a variant of the product replacement algorithm we can compute random elements of the normal closure of a subgroup H of G . Therefore, using a variant of ELEMENTINN where RANDOM(N) in Line 3 is replaced by a function computing a random element of the normal closure of N , we can determine if each generator of G lies in the normal closure of H . Once again, this method can return false negatives but not false positives and, if the derived group of G is quasisimple, then we can estimate upper bounds for the probability of a false negative.

This in turn enables us to define a procedure STABLEDERIVATIVE, which will compute the last member $G^{(\infty)}$ of the derived series of a group G and, if $G^{(\infty)}$ is quasisimple, then the probability of an incorrect result can be estimated and made as low as we wish.

Finally, given a normal subgroup N of G for which the index $|G : N|$ is small, we can use ELEMENTINN to enumerate the cosets of N in G , and then construct the permutation representation COSETIMAGE(G, N) of G on the right cosets of N in G .

We are now able to present the algorithm ALMOSTSIMPLEMAPS. Notice that we apply COSETIMAGE(G, N) only when N is quasisimple, and we apply STABLEDERIVATIVE(G) only when $G^{(\infty)}$ is quasisimple, so we can choose parameters to make the probability of a false result as small as we please. Note also that in the application of COSETIMAGE(G, N), G/N is contained in the outer automorphism group of a finite simple group, and so will be moderately small.

If we did make an error and underestimate the group N at this stage, then the error would be detected when we carried out constructive recognition of the simple group, since we would then be able to carry out membership testing in N

deterministically, and we could verify that our coset representatives of N in G were all genuinely in distinct cosets.

ALMOSTSIMPLEMAPS(G)

Assume: G is absolutely irreducible and almost simple modulo scalars.

- 1 $\hat{G} := \langle G, z \rangle$ where z is a generator of $Z_{\text{GL}(n,q)}$;
 (* We actually compute a preliminary sequence for \hat{G} *)
- 2 $\hat{N} := \text{STABLEDERIVATIVE}(\hat{G})$;
- 3 $\hat{N} := \langle \hat{N}, z \rangle$;
- 4 $\phi_1 := \text{COSETIMAGE}(\hat{G}, \hat{N})$;
- 5 Let $\phi_2 : \hat{N} \rightarrow \hat{N}/Z(\hat{N})$ be defined by the projective homomorphism $\hat{N} \rightarrow \hat{N}$ which maps $g \mapsto zg$, where z is a scalar with $\text{ELEMENTIN}(\hat{N}, N, zg) = \text{true}$;
- 6 **return** $(\phi_1, \phi_2, \text{SCALARMAP}(n, q))$;

During the computations that we carry out when applying NORMALSERIESMAPS to a group $G \leq \text{GL}(n, q)$, we can compute a multiset, $M = \{S_1^{e_1}, \dots, S_m^{e_m}, \mathbb{Z}_{p_1}^{t_1}, \dots, \mathbb{Z}_{p_r}^{t_r}\}$, where each S_i is a nonabelian simple group, and the multiset of composition factors of $G/O_p(G)$ is a subset of M . We only require M to contain the names of the simple groups S_i , and not any representations of the groups themselves. We shall use this multiset M in Subsection 2.3 to obtain a probability estimate of the number of generators required to generate the subgroups in a chief series of G .

We may compute M as follows. We assume that there is no difficulty in identifying the composition factors that arise from permutation groups. The only algorithms above that do not use recursion are ALMOSTSIMPLEMAPS and the four C8MAPS procedures. At each step of these algorithms we store either the factorisation of the order of the soluble image or the standard name of the quasisimple image, which (in the case of ALMOSTSIMPLEMAPS), we can find using non-constructive recognition. Hence, by recursion, with all maps ϕ_i constructed by NORMALSERIESMAPS we may store either a factorization of $\text{im}(\phi_i)$ if it is soluble, or the standard name of one the isomorphic simple groups in $\text{im}(\phi_i)$ along with an upper bound for the number of copies. Observe also that this provides us with a test for the solubility of a matrix group.

2.3. Evaluating the Images and Kernels

We assume now that we have successfully computed a preliminary sequence $[\phi]^{(k)}$ for $G \leq \text{GL}(n, q)$. At this stage we know only a superset of the composition factors of G . The next step is to compute the images and kernels of the ϕ_i , which we do as follows.

Let $G_0 = G$. For $i = 1, \dots, k$, do the following.

1. Compute generators of the image Q_i of the restriction of ϕ_i to G_{i-1} by applying ϕ_i to a set of generators of G_{i-1} . (For $i = 0$ we just use the given generating set of G .)
2. If Q_i is insoluble then identify Q_i constructively (see Subsection 1.1).
3. If $i < k$ then compute generators of a subgroup G_i of $\ker(\phi_i) \cap G_{i-1}$, for which we hope that $G_i O_p(G) = (\ker(\phi_i) \cap G_{i-1}) O_p(G)$.

Since the ϕ_i are directly computable homomorphisms, Step 1 is straightforward, and we shall discuss Steps 2 and 3 shortly. In the next subsection, we shall describe the process in which we verify the correctness of $G_i O_p(G) = (\ker(\phi_i) \cap G_{i-1}) O_p(G)$ for

$0 < i < k$. After this verification we know that $G_i O_p(G) = K_i$ for $0 \leq i < k$, where K_i are the subgroups associated with the preliminary sequence (Definition 2.1). Note that, from the construction of $[\phi]^{(k)}$, we know that $\ker(\phi_k) = O_p(G)$, but we do not attempt to find generators of $O_p(G)$ at this stage. We shall do that later, in Subsection 2.5.

