Algorithms for matrix groups

E.A. O'Brien

Department of Mathematics, University of Auckland, Auckland, New Zealand
Email: obrien@math.auckland.ac.nz

Abstract

Existing algorithms have only limited ability to answer structural questions about subgroups G of $\mathrm{GL}(d, F)$, where F is a finite field. We discuss new and promising algorithmic approaches, both theoretical and practical, which as a first step construct a chief series for G.

1 Introduction

Research in Computational Group Theory has concentrated on four primary areas: permutation groups, finitely-presented groups, soluble groups, and matrix groups. It is now possible to study the structure of permutation groups having degrees up to about ten million; Seress [97] describes in detail the relevant algorithms. We can compute *useful* descriptions for quotients of finitely-presented groups; as one example, O'Brien & Vaughan-Lee [90] computed a power-conjugate presentation for the largest finite 2-generator group of exponent 7, showing that it has order 7^{20416}. Practical algorithms for the study of polycyclic groups are described in [59, Chapter 8].

We contrast the success in these areas with the paucity of algorithms to investigate the structure of matrix groups. Let $G = \langle X \rangle \leq \mathrm{GL}(d, F)$ where $F = \mathrm{GF}(q)$. Natural questions of interest to group-theorists include: What is the order of G? What are its composition factors? How many conjugacy classes of elements does it have? Such questions about a subgroup of S_n, the symmetric group of degree n, are answered both theoretically and practically using highly effective polynomial-time algorithms. However, for linear groups these can be answered only in certain limited contexts. As one indicator, it is difficult (using standard functions) to answer such questions about $\mathrm{GL}(8, 7)$ using either of the major computational algebra systems, GAP [46] and Magma [16].

A major topic of research over the past 15 years, the so-called "matrix recognition" project, has sought to address these limitations by developing effective well-understood algorithms for the study of such groups. A secondary goal is to realise the performance of these algorithms in practice, via publicly available implementations.

Thanks to Peter Brooksbank, Heiko Dietrich, Stephen Glasby, Derek Holt, Colva Roney-Dougal and Ákos Seress for their comments and corrections to the paper. This work was partially supported by the Marsden Fund of New Zealand via grant UOA721.

Two approaches dominate. The *black-box approach*, discussed in Section 4, aims to construct a characteristic series $\mathcal{C}$ of subgroups for G which can be readily refined to provide a chief series; the associated algorithms are independent of the given representation. The *geometric approach*, discussed in Section 5, aims to exploit the natural linear action of G on its underlying vector space to construct a composition series for G; the associated algorithms exploit the linear representation of G. Both approaches rely on the solution of certain key tasks for simple groups which we discuss in Section 3; we survey their solutions in Sections 6–9. Presentations for the groups of Lie type on certain *standard generators* are used to ensure correctness; these are discussed in Section 10.

As we demonstrate in Section 11, the geometric approach is realised via a *composition tree*. In practice, the composition series produced from the geometric approach is readily modified to produce a chief series of G exhibiting $\mathcal{C}$. In Section 12 we consider briefly algorithms which exploit the chief series and its associated *Trivial Fitting* paradigm to answer structural questions about G. While it is not yet possible to make definitive statements about the outcome of this project, a realistic and achievable goal is to provide algorithms to answer many questions for linear groups of "small" degree, say up to degree 20 defined over moderate-sized fields.

In this paper, we aim to supplement and update the related surveys [65], [72] and [91]. Its length precludes comprehensiveness. For example, we consider neither nilpotent nor solvable linear groups. Nor do we discuss the algorithms of Detinko and Flannery and others to study finitely generated matrix groups defined over infinite fields. The excellent survey [43] addresses both omissions.

2 Basic concepts

We commence with a review of basic concepts.

2.1 Complexity

If f and g are real-valued functions defined on the positive integers, then $f(n) = O(g(n))$ means $|f(n)| < C|g(n)|$ for some positive constant C and all sufficiently large n.

One measure of performance is that an algorithm is *polynomial in the size of the input*. If $G = \langle X \rangle \leq \mathrm{GL}(d, q)$, then the size of the input is $|X| d^2 \log q$, since each of the d^2 entries in a matrix requires $\log q$ bits.

2.2 Black-box groups

The concept of a *black-box group* was introduced in [6]. In this model, group elements are represented by bit-strings of uniform length; the only group operations permissible are multiplication, inversion, and checking for equality with the identity element. Permutation groups and matrix groups defined over finite fields are covered by this model.

Seress [97, p. 17] defines a *black-box algorithm* as one which does not use specific features of the group representation, nor particulars of how group operations are performed; it can only use the operations listed above. However, a common assumption is that *oracles* are available to perform certain tasks – usually those not known to be solvable in polynomial time.

One such is a *discrete log oracle*: for a given non-zero $\mu \in \mathrm{GF}(q)$ and a fixed primitive element ω of $\mathrm{GF}(q)$, it returns the unique integer k in the range $1 \leq k < q$ for which $\mu = \omega^k$. The most efficient algorithms for this task run in sub-exponential time (see [98, Chapter 4]).

If the elements of a black-box group G are represented by bit-strings of uniform length n, then n is the *encoding length* of G and $|G| \leq 2^n$. If G is described by a bounded list of generators, then the size of the input to a black-box algorithm is $O(n)$. If G also has Lie rank r and is defined over a field of size q, then $|G| \geq (q-1)^r$, so both r and $\log q$ are $O(n)$.

2.3 Algorithm types and random elements

Most algorithms for linear groups are *randomised*: they rely on random selections. A *Monte Carlo* algorithm is a randomised algorithm that, with prescribed probability less than $1/2$, may return an incorrect answer to a decision question. A *Las Vegas* algorithm is one that never returns an incorrect answer, but may report failure with probability less than some specified value $\epsilon \in (0,1)$. At the cost of n iterations, the probability of a correct answer can be increased to $1-\epsilon^n$. We refer the reader to [5] for a discussion of these concepts.

Monte Carlo algorithms to construct the normal closure of a subgroup and the derived group of a black-box group are described in [97, Chapter 2].

Many algorithms use random search in a group $G \leq \mathrm{GL}(d,q)$ to find elements having prescribed property $\mathcal{P}$. Examples of $\mathcal{P}$ are having a characteristic polynomial with a factor of degree greater than $d/2$, or order divisible by a prescribed prime.

A common feature is that these algorithms depend on detailed analysis of the *proportion* of elements of finite simple groups satisfying $\mathcal{P}$. Assume we determine a lower bound, say $1/k$, for the proportion of elements in G satisfying $\mathcal{P}$. To find an element satisfying $\mathcal{P}$ by random search with probability of failure less than a given $\epsilon \in (0,1)$, we choose a sample of uniformly distributed random elements in G of size at least $\lceil \log_e(1/\epsilon) \rceil k$.

Following [97, p. 24], an algorithm constructs an ϵ-uniformly distributed random element x of a finite group G if $(1-\epsilon)/|G| < \mathrm{Prob}(x = g) < (1+\epsilon)/|G|$ for all $g \in G$; if $\epsilon < 1/2$, then the algorithm constructs *nearly uniformly distributed* random elements of G. Babai [4] presents a black-box Monte Carlo algorithm to construct such elements in polynomial time. An alternative is the *product replacement algorithm* of Celler *et al.* [34]. That this runs in polynomial time was established by Pak [92]. Its implementations in GAP and Magma are widely used. For a discussion of both algorithms, see [97, pp. 26–30]. Another algorithm, proposed by Cooperman [39], was analysed by Dixon [44].

2.4 Some basic operations

Consider the task of multiplying two $d \times d$ matrices. Its complexity is $O(d^\omega)$ field operations, where $\omega = 3$ if we employ the traditional algorithm. Strassen's divide-and-conquer algorithm [100] reduces ω to $\log_2 7$ but at a cost: namely, the additional intricacy of an implementation and larger memory demands. Coppersmith & Winograd's result [40] that ω can be smaller than 2.376 remains of limited practical significance.

We can compute large powers m of a matrix g in at most $2 \lfloor \log_2 m \rfloor$ multiplications by the standard doubling algorithm: $g^m = g^{m-1}g$ if m is odd and $g^m = g^{(m/2)2}$ if m is even.