We turn now to the constructive recognition of the insoluble images Q_i in Step 2. Assume that $\text{im}(\phi_i)$ is a direct product S^l of isomorphic nonabelian simple groups. Since (assuming that the kernels calculated so far are correct modulo $O_p(G)$) $Q_i = \phi_i(G_{i-1})$ is a normal subgroup of $\text{im}(\phi_i)$, Q_i is a direct product S^m of some possibly smaller number m of copies of S (Q_i could even be trivial). Furthermore, in some cases, particularly when ϕ_i arises from an application of IMPRIMITIVE MAPS or TENSORINDUCED MAPS, $\text{im}(\phi_i)$ is defined as a subdirect product of a possibly larger direct product of copies of S . So we must first determine l and then m .

Subdirect products of direct products of isomorphic nonabelian simple groups are described by the following result, of which we leave the proof to the reader.

PROPOSITION 2.9. *Let $G = S_1 \times S_2 \times \cdots \times S_n$, where the S_i are all isomorphic to the same nonabelian simple group S , and let H be a subdirect product of G . Then there is a partition \mathcal{P}_i ($1 \leq i \leq l$) of the set $\{1, 2, \dots, n\}$ such that $H \cong S^l$, and the intersection of H with each of the groups $G_i := \times_{j \in \mathcal{P}_i} S_j$ for $1 \leq i \leq l$ is a diagonal subgroup of G_i isomorphic to S .*

In practice, provided that we can compute the projections of G onto the S_i and orders of elements in G , it is easy to calculate the partition \mathcal{P}_i with an arbitrarily small probability of error. We choose random elements $g = (g_1, \dots, g_n)$ of G and compute the orders of the components g_i . Then $|g_j| = |g_k|$ whenever j and k are in the same set \mathcal{P}_i but, with probability greater than $1/2$, $|g_j| \neq |g_k|$ when j and k are in different sets \mathcal{P}_i .

So we can identify $\text{im}(\phi_i)$ and then $\phi_i(G_{i-1})$, which is a normal subgroup of $\text{im}(\phi_i)$, and consists of a direct product of some of the direct factors of $\text{im}(\phi_i)$, which are again easily identified. So we can assume that we have a representation of Q_i as a direct product S^m of a known number m of copies of S .

Now, again by choosing random elements g of Q_i , computing the orders of the components of g , and taking suitable powers of g , it is not difficult to find elements in the simple direct factors S_i of Q_i , and hence to find generators of the S_i . We then carry out constructive recognition of the S_i as discussed in Subsection 1.1. By checking that the relations of the presentation of the corresponding finite simple group are satisfied in S_i , we prove that S_i is genuinely isomorphic to this simple group. So, if we had failed to find an Aschbacher decomposition for S_i and wrongly concluded that S_i was simple, then we would detect the error at this point. We would also find out at this stage if we had underestimated the number m of copies of S , although that particular error is very unlikely indeed.

The constructive recognition of the S_i enables us to solve the rewriting problem in S_i , and it is straightforward to glue these solutions together to solve the rewriting problem in Q_i . This process can be made more efficient by using the fact that the S_i are permuted under the action of G induced by conjugation in G and the application of the map ϕ_i . So, once we have solved the rewriting problem in one S_i , we can solve it immediately in any S_j in the same orbit under this action of G . This refinement results in a very significant improvement in performance. We refer the reader to Section 2.3 of the second author's PhD thesis [31] for further details.

Now we turn to the construction of a generating set of $\ker(\phi_i) \cap G_{i-1}$ modulo $O_p(G)$ in Step 3 of the process summarised earlier. We may construct individual random elements of $\ker(\phi_i) \cap G_{i-1}$ using the following method, which is described in [23].

Let g be a random element of G_{i-1} . Then we can compute $\phi_i(g) \in Q_i$. Let $\{x_1, \dots, x_k\}$ be our generating set for G_{i-1} , so Q_i is generated by $\{\phi_i(x_1), \dots, \phi_i(x_k)\}$. If Q_i is a soluble group defined by a polycyclic presentation, then we can use the standard methods for that class of groups to write $\phi_i(g)$ as a word $w(\phi_i(x_1), \dots, \phi_i(x_k))$; see, for example, [17, Section 8.3]. If Q_i is insoluble, then we can use constructive recognition in Q_i to do the same. Then $gw(x_1, \dots, x_k)^{-1}$, which we shall call the *residue* of g under ϕ_i , is in the kernel of ϕ_i and it can be shown that, if g is a uniformly distributed random element of G_{i-1} , then the residue of g is a uniformly distributed random element of $\ker(\phi_i) \cap G_{i-1}$. By computing a sufficient number of residues of random elements of G_{i-1} , we hope to construct a generating set for $\ker(\phi_i) \cap G_{i-1}$ modulo $O_p(G)$.

We describe now how we estimate how many random elements of G_{i-1} we need in order to have a high probability of generating $\ker(\phi_i) \cap G_{i-1}$ modulo $O_p(G)$ for $i < k$. For a finite group G , we define $P_k(G)$ to be the probability that k uniformly randomly chosen elements of G generate G . Then our problem reduces to finding a lower bound for $P_k(K_i/O_p(G))$.