Lemma 2.1

(i) *Multiplication and division operations for polynomials of degree d defined over* $\mathrm{GF}(q)$ *can be performed deterministically in* $\mathrm{O}(d \log d \log\log d)$ *field operations. Using a Las Vegas algorithm, such a polynomial can be factored into its irreducible factors in* $\mathrm{O}(d^2 \log d \log\log d \log(qd))$ *field operations.*

(ii) *Using Las Vegas algorithms, both the characteristic and minimal polynomial of* $g \in \mathrm{GL}(d,q)$ *can be computed in* $\mathrm{O}(d^3 \log d)$ *field operations.*

For the cost of polynomial operations, see [101, §8.3, §9.1, Theorem 14.14]. Characteristic and minimal polynomials can be computed in the claimed time using the Las Vegas algorithms of [2, 69] and [47] respectively. Neunhöffer & Praeger [87] describe Monte Carlo and deterministic algorithms to construct the minimal polynomial; these have complexity $O(d^3)$ and $O(d^4)$ respectively and are implemented in GAP.

2.5 The pseudo-order of a matrix

To determine the order of $g \in \mathrm{GL}(d,q)$ currently requires factorisation of numbers of the form $q^i - 1$, a problem generally believed not to be solvable in polynomial time. Since $\mathrm{GL}(d,q)$ has elements of order $q^d - 1$ (namely, Singer cycles), it is not practical to compute powers of g until we obtain the identity.

Celler & Leedham-Green [35] present the following algorithm to compute the order of $g \in \mathrm{GL}(d,q)$.

- Compute a "good" multiplicative upper bound B for $|g|$.
- Factorise $B = \prod_{i=1}^m p_i^{\alpha_i}$ where the primes p_i are distinct.
- If $m = 1$, then calculate $g^{p_1^j}$ for $j = 1, 2, \ldots, \alpha_1 - 1$ until the identity is constructed.
- If $m > 1$ then express $B = uv$, where u, v are coprime and have approximately the same number of distinct prime factors. Now g^u has order k dividing v and g^k has order ℓ say dividing u, and the order of g is $k\ell$. Hence the algorithm proceeds by recursion on m.

They prove the following:

Theorem 2.2 *If we know a factorisation of B, then the cost of the algorithm is $O(d^4 \log q \log\log q^d)$ field operations.*

We can readily compute in polynomial time a "good" multiplicative upper bound for $|g|$. Let the factorisation over GF(q) of the minimal polynomial $f(x)$ of g into powers of distinct irreducible monic polynomials be given by $f(x) = \prod_{i=1}^{t} f_i(x)^{n_i}$, where $\deg(f_i) = e_i$. Then $|g|$ divides $B := \mathrm{lcm}(q^{e_1} - 1, \ldots, q^{e_t} - 1) \times p^{\beta}$, where $\beta = \lceil \log_p \max n_i \rceil$ and GF(q) has characteristic p.

The GAP and Magma implementations of the order algorithm are very efficient, and use databases of factorisations of numbers of the form $q^i - 1$, prepared as part of the Cunningham Project [20].

From B, we can learn in polynomial time the *exact* power of 2 (or of any specified prime) which divides $|g|$. By repeated division by 2, we write $B = 2^m b$ where b is odd. Now we compute $h = g^b$, and determine (by powering) its order, which divides 2^m. In particular, we can deduce if g has *even order*.

For most applications, it suffices to know the *pseudo-order* of $g \in \mathrm{GL}(d, q)$, a refined version of B. Leedham-Green & O'Brien [73, Section 2] define this formally and show that it can be computed in $\mathrm{O}(d^3 \log d + d^2 \log d \log\log d \log q)$ field operations.

2.6 Straight-line programs

One may intuitively think of a *straight-line program* (SLP) for $g \in G = \langle X \rangle$ as an efficiently stored word in X that evaluates to g; for a formal definition and discussion of their significance, see [97, p. 10]. While the length of a word in a given generating set constructed in n multiplications and inversions can increase exponentially with n, the length of the corresponding SLP is *linear* in n. Babai & Szemerédi [6] prove that every element of a finite group G has an SLP of length $O(\log^2 |G|)$ in every generating set. Both Magma and GAP use SLPs.

3 The major tasks

We identify three major problems for a (quasi)simple group $G = \langle X \rangle$. (Recall that G is *quasisimple* if G is perfect and $G/Z(G)$ is simple.)

(i) The *naming problem*: determine the name of G.

(ii) The *constructive recognition problem*: construct an isomorphism (possibly modulo scalars) between G and a "standard copy" of G.

(iii) The *constructive membership problem*: if $x \in G$, then write x as an SLP in X.

An algorithm to solve (i) may simply establish that G *contains* a named group as its unique non-abelian composition factor. Such information is useful: if we learn that G is a member of a particular family of finite simple groups, then we can apply algorithms to G which are specific to this family.

For each finite (quasi)simple group, we designate one explicit representation as its *standard copy* and designate a particular generating set as its *standard generators.*

For example, the standard copy of A_n is on n points; its standard generators are $(1, 2, 3)$ and either of $(3, \ldots, n)$ or $(1, 2)(3, \ldots, n)$ according to the parity of n.

To aid exposition, we focus on one common situation. Consider the classical groups, where the standard copy is the natural representation. Let $H \leq \mathrm{GL}(d, q)$ denote the natural representation of a classical group. Given as input an arbitrary permutation or projective matrix representation $G = \langle X \rangle$, a constructive recognition algorithm sets up an isomorphism between G and $H/Z(H)$.

To enable this construction, we define standard generators $\mathcal{S}$ for H. Assume we can construct the image $\bar{\mathcal{S}}$ of these standard generators in G as SLPs in X. We may now define the isomorphism $\phi : H/Z(H) \to G$. If we can solve the constructive membership problem in H, then the image in G of an arbitrary element of H can be constructed: if h has a known SLP in $\mathcal{S}$ then $\phi(h)$ is the SLP evaluated in $\bar{\mathcal{S}}$. Similarly if we can solve the constructive membership problem in G, then we can define $\tau : G \to H/Z(H)$. We say that these isomorphisms are *constructive.*

4 The black-box approach

The *black-box group approach*, initiated and pioneered by Babai and Beals (see [7] for an excellent account), focuses on the abstract structure of a finite group G. Recall, for example from [59, pp. 31–32], that G has a characteristic series of subgroups:

$$1 \leq O_\infty(G) \leq S^*(G) \leq P(G) \leq G$$

where

- $O_\infty(G)$ is the largest soluble normal subgroup of G, the *soluble radical*;
- $S^*(G)/O_\infty(G)$ is the socle of $G/O_\infty(G)$ and equals $T_1 \times \cdots \times T_k$, where each T_i is non-abelian simple;
- $\phi : G \to \mathrm{Sym}(k)$ is the representation of G induced by conjugation on $\{T_1, \ldots, T_k\}$, and $P(G) = \ker \phi$;
- $P(G)/S^*(G) \leq \mathrm{Out}(T_1) \times \cdots \times \mathrm{Out}(T_k)$ and so is soluble (by the proof of the Schreier conjecture);
- $G/P(G) \leq \mathrm{Sym}(k)$ where $k \leq \log |G| / \log 60$.

In summary, the black-box approach aims to construct this characteristic series $\mathcal{C}$ for $G \leq \mathrm{GL}(d, q)$ using black-box algorithms. In 2009, as a culmination of 25 years of work, Babai, Beals & Seress [10] proved that, subject to the existence of a discrete log oracle and the ability to factorise integers of the form $q^i - 1$ for $1 \leq i \leq d$, there exist black-box polynomial-time Las Vegas algorithms to construct $\mathcal{C}$ for a large class of matrix groups. Building on results of [9], [56], [81] and [93], they solve the major tasks identified in Section 3 (and others) for groups in this class. We refer the reader to [7] and [10] for details.

In Section 12 we consider how the black-box approach underpins various practical algorithms for matrix groups.

5 Geometry following Aschbacher

By contrast, the *geometric approach* investigates whether a linear group satisfies natural and inherent geometric properties in *its action on the underlying space.* A classification of the maximal subgroups of classical groups by Aschbacher [3] underpins this approach. Let Z denote the subgroup of scalar matrices of $G \leq \mathrm{GL}(d, q)$. Then G is *almost simple modulo scalars* if there is a non-abelian simple group T such that $T \leq G/Z \leq \mathrm{Aut}(T)$, the automorphism group of T. We paraphrase Aschbacher's theorem as follows.

Theorem 5.1 *Let V be the vector space of row vectors on which $\mathrm{GL}(d, q)$ acts, and let Z be the subgroup of scalar matrices of G. If G is a subgroup of $\mathrm{GL}(d, q)$, 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 $V_1 \oplus V_2 \oplus \cdots \oplus V_r$ of $r > 1$ subspaces of dimension s, which are permuted transitively by G, and so $G \leq \mathrm{GL}(s, q) \wr \mathrm{Sym}(r)$.*

C3. *G acts on V as a group of semilinear automorphisms of a (d/e)-dimensional space over the extension field $\mathrm{GF}(q^e)$ for some $e > 1$, and so G embeds in $\Gamma\mathrm{L}(d/e, q^e)$. (This includes the class of "absolutely reducible" linear groups, where G embeds in $\mathrm{GL}(d/e, q^e)$.)*

C4. *G preserves a decomposition of V as a tensor product $U \otimes W$ of spaces of dimensions $d_1, d_2 > 1$ over $\mathrm{GF}(q)$. Then G is a subgroup of the central product of $\mathrm{GL}(d_1, q)$ and $\mathrm{GL}(d_2, q)$.*

C5. *G is definable modulo scalars over a subfield: for some proper subfield $\mathrm{GF}(q')$ of $\mathrm{GF}(q)$, $G^g \leq \mathrm{GL}(d, q').Z$, for some $g \in \mathrm{GL}(d, q)$.*

C6. *For some prime r, $d = r^n$, and G is contained in the normaliser of an extraspecial group of order r^{2n+1}, or of a group of order 2^{2n+2} and symplectic-type (namely, the central product of an extraspecial group of order 2^{2n+1} with a cyclic group of order 4, amalgamating central involutions).*

C7. *G is tensor-induced: G preserves a decomposition of V as $V_1 \otimes V_2 \otimes \cdots \otimes V_m$, where each V_i has dimension $r > 1$, $d = r^m$, and the set of V_is is permuted transitively by G, and so $G/Z \leq \mathrm{PGL}(r, q) \wr \mathrm{Sym}(m)$.*

C8. *G normalises a classical group in its natural representation.*

C9. *G is almost simple modulo scalars.*

We summarise the outcome: a linear group preserves some natural linear structure in its action on the underlying space and has a normal subgroup related to this structure, or it is almost simple modulo scalars.

In broad outline, it suggests that a first step in investigating a linear group is to determine (at least one of) its categories in the Aschbacher classification. If a category is recognised, then we can investigate the group structure more completely using algorithms designed for this category. Usually, we have reduced the size and nature of the problem. For example, if $G \leq \mathrm{GL}(d, q)$ acts imprimitively, then we obtain a permutation representation of degree dividing d for G; if G preserves a tensor product, we obtain two linear groups of smaller degree. If a proper normal subgroup N exists, we investigate N and G/N recursively, ultimately obtaining a composition series for G.

The *base cases* for the geometric approach are groups in C8 and C9: classical groups in their natural representation, and other groups which are almost simple modulo scalars. Liebeck [74] proved that "most" maximal subgroups of $\mathrm{GL}(d, q)$ have order at most q^{3d}, small by contrast with $|\mathrm{GL}(d, q)|$; the exceptions are known. Further, the absolutely irreducible representations of degree at most 250 of all quasisimple finite groups are now explicitly known: see Hiss & Malle [55] and Lübeck [78].

Landazuri & Seitz [71] and Seitz & Zalesskii [96] provide lower bounds for degrees of non-linear irreducible projective representations of finite Chevalley groups. They show that a faithful projective representation in cross characteristic has degree that is polynomial in the *size* of the defining characteristic. Hence our principal focus is on matrix representations in *defining characteristic.*

5.1 Deciding membership of an Aschbacher category

In [91] we reported in detail on the algorithms developed to decide if $G = \langle X \rangle \leq \mathrm{GL}(d, q)$, acting on the underlying vector space V, lies in one of the first seven Aschbacher categories. Consequently we only update that report. In Section 6.1 we report on a Monte Carlo algorithm which decides if G is in C8.

5.1.1 Reducible groups

The MeatAxe algorithm of Holt & Rees [57] is Las Vegas and has complexity $O(d^3(d \log d + \log q))$. A key component is a search in the $\mathrm{GF}(q)$-algebra generated by X for a random element whose characteristic polynomial has an irreducible factor of multiplicity one. The analysis of [57], completed in [64], shows that the proportion of such elements is at least 0.08.

A matrix A over $\mathrm{GF}(q)$ for which the underlying vector space, considered as a $\mathrm{GF}(q)[A]$-module, has at least one cyclic primary component is *f-cyclic.* Glasby & Praeger [49] present and analyse a test for the irreducibility of G using the set of f-cyclic matrices in G, which contains as a proper subset those considered in [57].

5.1.2 C3 and C5

Holt *et al.* [58] present the Smash algorithm: effectively an algorithmic realisation of Clifford's theorem [36] about decompositions of V preserved by a non-scalar normal subgroup of G.

If G acts absolutely irreducibly, then we apply Smash to a normal generating set for its derived group G' to decide if G acts semilinearly. The polynomial-time algorithm of [48] to decide membership in C5 requires that G' acts absolutely irreducibly on V. Implementations of both are available in Magma.

Carlson, Neunhöffer & Roney-Dougal [33] present a polynomial-time Las Vegas algorithm to find a non-trivial "reduction" of an irreducible group G that either lies in C3 or C5, or whose derived group does not act absolutely irreducibly on V. In particular, they deduce that G is in one of C2, C3, C4, or C5; or obtain a homomorphism from G to $\mathrm{GF}(q)^\times$. An implementation is available in GAP.

5.1.3 Normalisers of p-groups

If G is in C6, then it normalises a group R of order either r^{2n+1} (extraspecial) or 2^{2n+2} (symplectic-type).

Brooksbank, Niemeyer & Seress [25] present an algorithm to produce a non-trivial homomorphism from G to either $\mathrm{GL}(2m, r)$ or $\mathrm{Sym}(r^m)$ where $1 \leqslant m \leqslant n$. They prove that this algorithm runs in polynomial time when G is either the full normaliser in $\mathrm{GL}(d, q)$ of R, or $d = r^2$. The special case where $d = r$ was solved by Niemeyer [89]. Implementations are available in GAP and Magma.

5.1.4 Towards polynomial time?

A major theoretical challenge is the following: decide membership of a given group $G \leq \mathrm{GL}(d, q)$ in a *specific* Aschbacher category in polynomial time. This we can always do for C1 and C8, and sometimes for C3, C5 and C6.

Recently Neunhöffer [86] has further developed and analysed variations of the Smash algorithm, and has also reformulated the Aschbacher categories to facilitate easier membership problems. This work and the "reduction algorithms" of [25] and [33] suggest that, subject to the availability of discrete log and integer factorisation oracles, it may be possible using matrix group algorithms to construct in polynomial time the composition factors of G. We contrast this with the results obtained in the black-box context [10].

6 Naming algorithms

Let b and e be positive integers with $b > 1$. A prime r dividing $b^e - 1$ is a *primitive prime divisor* of $b^e - 1$ if $r|(b^e - 1)$ but $r \not| (b^i - 1)$ for $1 \leq i < e$. Zsigmondy [107] proved that $b^e - 1$ has a primitive prime divisor unless $(b, e) = (2, 6)$, or $e = 2$ and $b + 1$ is a power of 2. Recall that

$$|\mathrm{GL}(d, q)| = q^{\binom{d}{2}} \prod_{i=1}^{d} (q^i - 1).$$

Hence primitive prime divisors of $q^e - 1$ for various $e \leq d$ divide both the orders of $\mathrm{GL}(d, q)$ and of the other classical groups. We say that $g \in \mathrm{GL}(d, q)$ is a *ppd-element* if its order is divisible by some primitive prime divisor of $q^e - 1$ for some $e \in \{1, \ldots, d\}$.

6.1 Classical groups in natural representation

Much of the recent activity on algorithms for linear groups was stimulated by Neumann & Praeger [84], who presented a Monte Carlo algorithm to decide whether or not a subgroup of $\mathrm{GL}(d, q)$ contains $\mathrm{SL}(d, q)$.

Niemeyer & Praeger [88] answer the equivalent question for an arbitrary classical group. This they do by refining a classification by Guralnick *et al.* [51] of the subgroups of $\mathrm{GL}(d, q)$ which contain ppd-elements for $e > d/2$. The resulting Monte Carlo algorithms have complexity $O(\log\log d(\xi + d^\omega(\log q)^2))$, where ξ is the cost of selecting a random element and d^ω is the cost of matrix multiplication. For an excellent account, see [94]. Their implementation is available in Magma.