To do this, we make use of the superset $M = \{S_1^{e_1}, \dots, S_m^{e_m}, \mathbb{Z}_{p_1}^{t_1}, \dots, \mathbb{Z}_{p_r}^{t_r}\}$ of the composition factors of $G/O_p(G)$ that we were able to estimate after finding the preliminary sequence of maps, as described at the end of Subsection 2.1. After identifying the images Q_k for $1 \leq k \leq i$, we can refine this to a superset of the composition factors of $K_i/O_p(G)$ by removing those factors that we know lie in one the Q_k already found.

There are a number of expositions concerning this problem and most of the results referred to here are due to Pak (unpublished). We shall give only a brief outline here and refer to the reader to [31] for further details. Using elementary arguments, the following result can be proved, which reduces the problem to the case when G is a direct product of isomorphic simple groups.

THEOREM 2.10. *If M , as defined above, is a superset of the composition factors of G , then*

$$P_k(G) \geq \prod_{i=1}^m P_k(S_i^{e_i}) \cdot \prod_{i=1}^r P_k(\mathbb{Z}_{p_i}^{t_i}).$$

For the abelian case, it can be shown that

$$P_k(\mathbb{Z}_p^t) = \prod_{j=1}^t \left(1 - \frac{p^{j-1}}{p^k}\right),$$

and for the nonabelian case we have

$$P_k(S^e) = P_k^e(S) \prod_{j=1}^{e-1} \left(1 - \frac{j|\text{Out}(S)|}{P_k(S)|S|^{k-1}}\right).$$

Because we have identified the nonabelian composition factors of G non-constructively, we can assume that each $|\text{Out}(S)|$ is known, and it is easy to see that

$$P_k(S) \geq 1 - (1 - P_2(S))^{\lfloor \frac{e}{k} \rfloor},$$

so it remains only to estimate $P_2(S)$. For this, we are forced to rely on the following conjecture

CONJECTURE 2.11. *Let S be a nonabelian finite simple group. Then*

$$P_2(S) \geq P_2(\text{Alt}(6)) = \frac{53}{90}.$$

This has not yet been proved, but it is known that $P_2(S) \rightarrow 1$ as $|S| \rightarrow \infty$ (see [26]), and an examination of the simple groups up to order 10^6 indicates that the probability is much closer to 1 than $\frac{53}{90}$ for all but the smallest of the simple groups, so we are highly confident that this conjecture is true.

So, putting all of this together, we have

THEOREM 2.12. *Given a superset of the multiset of composition factors of a group G and $\epsilon > 0$ we can find k such that $P_k(G) \geq 1 - \epsilon$.*

This enables us to compute a reasonable estimate for the number of random elements that are required to generate the kernel of any of the ϕ_i modulo $O_p(G)$ for $i < k$.

2.4. Verifying the kernels

In Subsection 2.3 we constructed a chain of subgroups $G = G_0 \geq G_1 \geq \dots \geq G_{k-1}$ with $G_{i-1} \leq \text{Domain}(\phi_i)$ and we identified the images $Q_i = \phi_i(G_{i-1})$ for $1 \leq i \leq k$. From the construction we know that $G_i \leq \ker(\phi_i)$, and we want to verify that $G_i O_p(G) = (\ker(\phi_i) \cap G_{i-1}) O_p(G)$ for $1 \leq i < k$. We shall also explain in this subsection how to test elements of $\text{GL}(n, q)$ for membership of the subgroups G_i .

We make use of the following theorem.

THEOREM 2.13. *Let $G = \langle \Gamma \rangle$, and let $\pi : G \rightarrow Q$ be an epimorphism. Let $\langle \overline{X} \mid \overline{R} \rangle$ be a presentation for Q and, for each $h \in Q$, let $w_{\overline{X}}(h)$ be a word for h over \overline{X} .*

For each $\overline{x} \in \overline{X}$ choose $x \in G$ with $\pi(x) = \overline{x}$. Define a map $\theta : F_{\overline{X}} \rightarrow G$ (where $F_{\overline{X}}$ is the free group on \overline{X}) by extending the map $\overline{x} \mapsto x$.

Let $K = \langle Y \rangle$ be a subgroup of $\ker(\pi)$. Then $K = \ker(\pi)$ if and only if the following hold:

- (i) $g \cdot \theta(w_{\overline{X}}(\pi(g)))^{-1} \in K$ for all $g \in \Gamma$;
- (ii) $\theta(\overline{\tau}) \in K$ for all $\overline{\tau} \in \overline{R}$;
- (iii) $y^g \in K$ for all $g \in \Gamma$ and $y \in Y$.