6.2 Black-box groups of Lie type

Babai *et al.* [8] present a black-box algorithm to name a group G of Lie type in known defining characteristic p. The algorithm selects a sample $\mathcal{L}$ of random elements in G, and determines the three largest integers $v_1 > v_2 > v_3$ such that at least one member of $\mathcal{L}$ has order divisible by a primitive prime divisor of $p^v - 1$ for $v = v_1, v_2$, or v_3. Usually $\{v_1, v_2, v_3\}$ determines $|G|$ and so names G. The algorithm of Altseimer & Borovik [1] distinguishes between $\mathrm{P\Omega}(2m + 1, q)$ and $\mathrm{PSp}(2m, q)$ for odd q. The central result of [8] is the following.

Theorem 6.1 *Given a black-box group G isomorphic to a simple group of Lie type of known characteristic, the standard name of G can be computed using a polynomial-time Monte Carlo algorithm.*

An implementation developed by Malle and O'Brien is distributed with GAP and Magma. It includes naming procedures for the other quasisimple groups: if the non-abelian composition factor is alternating or sporadic, then we identify it by considering the orders of random elements.

6.3 Determining the defining characteristic

Theorem 6.1 assumes that the defining characteristic of the input group of Lie type is *known*.

Problem 6.2 Let G be a group of Lie type in *unknown* defining characteristic r. Determine r.

Liebeck & O'Brien [76] present a Monte Carlo polynomial-time black-box algorithm which proceeds recursively through centralisers of involutions of G to find $\mathrm{SL}(2, F)$, where F is a field in characteristic r. It is now easy to read off the value of r.

Kantor & Seress [67] prove that the three largest element orders determine the characteristic of Lie-type simple groups of odd characteristic, and use this result to underpin an alternative algorithm.

The former is distributed in Magma, the latter in GAP.

7 Constructing an involution centraliser

Involution centralisers played a key role in the classification of finite simple groups. They were also used extensively in early computations with sporadic groups; see for example [77]. Borovik [15], Parker & Wilson [93] and Yalçınkaya [106] study them in the general context of black-box groups.

The centraliser of an involution in a black-box group having an order oracle can be constructed using an algorithm of Bray [18], who proves the following.

Theorem 7.1 *If x is an involution in a group H, and w is an arbitrary element of H, then $[x,w]$ either has odd order $2k+1$, in which case $w[x,w]^k$ commutes with x, or has even order $2k$, in which case both $[x,w]^k$ and $[x,w^{-1}]^k$ commute with x. If w is uniformly distributed among the elements of the group for which $[x,w]$ has odd order, then $w[x,w]^k$ is uniformly distributed among the elements of the centraliser of x.*

Thus if the odd order case occurs sufficiently often (with probability at least a positive rational function of the input size), then we can construct random elements of the involution centraliser in Monte Carlo polynomial time. In practice, we also use the output of the even-order case to obtain a generating set for the centraliser.

Parker & Wilson [93] prove the following.

Theorem 7.2 *There is an absolute positive constant c such that if H is a finite simple classical group of Lie rank r defined over a field of odd characteristic, and x is an involution in H, then $[x,h]$ has odd order for at least a proportion c/r of the elements h in H.*

For each class of involutions, they find a dihedral group of twice odd order generated by two involutions of this class, and show that a significant proportion of pairs of involutions in this class generate such a dihedral group.

For exceptional groups, they show that the analogous result is true for at least a positive proportion of elements h in H.

For each sporadic group we can calculate explicitly the proportion of $[x,h]$ which have odd order: for every class of involutions, this proportion is at least 17%.

The work of Liebeck & Shalev [75, Theorem] implies that, with arbitrarily high probability, a constant number of random elements generates the centraliser of an involution in a finite simple group.

Holmes *et al.* [56] establish the cost of constructing an involution centraliser:

Theorem 7.3 *Let H be a simple group of Lie type defined over a field of odd characteristic, having a black-box encoding of length n and equipped with an order oracle. Let ξ and ρ denote the cost of selecting a random element and of an order oracle respectively. The centraliser in H of an involution can be computed in time $O(\sqrt{n}(\xi+\rho)\log(1/\epsilon)+\mu n)$ with probability of success at least $1-\epsilon$, for positive ϵ.*

8 Constructive recognition

Assume that we wish to construct isomorphisms between the central quotient of a given quasisimple group H and a projective representation $G = \langle X \rangle$ of H. Recall from Section 3 that we do this by defining standard generators for H and constructing the corresponding standard generators of G as SLPs in X.

8.1 Black-box classical groups

Kantor & Seress [65] prove the following.

Theorem 8.1 *There is a Las Vegas algorithm which, when given as input a black-box perfect group G where $G/Z(G)$ is isomorphic to a classical simple group C of known characteristic, produces a constructive isomorphism $G/Z(G) \to C$.*

Recall that $g \in G$ is *p-singular* if its order is divisible by p. As Isaacs, Kantor & Spaltenstein [63] and Guralnick & Lübeck [52] show, a group of Lie type in defining characteristic p has a small proportion of p-singular elements.

Theorem 8.2 *If G is a group of Lie type defined over* GF(q)*, then $\frac{2}{5q} < \rho(G) < \frac{5}{q}$, where $\rho(G)$ denotes the proportion of p-singular elements in G.*

A necessary first step of the Kantor & Seress algorithm [65] is to find an element of order p: hence its running time has a factor of $q = p^f$ and so it is not polynomial in the size of the input.

Brooksbank & Kantor [22] identify that the obstruction to a polynomial-time algorithm for constructive recognition of the classical groups is $\mathrm{PSL}(2, q)$. Babai & Beals [7] formulate the problem explicitly as follows.

Problem 8.3 Find an element of order p in $\mathrm{PSL}(2, p^f)$ as a word in its defining generators in polynomial time.

Since $\rho(\mathrm{PSL}(2, q)) \leq 2/q$, a random search will involve $O(q)$ selections.

A consequence of the work of [71] is that the degree of a faithful projective representation of $\mathrm{SL}(2, q)$ in cross characteristic is polynomial in q rather than in $\log q$. Hence the critical instances of this problem are matrix representations of $\mathrm{SL}(2, q)$ in defining characteristic.

Conder & Leedham-Green [37] and Conder, Leedham-Green & O'Brien [38] present an algorithm which, subject to the existence of a discrete log oracle, constructively recognises $\mathrm{SL}(2, q)$ as a linear group in defining characteristic in time polynomial in the size of the input. The principal result is the following.

Theorem 8.4 *Let G be a subgroup of* $\mathrm{GL}(d, F)$ *for $d \geq 2$, where F is a finite field of the same characteristic as* GF(q)*; assume that G is isomorphic modulo scalars to* $\mathrm{PSL}(2, q)$*. Subject to a fixed number of calls to a discrete log oracle for* GF(q)*, there is a Las Vegas algorithm that constructs an epimorphism from G to* $\mathrm{PSL}(2, q)$ *at a cost of $O(d^5 \tau(d))$ field operations, where $\tau(d)$ denotes the number of divisors of d.*

Brooksbank [21, 24] and Brooksbank & Kantor [22, 26] have exploited this work to produce better constructive recognition algorithms for black-box classical groups. We summarise the outcome.

Theorem 8.5 *There is a Las Vegas algorithm which, when given as input a black-box G such that $C \cong G/Z(G)$ is $\mathrm{PSL}(d,q)$, $\mathrm{PSp}(2m,q)$, $\mathrm{PSU}(d,q)$, or $\mathrm{P}\Omega^{\epsilon}(d,q)$ for $\epsilon \in \{\pm, 0\}$, and a constructive recognition oracle for $\mathrm{SL}(2,q)$, outputs a constructive isomorphism $G/Z(G) \to C$. Its running time is a polynomial in the input length plus the time of polynomially many calls to the $\mathrm{SL}(2,q)$ oracle.*

A partial implementation of the algorithm of [65], developed by Brooksbank, Seress and others, is available in GAP and Magma. The algorithm of [38] is available in Magma.

8.2 Classical groups in their natural representation

Leedham-Green & O'Brien [73] developed constructive recognition algorithms for the classical groups in their natural representation, over fields of odd defining characteristic. A key component is the use of involution centralisers, whose structure in such groups is well-known; see, for example, [50, Table 4.5.1].

Let ξ denote an upper bound to the number of field operations needed to construct a random element of a group, and let $\chi(q)$ denote an upper bound to the number of field operations equivalent to a call to a discrete logarithm oracle for $\mathrm{GF}(q)$.

Leedham-Green & O'Brien [73] prove the following.

Theorem 8.6 *There is a Las Vegas algorithm that takes as input a subset X of bounded cardinality of $\mathrm{GL}(d,q)$, where X generates a classical group G, and returns standard generators for G as SLPs of length $\mathrm{O}(\log^3 d)$ in X. The algorithm has complexity $\mathrm{O}(d(\xi + d^3 \log d + d^2 \log d \log\log d \log q + \chi(q)))$ if G is neither of type $\mathbf{SO}^-$ or $\mathbf{\Omega}^-$. Otherwise the complexity is $\mathrm{O}(d(\xi + d^3 \log d + d^2 \log d \log\log d \log q + \chi(q)) + \chi(q^2))$.*

We describe the algorithm for $H = \mathrm{SL}(d,q)$. Let V denote the natural H-module with basis $\{e_1, \ldots, e_d\}$. We first define standard generators $\mathcal{S} = \{s, \delta, u, v\}$ for H. The matrices s, δ, u lie in a copy of $\mathrm{SL}(2,q)$; they fix each of $e_3, \ldots, e_d$ and induce the following action on $\langle e_1, e_2 \rangle$:

$$s \longmapsto \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \delta \longmapsto \begin{pmatrix} \omega & 0 \\ 0 & \omega^{-1} \end{pmatrix} \quad u \longmapsto \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

The $d-$cycle v maps $e_1 \longmapsto e_d \longmapsto -e_{d-1} \longmapsto -e_{d-2} \longmapsto \cdots \longmapsto -e_1$.

The input to the algorithm is $G = \langle X \rangle = \mathrm{SL}(d,q)$. Its task is construct $\mathcal{S}$ as SLPs in X.

A *strong involution* in $\mathrm{SL}(d,q)$ has its -1-eigenspace of dimension in the range $(d/3, 2d/3]$. If $t \in G$ is an involution with 1- and -1-eigenspace E_+ and E_- respectively, then $C_G(t)$ is $(\mathrm{GL}(E_+) \times \mathrm{GL}(E_-)) \cap \mathrm{SL}(d,q)$.

The steps of the recursive algorithm are:

1. Find and construct a strong involution t having its -1-eigenspace of dimension e. Rewrite G with respect to the new basis $E_- \cup E_+$.
2. Now construct $C_G(t)$. Construct the direct summands of its derived group to obtain $\mathrm{SL}(e,q)$ and $\mathrm{SL}(f,q)$ as *subgroups* of G where $f = d - e$.
3. Construct standard generators for $\mathrm{SL}(e,q)$ and $\mathrm{SL}(f,q)$.
4. Construct the centraliser C in G of the involution

$$\begin{pmatrix} I_{e-2} & 0 & 0 \\ 0 & -I_4 & 0 \\ 0 & 0 & I_{f-2} \end{pmatrix}.$$

5. Within C solve constructively for the matrix g

$$\begin{pmatrix} I_{e-2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{f-2} \end{pmatrix}.$$

6. Now use g to "glue" the e-cycle $v_e \in \mathrm{SL}(e,q)$ and f-cycle $v_f \in \mathrm{SL}(f,q)$ to obtain the d-cycle $v := v_e g v_f$.

The first step of the algorithm is to search for an element of $\mathrm{SL}(d,q)$ of even order that powers to a strong involution. Lübeck, Niemeyer & Praeger [80] prove the following.

Theorem 8.7 *For some absolute positive constant c, the proportion of $g \in \mathrm{SL}(d,q)$ such that a power of g is a strong involution is at least $c/\log d$.*

Observe that Step 3 is recursive, prompting invocations of the same procedure for $\mathrm{SL}(e,q)$ and $\mathrm{SL}(f,q)$. Since g is a strong involution, each group has degree less than $2d/3$; as shown in [73], the recursive calls do not affect the degree of complexity of the overall algorithm.

Recursion to smaller cases requires additional results about involutions which are not strong. We summarise the relevant results of [73].

Theorem 8.8 *For some absolute positive constant c, the proportion of $g \in \mathrm{SL}(d,q)$ such that a power of g is a "suitable" involution is at least c/d.*

The base cases for the recursion are $\mathrm{SL}(d,q)$ where $d \leq 4$. For $\mathrm{SL}(2,q)$ we use the algorithm of [38] to construct standard generators as SLPs in the input generators; for $\mathrm{SL}(3,q)$ we use the algorithm of [79]; for $\mathrm{SL}(4,q)$ we use the *involution-centraliser algorithm* of [56]. An implementation is available with Magma.

Black-box versions of these algorithms are being developed by Damien Burns. We are developing similar algorithms for classical groups in characteristic 2. As Theorem 8.2 indicates, the principal challenge is to construct a strong involution.

Brooksbank [23] also developed constructive recognition algorithms for classical groups in their natural representation: their effective cost is $O(d^5 \log^2 q)$, subject to calls to an $\mathrm{SL}(2,q)$ oracle.

8.3 Small degree matrix representations of $\mathbf{SL}(d, q)$

Let $\mathrm{SL}(d, q) \leq H \leq \mathrm{GL}(d, q)$ with $q = p^f$, where V is the natural H-module and V^* is its dual module. Define the Frobenius map $\delta : \mathrm{GL}(d, q) \to \mathrm{GL}(d, q)$ by $(a_{i,j})^\delta = (a_{i,j}^p)$ for $(a_{i,j}) \in \mathrm{GL}(d, q)$.

Let H act on an irreducible $\mathrm{GF}(q)$-module W of dimension at most d^2. Consider $V^* \otimes V$ with basis $\{e_i \otimes e_j \mid 1 \leq i, j \leq d\}$ and let

$$w := \sum_{i=1}^{d} e_i \otimes e_i, \qquad U := \left\{ \sum_{i,j} \alpha_{i,j} e_i \otimes e_j \mid \sum_{i=1}^{d} \alpha_{i,i} = 0 \right\}, \qquad W_1 := U \cap \langle w \rangle.$$

The *adjoint module* of V is $W := U/W_1$. If $d \equiv 0 \bmod p$ then W has dimension $d^2 - 2$, otherwise $d^2 - 1$. The remaining irreducible representations of dimension at most d^2 are $V \otimes V^{\delta^e}$ and $V^* \otimes V^{\delta^e}$ where $0 < e < f$. For a discussion, see [74].

Magaard, O'Brien & Seress [82] describe algorithms which, given as input W, construct a d-dimensional projective representation of H. Their principal result is the following.

Theorem 8.9 *Let $d \geq 2$ and let $q = p^f$ be a prime power. Let* $\mathrm{SL}(d, q) \leq H \leq \mathrm{GL}(d, q)$ *where H has natural module V. Let $G = \langle X \rangle$ be a representation of H acting irreducibly on a* $\mathrm{GF}(q)$*-vector space W of dimension $n \leq d^2$.*

Given as input G, the value of d, and error probability $\epsilon > 0$, there is a Las Vegas algorithm that, with probability at least $1 - \epsilon$, constructs the projective action of G on V.

The algorithms are specific to each representation type and in all but one case run in polynomial time. A common feature is to search randomly in G for (a power of) a Singer cycle s, and identify a basis for W consisting of eigenvectors for the action of s on $W \otimes \mathrm{GF}(q^d)$. Implementations are available in Magma.

Ryba [95] presents a polynomial-time Las Vegas algorithm that, given as input an absolutely irreducible representation in odd defining characteristic of a finite Chevalley group, constructs its action on the adjoint module. A combination of his algorithm and that of [82] can be used to construct the natural projective action of $\mathrm{SL}(d, q)$.

8.4 Alternating groups

Beals *et al.* [13] prove the following.

Theorem 8.10 *Black-box groups isomorphic to A_n or S_n with known value of n can be recognised constructively in $O(\xi n + \mu |X| n \log n)$ time, where ξ is the time to construct a random element, μ is the time for a group operation, and X is the input generating set for the group.*

Beals *et al.* [14] present a more efficient algorithm for the deleted permutation module viewed as a linear group. Implementations are available in GAP and Magma.

An alternative black-box algorithm, developed by Bratus & Pak [17], was further refined and implemented in Magma by Derek Holt.