Proof. Suppose that the conditions (i), (ii) and (iii) are satisfied. Then (iii) says that $K \trianglelefteq G$, and (i) says that $g \equiv \theta(w_{\overline{X}}(\pi(g))) \pmod K$ for all $g \in \Gamma$. So, for $g = g_1^{\epsilon_1} \dots g_r^{\epsilon_r} \in G$ with $g_i \in \Gamma$ and $\epsilon_i = \pm 1$, we have

$$g \equiv \theta(w_{\overline{X}}(\pi(g_1))^{\epsilon_1} \dots w_{\overline{X}}(\pi(g_r))^{\epsilon_r}) \pmod K.$$

If $g \in \ker(\pi)$, then (since $K \leq \ker(\pi)$), $w_{\overline{X}}(\pi(g_1))^{\epsilon_1} \dots w_{\overline{X}}(\pi(g_r))^{\epsilon_r}$ is a product in $F_{\overline{X}}$ of conjugates of elements of \overline{R} and their inverses. But, by (ii), $\theta(\overline{\tau}) \equiv 1 \pmod K$ for each $\overline{\tau} \in \overline{R}$, so $\theta(w_{\overline{X}}(\pi(g_1))^{\epsilon_1} \dots w_{\overline{X}}(\pi(g_r))^{\epsilon_r}) \equiv 1 \pmod K$ and hence $g \in K$ and $K = \ker(\pi)$.

Conversely, it is easy to see that if $K = \ker(\pi)$ then the three conditions hold. \square

We shall apply this result with $\pi = \phi_i|_{G_{i-1}}$, $G = G_{i-1}$, $Q = Q_i$ and $K = G_i$, but we shall check that the elements to be tested in (i), (ii) and (iii) lie in $G_i O_p(G)$, thereby proving that $G_i O_p(G) = (\ker(\phi_i) \cap G_{i-1}) O_p(G)$. (So we are actually applying the theorem in $G_{i-1} O_p(G)/O_p(G)$.)

For this purpose, we need to be able to test elements of $\text{GL}(n, q)$ for membership in the subgroups $G_i O_p(G)$. We shall describe now how we can use the information computed so far to carry out this membership testing, but we observe that this method requires us to complete the computation of all of the images and kernels before we attempt the verification process.

We can perform membership testing in $O_p(G)$ as follows. If G is irreducible then $O_p(G) = 1$, so assume G to be reducible with m irreducible blocks of dimensions d_1, \dots, d_m . Using the Meataxe we compute a matrix x (in fact such a matrix has already been computed during the application of REDUCIBLEMAPS to G) such that elements $g \in G$ conjugated by x have the form

$$\begin{pmatrix} g_1 & & 0 \\ & \ddots & \\ * & & g_m \end{pmatrix},$$

for $g_i \in \text{GL}(d_i, q)$. Then $g \in O_p(G)$ if and only if $g_i = I_{d_i}$ for $1 \leq i \leq m$.

We now test membership in $G_{k-1} O_p(G)$ as follows. An element $g \in \text{Domain}(\phi_k)$ belongs to $G_{k-1} O_p(G)$ if and only if $\phi_k(g) \in Q_k$ and the residue of g (as defined in Subsection 2.3) under ϕ_k belongs to $O_p(G)$. (This assumes that we can test elements of $\text{GL}(n, q)$ for membership of the domains of the maps ϕ_i , but this presents no difficulty. Most of the maps are based on Aschbacher decompositions, and testing for membership of g in $\text{Domain}(\phi_i)$ is equivalent to checking that g preserves the decomposition in question.)

Using Theorem 2.13 we can now verify $G_{k-1} O_p(G) = (\ker(\phi_{k-1}) \cap G_{k-2}) O_p(G)$. This, in turn, enables us to test membership in $G_{k-2} O_p(G)$: an element $g \in \text{Domain}(\phi_{k-1})$ belongs to $G_{k-2} O_p(G)$ if and only if $\phi_{k-1}(g) \in Q_k$ and the residue of g under ϕ_{k-1} belongs to $G_{k-1} O_p(G)$.

We repeat this process to produce membership tests for $G_i O_p(G)$ for $i = k-1, k-2, \dots, 1, 0$, and to verify that $G_i O_p(G) = (\ker(\phi_i) \cap G_{i-1}) O_p(G)$ for $i > 0$. If at any stage this process fails and we find elements in $(\ker(\phi_i) \cap G_{i-1}) \setminus G_i O_p(G)$, then we add these elements to the generating set for G_i and recompute the series from this point.

It is possible, at the same time, to use the presentations of the Q_i and the conjugation action of G_i on G_{i-1} to compute presentations of $G_i O_p(G)/O_p(G)$ for $i = k-1, k-2, \dots, 0$ (see [17, Subsection 2.4.3] for the theory of this). We need to do this if we wish to prove the correctness of the structure of $O_p(G)$, which we shall be discussing in the next subsection.

2.5. From $O_p(G)$ to 1

We now have a normal series for $G \leq \text{GL}(n, q)$, with $q = p^e$, with last term $O_p(G)$. To complete our analysis of G , we need to construct a chain of subgroups

$$1 \leq H_r \leq H_{r-1} \leq \dots \leq H_0 = O_p(G)$$

with each $H_i \trianglelefteq G$ and each H_i/H_{i+1} an elementary abelian p -group. This can be difficult in matrix groups of dimension greater than about 100, and this topic

ysis of $O_p(G)$, and there is no need to compute a complete generating set for H_0 as elements of G . When $m > 2$, we need such a generating set for H_{i-1} in order to find random elements of H_i , and in order to compute residues during the verification process. However, the generators of H_{i-1} that are calculated as vectors during the application of SPINBASIS can be stored as words, which are products of conjugates by generators of G of the module generators, rather than as matrices. It is straightforward to adapt SPINBASIS to compute these words.

There are many possible improvements to the basic outline of the approach that we have described here, and some of these are being investigated using experimental implementations. In many examples, the H_{i-1}/H_i can be regarded as modules over \mathbb{F}_q rather than just over \mathbb{F}_p , which renders module computations much more efficient when $q = p^e$ for $e > 1$. We should also make use of the fact that the full modules, of which H_{i-1}/H_i are submodules, are direct sums arising in the obvious way from the individual blocks $B_{i,j}$.

2.6. Refining the Normal Series

We have now computed a normal series for G in which the factors are either soluble groups or direct products of isomorphic non-abelian simple groups. Refining this series to a chief series of G is straightforward, and we shall describe how to do this only briefly.

A non-abelian layer in the series is of the form $K_{i-1}/K_i \cong S^m$, where the m direct factors in the product S^m are permuted under the conjugation action of G . The new subgroups that we need to introduce to refine the series correspond to the orbits of this action on these m factors, which are easily computed using the map ϕ_i .

The soluble layers are either the elementary abelian p -groups H_{i-1}/H_i within $O_p(G)$ that were discussed in the previous subsection, or are defined by means of a polycyclic presentation. In the second of these cases we can compute a series of characteristic subgroups with elementary abelian layers, thereby effectively reducing the problem to the case when the layer is an elementary abelian r -group for some prime r . We can then use the conjugation action of G to make this layer into a module for G over \mathbb{F}_r , and use the Meataxe [17, Section 7.4] to find a composition series for the module. The terms in this series correspond to the required refinement of the layer in G .

3. Rearranging the chief series

At this stage, we have computed a chief series

$$G = G_0 \geq G_1 \geq \dots \geq G_n = 1$$

of our given group $G \leq \text{GL}(n, q)$, and our final objective is to replace it with a new chief series that passes through the subgroups $O_\infty(G)$, $\text{soc}^*(G)$ and $\text{Pker}(G)$ that were defined in Section 1. The proof of the following lemma is straightforward, given that these are characteristic subgroups of G .

LEMMA 3.1. *If N is any normal subgroup of any finite group G , then $O_\infty(N) = O_\infty(G) \cap N$, $\text{soc}^*(N) = \text{soc}^*(G) \cap N$ and $\text{Pker}(N) = \text{Pker}(G) \cap N$,*

The method described for rearranging the series in [31], which corresponds to the second author's implementation, involves considering each pair of adjacent chief

factors in the series and interchanging factors firstly in order to bring the soluble factors as low down in the series as possible, which results in the series passing through $O_\infty(G)$, and then secondly to bring the insoluble factors as low down as possible in the series modulo $O_\infty(G)$, which results in the series passing through $\text{soc}^*(G)$, and finally bringing the factors lying in $\text{Pker}(N)$ down to the bottom of the series modulo $\text{soc}^*(G)$. Here we shall describe a variation of this procedure which achieves all three objectives in a single pass upwards through the chief factors, and which we believe will turn out to be more efficient.

3.1. Identifying inner automorphisms

The rearranging process depends critically on our ability to solve the following algorithmic problem efficiently.

PROBLEM 3.2. *Given a finite nonabelian simple group S and an automorphism σ of S , determine whether σ is an inner automorphism and, if so, find $g \in S$ such that σ is conjugation by g .*

We assume that g^σ is easily computable for $g \in S$. In our application, σ will be induced by conjugation by an element of a larger group. We assume also that we have solved the constructive recognition problem for S , which enables us to work within the standard copy \hat{S} of S .

If \hat{X} is a generating set for \hat{S} , then the problem is equivalent to the following. Does there exist $g \in \hat{S}$ with $x^g = x^\sigma$ for all $x \in \hat{X}$? If \hat{S} is a permutation group, then we can solve this by a sequence of conjugacy tests and centraliser calculations within \hat{S} and subgroups of \hat{S} .

So we shall assume that \hat{S} is a quasisimple absolutely irreducible subgroup of $\text{GL}(d, r)$ for some d and r . We then have the additional complication that \hat{S} may contain scalars with $\hat{S}/Z(\hat{S}) \cong S$. So, given a generating set \hat{X} of \hat{S} , we are looking for an element $g \in \hat{S}$ with $x^g \equiv x^\sigma \pmod{Z(\hat{S})}$ for all $x \in \hat{X}$. However, it is readily checked that every finite simple group S is divisible by a prime which does not divide the order of the Schur multiplier of S , so we can choose \hat{X} to consist of elements with order coprime to $|Z(\hat{S})|$ and then the problem reverts to finding $g \in \hat{S}$ with $x^g = x^\sigma$ for all $x \in \hat{X}$.

Deciding whether there exists $g \in \text{GL}(d, r)$ with $x^g = x^\sigma$ for all $x \in \hat{X}$ is equivalent to testing the \hat{S} -modules defined by the matrices in \hat{X} and in \hat{X}^σ for isomorphism, and this can be done readily using the algorithm described in [17, Section 7.5.3]. Since \hat{S} is absolutely irreducible, if g exists then it is unique modulo scalars, and so we can complete the test by checking whether gz lies in \hat{S} for some scalar matrix z , which we are able to do as part of the defining assumptions of the standard copy of a simple group.

3.2. Moving chief factors

Now we present a brief summary of how we rearrange the chief series of G to pass through the three characteristic subgroups defined above.