8.5 Exceptional groups

Algorithms to recognise constructively matrix representations of the Suzuki, large and small Ree groups were developed by Bäärnhielm [11, 12]. Implementations are available in Magma.

Kantor & Magaard [68] present black-box Las Vegas algorithms to recognise constructively the exceptional simple groups of Lie type and rank at least 2, other than ${}^2F_4(q)$, defined over a field of known size.

8.6 Sporadic groups

Wilson [103] introduced the concept of *standard generators* for the sporadic groups. He, Bray and others provide black-box algorithms for their construction. For further details, see the Atlas web site [104].

9 The constructive membership problem

Recall our definition of the *constructive membership problem* for a quasisimple group $G = \langle X \rangle$: if $g \in G$ then write g as an SLP in X.

Assume we have solved the constructive recognition problem for G: namely, we have constructed standard generators $\bar{S}$ for G as SLPs in X. If we can express $g \in G$ as an SLP in $\bar{\mathcal{S}}$, then we rewrite the SLP in $\bar{S}$ for g to obtain one in X. Hence we focus on the task of writing $g \in G$ as an SLP in $\bar{\mathcal{S}}$.

9.1 Classical groups

Costi [41] developed algorithms to write an element of a classical group $H \leq \mathrm{GL}(d, q)$ in its natural representation as an SLP in the standard generators of H. These algorithms are natural (but quite technical) extensions of row and column operations, and have complexity $O(d^3 \log q)$ field operations.

Consider now the case where G is a defining characteristic (projective) irreducible representation of H. Again Costi [41] developed algorithms to solve the membership problem for G; these have complexity $O(d^4n^3 \log^3 q + d^2n^4 \log q)$ where n is the degree of G. Implementations of both are available in Magma.

Key components are two polynomial-time algorithms for unipotent groups:

1. Subspace-Stabiliser algorithm
   Input: a unipotent matrix group S and a subspace U of its underlying vector space.
   Output: a canonical element $\overline{U}$ of the orbit of U under S; and $s \in S$ such that $U^s = \overline{U}$; and generators for the stabiliser of U in S.
2. An algorithm to solve the constructive membership problem in a unipotent matrix group.

We summarise Costi's algorithm when $H = \mathrm{SL}(d, q)$. Let $G \leq \mathrm{GL}(n, F)$ be a defining characteristic projective irreducible representation of H. Let G act on the underlying vector space V, and let $\phi : H/Z(H) \to G$ be a constructive isomorphism. Assume that we wish to write $g \in G$ as an SLP in $\bar{S}$.

1. Let K be the maximal parabolic subgroup of H that fixes the space spanned by the first element of the standard basis for the underlying space of H. Namely, elements of K have shape
$$\begin{pmatrix} \det^{-1} & 0 & 0 & 0 \\ \star & & & \\ \vdots & & \mathrm{GL}(d-1,q) & \\ \star & & & \end{pmatrix}$$
where each $\star$ is an arbitrary element of $\mathrm{GF}(q)$. Since $K\phi$ is a p-local subgroup in defining characteristic p, it stabilises a proper $K\phi$-submodule U of V.
2. Consider the elementary abelian subgroup E of H generated by elements
$$\begin{pmatrix} 1 & \star & \dots & \star \\ 0 & & & \\ \vdots & & I_{d-1} & \\ 0 & & & \end{pmatrix}.$$
Use Subspace-Stabiliser to construct $x \in E\phi$ as an SLP that maps U^g to U. Hence $U^{gx} = U$ and so the preimage of gx is in K. Thus we have "killed" the first row of the preimage of gx.
3. Dualise to kill first column, obtaining $g_1 := \begin{pmatrix} \alpha & 0 \\ 0 & A \end{pmatrix}$.
4. Observe that $t\phi := g_1^{-1} \cdot T_{1,j}^{\phi} \cdot g_1 \in E\phi$ where $T_{1,j}$ is a transvection with non-zero entry in $(1, j)$ position. Use the constructive unipotent membership test for $t\phi$ in $E\phi$ to obtain its preimage $t \in E$.
5. Read off from t (a scalar multiple of) the j-th row of the preimage in $\mathrm{SL}(d, q)$ of g_1.
6. We have now reduced the constructive membership problem to the *natural representation* in rank $d - 1$; use the corresponding natural representation algorithm to solve this simpler problem.

The two "unipotent" components of this algorithm depend critically on the assumption that G is a matrix representation of H in defining characteristic.

In ongoing work, Murray, Praeger and Schneider are developing black-box algorithms to solve the problem for classical groups on the standard generators defined in [73]. The basic structure of their algorithms is similar to Costi's, but the problems addressed using the unipotent components must now be solved in a black-box group.

The black-box algorithms of [22, 24, 65] solve the same task, again using a similar approach, on different and significantly larger generating sets. An implementation of [65] is available in Magma for $\mathrm{SL}(d, q)$, as are implementations by Brooksbank of some small rank cases from [22, 24].

9.2 Other algorithms

The *centraliser-of-involution* algorithm [56] reduces the problem of testing whether an arbitrary element g of a black-box group G lies in a fixed subgroup H to instances of the same problem for $C_H(t)$ for (at most) three involutions $t \in H$. The reduction occurs in polynomial time. The algorithm is constructive: if $g \in H$ then it returns an SLP for g in the generators of H. Our implementation in Magma uses CompositionTree, described in Section 11, to solve the problem for each centraliser.

The Schreier-Sims algorithm, and its variations, solves the constructive membership problem for a permutation or matrix group G. First introduced by Sims [99], it constructs a *base* for G which determines a stabiliser chain in G. For a basic outline, see [91]; for an analysis, see [97, p. 64].

9.3 Sporadic groups

For each sporadic group, O'Brien and Wilson developed a black-box algorithm to construct a chain of its subgroups; as described in [91], they exploit the reducibility of members of this chain to obtain a "good" base for the Schreier-Sims algorithm.

With this assistance, either the Schreier-Sims algorithm or the algorithm of [56] solves the constructive membership problem for all Atlas representations [104] of most sporadic groups; the exceptions are the Baby Monster and the Monster where strategies developed by Wilson and others are employed [105]. Implementations are available in Magma.

10 Short presentations

Standard generators may be used to define a surjection from a supplied group $G = \langle X \rangle$ to a simple group H. Is this surjection an isomorphism? If not, what is its kernel? If we have a presentation $\mathcal{P}$ for H on standard generators, then we can evaluate relations of $\mathcal{P}$ in standard generators of G and so obtain normal generators for the kernel of the map from G to H. This motivates our interest in presentations for groups of Lie type on particular generating sets.

Babai & Szemerédi [6] define the *length* of a presentation to be the number of symbols required to write down the presentation. Each generator is a single symbol, and a relator is a string of symbols, where exponents are written in binary. The length of a presentation is the number of generators plus the sum of the lengths of the relators. They also formulated the *Short Presentation Conjecture*: there exists a constant c such that every finite simple group G has a presentation of length $O(\log^c |G|)$.

Perhaps the best known presentations for the finite symmetric groups are those of Moore [83]; see also [42, 6.22]. There, the symmetric group S_n of degree n is presented in terms of the transpositions $t_k = (k, k{+}1)$ for $1 \leq k < n$, which generate S_n and satisfy the defining relations $t_k^2 = 1$ for $1 \leq k < n$, and $(t_{k-1}t_k)^3 = 1$ for $1 < k < n$, and $(t_j t_k)^2 = 1$ for $1 \leq j < k-1 < n-1$. For $n > 1$ the number of these relations is $n(n-1)/2$, and since each relator has bounded length, the presentation length is $O(n^2)$.

If, for example, S_n acts on the deleted permutation module, then the cost of evaluating these relations is $O(n^5)$: this is *more expensive* than constructive recognition of this representation (which can be performed using the algorithm of [14]).

Hence a goal of both theoretical and practical interest is to obtain "short" presentations for the finite simple groups on particular generating sets.

A key step is to obtain short presentations for A_n and S_n. Independently in 2006, Bray *et al.* [19] and Guralnick *et al.* [53] proved the following.

Theorem 10.1 *A_n and S_n have presentations with a bounded number of generators and relations, and length $O(\log n)$.*

This is best possible since it requires $\log n$ bits to represent n; the previous best result was a modification of the Moore presentation having length $O(n \log n)$.

Guralnick *et al.* [54] prove that A_n has a presentation on 3 generators, 4 relations, and length $O(\log n)$. Bray *et al.* [19] prove that S_n has a presentation of length $O(n^2)$ on generators $(1, 2)$ and $(1, 2, \ldots, n)$, and at most 123 relations.