DEFINITION 3.3. *We say that a chief factor G_{i-1}/G_i of G belongs to (or lies in) a normal subgroup N of G if $G_{i-1} \leq NG_i$.*

Our aim is to rearrange the chief factors of G such that those that belong to $O_\infty(G)$ occur first, followed by all others that belong to $\text{soc}^*(G)$, then by all others that belong to $\text{Pker}(G)$, and finally by those not belonging to $\text{Pker}(G)$.

We consider the chief factors G_{i-1}/G_i for $i = n, n-1, \dots, 1$ in turn. If this factor is soluble then we decide whether we can rearrange the series so that it belongs to $O_\infty(G)$ and, if so, then we carry out this rearrangement within G_{i-1} . If not, then we consider it for membership of $\text{Pker}(G)$ and, if this is the case, then we move it down into $\text{Pker}(G_{i-1})$. If G_{i-1}/G_i is insoluble, then we decide whether we can rearrange the series so that this chief factor belongs to $\text{soc}^*(G)$ and, if so, rearrange the series accordingly.

So, when we come to consider G_{i-1}/G_i , we can assume that those chief factors of G that lie in G_i have already been rearranged to occur in the required order. In other words, the series for G_i passes through $O_\infty(G_i) = O_\infty(G) \cap G_i$, $\text{soc}^*(G_i) = \text{soc}^*(G) \cap G_i$ and $\text{Pker}(G_i) = \text{Pker}(G) \cap G_i$. We assume, in addition, that we have stored the following extra information during the analysis of the chief factors of G within G_i . For those factors in $\text{Pker}(G_i)$ but not in $\text{soc}^*(G_i)$, we store the outer automorphisms of the simple factors of $\text{soc}^*(G_i)/O_\infty(G_i)$ that are induced by the generators of these chief factors. For those that lie outside of $\text{Pker}(G_i)$, we store the permutations of the simple factors of $\text{soc}^*(G_i)/O_\infty(G_i)$ that are induced by the conjugation action of their generators.

(Computing within the permutation group $G_i/\text{Pker}(G_i)$ is easy, but we are assuming also that we can compute effectively within the outer automorphism groups of the finite nonabelian simple groups. Since these are all relatively small soluble groups, this is a reasonable assumption. Indeed, we have already implemented this part of the process for the classical groups.)

We now explain how we analyse the next chief factor G_{i-1}/G_i . We already know a set of elements X_{i-1} of G_{i-1} that generate G_{i-1} modulo G_i . During the analysis, we may change this set X_{i-1} by multiplying its members by suitable elements of G_{i-1} for the purpose of making X_{i-1} a subset of $O_\infty(G_{i-1})$, $\text{soc}^*(G_{i-1})$, or $\text{Pker}(G_{i-1})$ whenever this is possible.

Step 1: does the factor lie in $\text{Pker}(G)$? Let x be the first element of X_{i-1} . We first test whether the conjugation action of x on the simple factors of $\text{soc}^*(G_i)/O_\infty(G_i)$ lies within the permutation group on these factors that we have stored already for G_i . If not, then G_{i-1}/G_i does not lie in $\text{Pker}(G)$, and we compute and store the permutations of these simple factors induced by the elements of X_{i-1} . If it is then, since G_{i-1}/G_i is a chief factor of G , the same will be true for each element of X_{i-1} , and we multiply each such element by the inverse of an element of G_i that induces the same permutation and thereby effectively move G_{i-1}/G_i into $\text{Pker}(G_{i-1})$.

Step 2: does it lie in $\text{soc}^(G)$?* If the chief factor lies in $\text{Pker}(G_{i-1})$, then we test whether the outer automorphisms of the simple factors of $\text{soc}^*(G_i)/O_\infty(G_i)$ all lie in the subgroups of these outer automorphism groups that we have stored already for G_i . If not, then G_{i-1}/G_i is not a chief factor of $\text{soc}^*(G)$, and we compute and store the outer automorphisms of the simple factors induced by the elements of X_{i-1} . If it is (which will necessarily be the case if G_{i-1}/G_i is not soluble), then the same will be true for each element of X_{i-1} , and we replace each such element by the result of multiplying it by the inverse of an element of G_i that induces the same outer automorphism. We thereby move G_{i-1}/G_i into $\text{soc}^*(G_{i-1})$.

Step 3: does it lie in $O_\infty(G)$? If the chief factor now lies in $\text{soc}^*(G_{i-1})$, then the elements of X_{i-1} all induce inner automorphisms of the simple factors of $\text{soc}^*(G_i)/O_\infty(G_i)$, and we can identify the corresponding conjugating elements of these simple factors using the method described in Subsection 3.1. We then replace each element of X_{i-1} by the result of multiplying it by the inverse of an appropriate

element of $\text{soc}^*(G_i)$, after which the elements of X_{i-1} will all centralise $\text{soc}^*(G_i)$ modulo $O_\infty(G_i)$. So now, if G_{i-1}/G_i is soluble, then we will have effectively moved it into $O_\infty(G_{i-1})$, whereas if it is insoluble, we will have achieved the desirable effect of making the group generated by X_{i-1} modulo $O_\infty(G_{i-1}) = O_\infty(G_i)$ equal to some of the simple factors of $\text{soc}^*(G_i)/O_\infty(G_i)$.