Problem 10.2 Is there a shorter presentation for S_n defined on generators $(1, 2)$ and $(1, 2, \ldots, n)$ with a uniformly bounded number of relations?

In a major extension, Guralnick *et al.* [53] prove the following.

Theorem 10.3 *Every non-abelian finite simple group of rank n over* $\mathrm{GF}(q)$*, with the possible exception of the Ree groups ${}^2G_2(q)$, has a presentation with a bounded number of generators and relations and total length $O(\log n + \log q)$.*

Again this is best possible. It exploits the following results.

- Campbell, Robertson & Williams [27]: $\mathrm{PSL}(2, q)$ has a presentation on (at most) 3 generators and a bounded number of relations.
- Hulpke & Seress [62]: $\mathrm{PSU}(3, q)$ has a presentation of length $O(\log^2 q)$.

In ongoing work, Leedham-Green and O'Brien are constructing explicit short presentations on our standard generators for the classical groups.

11 The composition tree

In ongoing work, Bäärnhielm, Leedham-Green and O'Brien are developing the concept of a *composition tree*, an integrated framework to realise and exploit the geometric approach. An early design was presented in [72]; our latest is implemented in Magma. A variation developed by Neunhöffer & Seress [85] is available in GAP.

A composition series for a group G can be viewed as a labelled rooted binary tree. A node corresponds to a section H of G, the root node to G. If a node is not a leaf, then it has a left child corresponding to a proper normal subgroup K of H and a right child I isomorphic to H/K.

The right child is an image under a homomorphism. Usually these arise naturally from the Aschbacher category of the group, but we exploit additional homomorphisms, including the determinant map and some applicable to unipotent and soluble groups. The left child of a node is the kernel of the chosen homomorphism.

The tree is constructed in *right depth-first order*. Namely, we process the node associated with H: if H is not a leaf, construct recursively the subtree rooted at its right child I, then the subtree rooted at its left child K.

A *leaf* of the composition tree is usually a composition factor of G: however, a non-abelian leaf need only be simple modulo scalars, and cyclic factors are not necessarily of prime order.

Assume $\phi : H \to I$ where $K = \ker \phi$. It is easy to construct I, since it is the image of H under a homomorphism ϕ. Sometimes it is easy theoretically to construct generating sets for $\ker \phi$, for example if H is in Aschbacher category C3. Otherwise, we first construct a normal generating set for K by evaluating in the generators of H the relators in a presentation for I and then take its normal closure using the algorithm of [97, Chapter 2].

To obtain a presentation for a node, we need only presentations for its associated kernel and image; an algorithm for this task is described in [72]. Hence inductively we require presentations only for the leaves. If we know a presentation on standard generators for the leaf, then this is used; otherwise we use the algorithm of [28] to construct such a presentation.

We solve the constructive membership problem *directly* for a leaf using the techniques of Section 9. If we solve the membership problem for the children of a node, then we readily solve the problem for the node, and so recursively obtain a solution for the root node.

Assume that $G = \langle X \rangle \leq \mathrm{GL}(d, q)$ is input to CompositionTree. Some of the algorithms used in constructing a composition tree for G are Monte Carlo. To verify the resulting construction, we write down a presentation for the group defined by the tree and show that G satisfies its relations.

The output of CompositionTree is:

- A composition series $1 = G_0 \lhd G_1 \lhd G_2 \lhd \cdots \lhd G_m = G$.
- A representation $S_k = \langle X_k \rangle$ of G_k/G_{k-1}.
- Effective maps $\tau_k : G_k \to S_k$ and $\phi_k : S_k \to G_k$. The map τ_k is an epimorphism with kernel G_{k-1}; if $g \in S_k$, then $\phi_k(g)$ is an element of G_k satisfying $\tau_k \phi_k(g) = g$.
- A map to write $g \in G$ as an SLP in X.

12 Applications

Over the past decade, Cannon, Holt and their collaborators have pioneered the development of certain practical algorithms to answer structural questions about finite groups. These exploit the characteristic series $\mathcal{C}$ of a finite group G

$$1 \leq O_\infty(G) \leq S^*(G) \leq P(G) \leq G$$

and a refined series for the soluble radical $O_\infty(G)$

$$1 = N_0 \lhd N_1 \lhd \cdots \lhd N_r = O_\infty(G) \lhd G$$

where $N_i \trianglelefteq G$ and N_i/N_{i-1} is elementary abelian. Since $G/O_\infty(G)$ has a trivial Fitting subgroup, we call it a *TF-group*.

The resulting framework is sometimes called the *Trivial Fitting model of computation.* It suggests the following paradigm to solve a problem.

Solve the problem first in G/N_r, and then, successively, solve it in G/N_i, for $i = r-1, \ldots, 0$.

Since $H := G/O_\infty(G)$ has the structure outlined in Section 4, the problem may have an "easy" solution in H. In particular, we can usually readily reduce the problem for H to a question about almost simple groups. Increasingly, explicit solutions are available for such groups.

Algorithms which use this paradigm include:

- Determine conjugacy classes of elements (see [29]).
- Determine conjugacy classes of subgroups (see [30]).
- Determine the automorphism group (see [31]).
- Determine maximal subgroups (see [32] and [45]).

While these algorithms are effectively black-box, their current Magma implementations use the Schreier-Sims algorithm for associated computations and so are limited in range. Recently, Holt refined the output of CompositionTree for a group to obtain a chief series exhibiting $\mathcal{C}$. In ongoing work Holt, Leedham-Green, O'Brien and Roney-Dougal are exploring how to exploit CompositionTree and this chief series to provide basic infrastructure for such algorithms.

12.1 Exploiting data for classical groups

We mention two examples where available data for classical groups can be exploited.

In 1963, Wall [102] described theoretically the conjugacy classes and centralisers of elements of classical groups. In ongoing work, Haller and Murray exploit this description and provide algorithms which construct these explicitly in the natural representation of groups contained in the conformal group (the group preserving the corresponding form up to scalars). The constructive isomorphisms obtained from constructive recognition allow us to map the class representatives and centralisers from the natural copy to an arbitrary projective representation.

Kleidman & Liebeck [70] describe the maximal subgroups in the Aschbacher categories C1-C8 of classical groups of degree $d \geq 13$. Holt & Roney-Dougal [60, 61] construct generating sets in the natural representation for these subgroups; in ongoing work with Bray they classify all maximals for $d \leq 12$. Again the constructive isomorphism is used to construct their images in an arbitrary projective representation.

12.2 Constructing the automorphism group of a finite group

As one illustration of the paradigm, we sketch the algorithm of Cannon & Holt [31] to compute the automorphism group of an arbitrary finite group G. (Special purpose algorithms exist for soluble groups.)

Recall that $H := G/O_\infty(G)$ permutes the direct factors of its socle S by conjugation. We embed H in the direct product $D := \prod_i \mathrm{Aut}(T_i) \wr \mathrm{Sym}(d_i)$, where T_i occurs d_i times as socle factor of S. Now $\mathrm{Aut}(H)$ is the normaliser of the image of H in D. Hence we effectively reduce the computation for the TF-group H to the finite simple case.

We now lift results through elementary abelian layers, computing $\mathrm{Aut}(G/N_i)$ successively. Suppose $N \leq M \leq G$, where both M and N are characteristic in G, and M/N is elementary abelian of order p^d.

Assume $\mathrm{Aut}(G/M)$ is known. All automorphisms of G fix both M and N. Observe that $\mathrm{Aut}(G/N)$ has normal subgroups $C \leq B$ where B induces the identity on G/M, and C induces the identity on both G/M and M/N. A key observation is that M/N is a $\mathrm{GF}(p)(G/M)$-module.

- Elements of C correspond to derivations from G/M to M/N and are obtained by solving systems of equations over $\mathrm{GF}(p)$.
- Elements of B/C correspond to module automorphisms of M/N. We can usually choose M and N to ensure that both this and the previous calculation are "easy".
- The remaining – and hardest – task is to determine the subgroup S of $\mathrm{Aut}(G/M)$ which lifts to G/N. Observe that $S \leq T$, the subgroup of $\mathrm{Aut}(G/M)$ whose elements preserve the (module) isomorphism type of M/N. Usually T can be computed readily. If G/N is a split extension of M/N by G/M, then all elements of T lift. Otherwise, we must test each element of T to decide whether it lifts to G/N.