4. Some timings

To conclude, we present some timings for the second author's MAGMA implementation of the algorithms described in Sections 2 and 3. The computations were run on an AMD Opteron Model 152 processor running at 2.6 GHz with 4 GB of memory. The times in the columns labelled **CSTime** and **SSTime** in the table are respectively for computing a chief series, and a chief series passing through the three subgroups $O_\infty(G)$, $\text{soc}^*(G)$ and $\text{Pker}(G)$. However, the latter times also include some additional computations, such as polycyclic presentations of $O_\infty(G)$, which accounts for the longer time in the soluble example $(\text{GL}(2, 3) \wr S_4) \wr S_4$.

It should be noted that presentations for the larger sporadic simple groups are currently unavailable in MAGMA, so the complete verification algorithms of Section 2.4 have not been run on these groups. Unfortunately, implementations of constructive recognition algorithms were only available for $\text{PSL}(2, q)$, $\text{Sz}(q)$ and $\text{Alt}(n)$ and most of the sporadic groups, and we had to use the default Schreier-Sims algorithm for other finite simple groups. This considerably restricts the scope of the current implementation, and we hope to extend this scope eventually by making use of further constructive recognition methods. In particular, the simple groups $\text{PSL}(3, 3^7)$ and $\text{P}\Omega(9, 9)$, which occur as composition factors of examples in the table, are about at the limit of what we can handle at present.

G	d	q	CSTime	SSTime
$3^{1+12} \cdot 2 \text{ Suz} \cdot 2$	78	3	4.5	4.5
$2^{576} \cdot \text{Co}_1 \wr S_2$	96	2	13.4	48.5
$\text{GL}(6, 5) \wr \text{Sym}(15)$	90	5	14.5	31.3
$(\text{GL}(2, 3) \wr S_4) \wr S_4$	32	3	1.8	40.7
$\text{Ly} \wr S_2$	222	5	33.9	38.4
$3^{18} \cdot \text{PSL}(3, 7)^6 \cdot 9 \cdot 2$	18	7	4.3	5.2
$2^{32} \cdot \text{SL}(4, 5) \cdot 2^4 \cdot \text{GL}(2, 4)$	64	5	4.3	4.4
$\text{GL}(3, 3^7) \cdot 7$	21	3	89.9	89.9
$\text{SO}(9, 9) \cdot 2$	18	3	157.5	158.0
$2^{1+14} \cdot \text{Sp}(14, 2)$	128	5	18.2	18.5
$\text{GL}(3, 7) \otimes S_5$	243	7	100.4	121.0
$3^{1+6} \cdot \text{Sp}(6, 3) \otimes \text{GL}(5, 5)$	135	25	97.5	109.5
$(\text{GL}(2, 5) \otimes \text{GL}(3, 5)) \wr S_{15}$	90	5	19.2	97.2

Table 1: Table of timings for computing SR-Data

Here is a brief description of the examples in the table. The first two are reducible groups in which $O_p(G)$ is non-trivial. The second of these also involves the imprimitive matrix group $\text{Co}_1 \wr S_2$ of degree 48. In both of these two examples, REDUCIBLEMAPS duplicated effort by analysing the same nonabelian chief factor on different irreducible constituents of the input group.

The following five examples are all imprimitive, with $3^{18}.\text{PSL}(3, 7)^6.9.2 \leq (\text{SL}(3, 7) \wr S_2) \wr D_{18}$ and $2^{32}.\text{PSL}(4, 5).2^4.\text{GL}(2, 4) \leq \text{SL}(4, 5) \wr \text{AGL}(2, 4)$. The next two examples $\text{GL}(3, 3^7).7$ and $\text{SO}(9, 9).2$ are semilinear, $2^{1+14}.\text{Sp}(14, 2)$ is the normaliser of a symplectic type 2-group and $\text{GL}(3, 7) \otimes S_5$ is tensor induced. The example $3^{1+6}.\text{Sp}(6, 3) \otimes \text{GL}(5, 5)$ is a tensor product of a normaliser of an extraspecial group with $\text{GL}(5, 5)$, and $(\text{GL}(2, 5) \otimes \text{GL}(3, 5)) \wr S_{15}$ is imprimitive with kernel 15 copies of the tensor product $\text{GL}(2, 5) \otimes \text{GL}(3, 5)$.

References

1. M. ASCHBACHER, ‘On the maximal subgroups of the finite classical groups’, *Invent. Math.* 76 (1984) 469–514. [226](#)
2. H. BÄÄRNHIELM, ‘Recognising the Ree groups in their natural representations’, Preprint. [226](#)
3. H. BÄÄRNHIELM, ‘Recognising the Suzuki groups in their natural representations’, *J. Algebra* 300 (2006) 171–198. [226](#)
4. L. BABAI and A. SHALEV, ‘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. 8 (de Gruyter, Berlin, 2001) 39–62. [238](#)
5. LÁSZLÓ BABAI, ‘Local expansion of vertex-transitive graphs and random generation in finite groups’, *Theory of Computing*, Los Angeles, 1991 (Association for Computing Machinery, New York, 1991) pp. 164–174. [224](#)
6. S. BRATUS and I. PAK, ‘Fast constructive recognition of a black box group isomorphic to A_n or S_n using Goldbach’s conjecture’, *J. Symbolic Comput.* 29 (2000) 33–57. [226](#)
7. J.J. CANNON, B.C. COX and D.F. HOLT, ‘Computing chief series, composition series and socles in large permutation groups’, *J. Symbolic Comput.* 24 (1997) 285–301. [229](#)
8. J.J. CANNON and D.F. HOLT, ‘Automorphism group computation and isomorphism testing in finite groups’, *J. Symbolic Comput.* 35 (2003) 241–267. [223](#)
9. J.J. CANNON and D.F. HOLT, ‘Computing maximal subgroups of finite groups’, *J. Symbolic Comput.* 37 (2004) 598–609. [223](#)
10. J.J. CANNON and D.F. HOLT, ‘Computing conjugacy class representatives in permutation groups’, *J. Algebra* 300 (2006) 213–222. [223](#)
11. F. CELLER, C.R. LEEDHAM-GREEN, S.H. MURRAY, A.C. NIEMEYER and E.A. O’BRIEN, ‘Generating random elements of a finite group’, *Comm. Algebra* 23 (1995) 4931–4948. [225](#)
12. FRANK CELLER and C.R. LEEDHAM-GREEN, ‘Calculating the order of an invertible matrix’, *Groups and Computation II* (DIMACS, 1995), Amer. Math. Soc. DIMACS Series 28 (American Mathematical Society, Providence, RI, 1997) 55–60. [224](#)
13. M.D.E. CONDER, C.R. LEEDHAM-GREEN and E.A. O’BRIEN, ‘Constructive recognition of $\text{PSL}(2, q)$ ’, *Trans. Amer. Math. Soc.* 358 (2006) 1203–1221. [226](#)

14. S.P. GLASBY, C.R. LEEDHAM-GREEN and E.A. O'BRIEN, 'Writing projective representations over subfields', *J. Algebra* 295 (2006) 1203–1221. [233](#)
15. D. GORENSTEIN, *Finite Groups* (Harper & Row, New York, Evanston, London, 1968). [233](#), [234](#)
16. P.E. HOLMES, S.A. LINTON, E.A. O'BRIEN, A.J.E. RYBA and R.A. WILSON, 'Constructive membership testing in black-box groups', Preprint. [226](#)
17. DEREK F. HOLT, BETTINA EICK and EAMONN A. O'BRIEN, *Handbook of computational group theory* (Chapman and Hall/CRC, London, 2005). [223](#), [224](#), [225](#), [226](#), [241](#), [243](#), [244](#), [245](#), [246](#)
18. DEREK F. HOLT, C.R. LEEDHAM-GREEN, E.A. O'BRIEN and SARAH REES, 'Computing matrix group decompositions with respect to a normal subgroup', *J. Algebra* 184 (1996) 818–838. [234](#), [235](#)
19. DEREK F. HOLT, C.R. LEEDHAM-GREEN, E.A. O'BRIEN and SARAH REES, 'Testing matrix groups for primitivity', *J. Algebra* 184 (1996) 795–817. [231](#)
20. DEREK F. HOLT and SARAH REES, 'Testing modules for irreducibility', *J. Austral. Math. Soc. Ser. A* 57 (1994) 1–16. [232](#)
21. W.M. KANTOR and À. SERESS, 'Black box classical groups', *Mem. Amer. Math. Soc.* 149 (2001), no. 708. [226](#)
22. P. KLEIDMAN and M. LIEBECK, *The Subgroup Structure of the Finite Classical Groups*, London Mathematical Society Lecture Notes Series 129 (Cambridge University Press, Cambridge, 1990). [236](#)
23. C.R. LEEDHAM-GREEN, 'The computational matrix group project', *Groups and Computation III*, Columbus, OH, 1999 (de Gruyter, Berlin, 2001) 229–248. [223](#), [231](#), [241](#)
24. C.R. LEEDHAM-GREEN and E.A. O'BRIEN, 'Recognising tensor products of matrix groups', *Internat. J. Algebra Comput.* 7 (1997) 541–559. [233](#)
25. C.R. LEEDHAM-GREEN and E.A. O'BRIEN, 'Recognising tensor induced matrix groups', *J. Algebra* 253 (2002) 14–30. [236](#), [238](#)
26. M. LIEBECK and A. SHALEV, 'The probability of generating a finite simple group', *Geom. Dedicata* 56 (1995) 103–113. [242](#)
27. F. LUEBECK, K. MAGAARD and E.A. O'BRIEN, 'Constructive recognition of $SL_3(q)$ ', Preprint. [226](#)
28. ALICE C. NIEMEYER and CHERYL E. PRAEGER, 'A recognition algorithm for classical groups over finite fields', *Proc. London Math. Soc.* (3) 77 (1998) 117–169. [236](#)
29. E.A. O'BRIEN, 'Towards effective algorithms for linear groups', *Finite Geometries, Groups and Computation*, Colorado, 2004 (de Gruyter, Berlin, 2006) 163–190. [223](#), [225](#)
30. À. SERESS, *Permutation Group Algorithms* (Cambridge University Press, Cambridge, 2003). [225](#)
31. MARK STATHER, 'Algorithms for computing with finite matrix groups', PhD thesis, University of Warwick, 2006. [224](#), [240](#), [241](#), [245](#)

- 32.** M.J. STATHER, ‘Sylow subgroups in matrix groups’, *J. Algebra*, to appear. [224](#)
- 33.** D.E. TAYLOR, *The Geometry of the Classical Groups*, Sigma Series in Pure Mathematics 9 (Heldermann, Berlin, 1992). [236](#)

Derek F. Holt dfh@maths.warwick.ac.uk

Mark J. Stather markstather@hotmail.com

Mathematics Institute

University of Warwick

Coventry

CV4 7AL

United Kingdom