References

[1] Christine Altseimer and Alexandre V. Borovik. Probabilistic recognition of orthogonal and symplectic groups. In *Groups and Computation, III (Columbus, OH, 1999)*, volume 8 of *Ohio State Univ. Math. Res. Inst. Publ.*, pages 1–20. De Gruyter, Berlin, 2001.

[2] Sophie Ambrose. Matrix Groups: Theory, Algorithms and Applications. PhD thesis, University of Western Australia, 2006.

[3] M. Aschbacher. On the maximal subgroups of the finite classical groups. *Invent. Math.* **76**, 469–514, 1984.

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

[5] László Babai. Randomization in group algorithms: conceptual questions. In *Groups and Computation, II (New Brunswick, NJ, 1995)*, 1–17, Amer. Math. Soc., Providence, RI, 1–17, 1997.

[6] László Babai and Endre Szemerédi. On the complexity of matrix group problems, I. In *Proc. 25th IEEE Sympos. Foundations Comp. Sci.*, pages 229–240, 1984.

[7] László Babai and Robert Beals. A polynomial-time theory of black box groups. I. In *Groups St. Andrews 1997 in Bath, I*, volume 260 of *London Math. Soc. Lecture Note Ser.*, pages 30–64, 1999. Cambridge Univ. Press.

[8] László Babai, William M. Kantor, Péter P. Pálfy and Ákos Seress. Black-box recognition of finite simple groups of Lie type by statistics of element orders. *J. Group Theory* **5**, 383–401, 2002.

[9] László Babai, Péter P. Pálfy and Jan Saxl. On the number of p-regular elements in finite simple groups. *LMS J. Comput. Math.* **12**, 82–119, 2009.

[10] László Babai, Robert Beals and Ákos Seress. Polynomial-time Theory of Matrix Groups. In Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, pages 55–64, 2009.

[11] Henrik Bäärnhielm. Algorithmic problems in twisted groups of Lie type. PhD thesis, Queen Mary, University of London, 2006.

[12] Henrik Bäärnhielm. Recognising the Suzuki groups in their natural representations. J. Algebra **300**, 171–198, 2006.

[13] Robert Beals, Charles R. Leedham-Green, Alice C. Niemeyer, Cheryl E. Praeger and Ákos Seress. A black-box group algorithm for recognizing finite symmetric and alternating groups. I. *Trans. Amer. Math. Soc.* **355**, 2097–2113, 2003.

[14] Robert Beals, Charles R. Leedham-Green, Alice C. Niemeyer, Cheryl E. Praeger and Ákos Seress. Constructive recognition of finite alternating and symmetric groups acting as matrix groups on their natural permutation modules. *J. Algebra* **292**, 4–46, 2005.

[15] Alexandre V. Borovik. Centralisers of involutions in black box groups. In *Computational and statistical group theory (Las Vegas, NV/Hoboken, NJ, 2001)*, 7–20, *Contemp. Math.*, 298, Amer. Math. Soc., Providence, RI, 2002.

[16] Wieb Bosma, John Cannon and Catherine Playoust. The Magma algebra system I: The user language. *J. Symbolic Comput.* **24**, 235–265, 1997.

[17] Sergey Bratus and Igor Pak. Fast constructive recognition of a black box group isomorphic to S_n or A_n using Goldbach's conjecture. *J. Symbolic Comput.* **29**, 33–57, 2000.

[18] John N. Bray. An improved method for generating the centralizer of an involution. *Arch. Math. (Basel)* **74**, 241–245, 2000.

[19] John Bray, M.D.E. Conder, C.R. Leedham-Green and E.A. O'Brien. Short presentations for alternating and symmetric groups. To appear *Trans. Amer. Math. Soc.* 2010.

[20] John Brillhart, D.H. Lehmer, J.L. Selfridge, Bryant Tuckerman, and S.S. Wagstaff, Jr. *Factorizations of $b^n \pm 1$*, volume 22 of *Contemporary Mathematics*. American Mathematical Society, Providence, RI, second edition, 1988. `www.cerias.purdue.edu/homes/ssw/cun/index.html`.

[21] Peter A. Brooksbank. A constructive recognition algorithm for the matrix group $\Omega(d,q)$. In *Groups and Computation, III (Columbus, OH, 1999)*, volume 8 of *Ohio State Univ. Math. Res. Inst. Publ.*, pages 79–93. De Gruyter, Berlin, 2001.

[22] Peter A. Brooksbank and William M. Kantor. On constructive recognition of a black box $\mathrm{PSL}(d,q)$. In *Groups and Computation, III (Columbus, OH, 1999)*, volume 8 of *Ohio State Univ. Math. Res. Inst. Publ.*, pages 95–111. De Gruyter, Berlin, 2001.

[23] Peter A. Brooksbank. Constructive recognition of classical groups in their natural representation. *J. Symbolic Comput.* **35**, 195–239, 2003.

[24] Peter A. Brooksbank. Fast constructive recognition of black-box unitary groups. *LMS J. Comput. Math.* **6**, 162–197 (electronic), 2003.

[25] Peter Brooksbank, Alice C. Niemeyer and Ákos Seress. A reduction algorithm for matrix groups with an extraspecial normal subgroup. *Finite Geometries, Groups and Computation*, (Colorado), pp. 1–16. De Gruyter, Berlin, 2006.

[26] Peter A. Brooksbank and William M. Kantor. Fast constructive recognition of black box orthogonal groups. *J. Algebra* **300**, 256–288, 2006.

[27] C.M. Campbell, E.F. Robertson and P.D. Williams. On Presentations of $\mathrm{PSL}(2,p^n)$. *J. Austral. Math. Soc.* **48**, 333–346, 1990.

[28] John J. Cannon. Construction of defining relators for finite groups. *Discrete Math.* **5**, 105–129, 1973.

[29] John Cannon and Bernd Souvignier. On the computation of conjugacy classes in permutation groups. In *Proceedings of International Symposium on Symbolic and Algebraic Computation, Hawaii, 1997*, pages 392–399. Association for Computing Machinery, 1997.

[30] John J. Cannon, Bruce C. Cox and Derek F. Holt. Computing the subgroups of a permutation group. *J. Symbolic Comput.* **31**, 149–161, 2001.

[31] John J. Cannon and Derek F. Holt. Automorphism group computation and isomorphism testing in finite groups. *J. Symbolic Comput.* **35**, 241–267, 2003.

[32] John J. Cannon and Derek F. Holt. Computing maximal subgroups of finite groups. *J. Symbolic Comput.* **37**, 589–609, 2004.

[33] Jon F. Carlson, Max Neunhöffer and Colva M. Roney-Dougal. A polynomial-time reduction algorithm for groups of semilinear or subfield class. *J. Algebra* **322**, 613–617, 2009.

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

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

[36] A.H. Clifford. Representations induced in an invariant subgroup. *Ann. of Math.* **38**, 533–550, 1937.

[37] Marston Conder and Charles R. Leedham-Green. Fast recognition of classical groups over large fields. In *Groups and Computation, III (Columbus, OH, 1999)*, volume 8 of *Ohio State Univ. Math. Res. Inst. Publ.*, pages 113–121. De Gruyter, Berlin, 2001.

[38] M.D.E. Conder, C.R. Leedham-Green and E.A. O'Brien. Constructive recognition of $\mathrm{PSL}(2, q)$. *Trans. Amer. Math. Soc.* **358**, 1203-1221, 2006.

[39] Gene Cooperman. Towards a practical, theoretically sound algorithm for random generation in finite groups. Posted on arXiv:math, May 2002.

[40] Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *J. Symbolic Comput.* **9**, 251–280, 1990.

[41] Elliot Costi. Constructive membership testing in classical groups. PhD thesis, Queen Mary, University of London, 2009.

[42] H.S.M. Coxeter and W.O.J. Moser. *Generators and Relations for Discrete Groups*, 4th ed. Springer-Verlag (Berlin), 1980, ix+169 pp.

[43] A.S. Detinko, B. Eick and D.L. Flannery. Computing with matrix groups over infinite fields. These Proceedings.

[44] John D. Dixon. Generating random elements in finite groups. Electron. J. Combin. **15** (2008), no. 1, Research Paper 94, 13 pp.

[45] Bettina Eick and Alexander Hulpke. Computing the maximal subgroups of a permutation group. I. In *Groups and Computation, III (Columbus, OH, 1999)*, volume 8 of *Ohio State Univ. Math. Res. Inst. Publ.*, pages 155–168. De Gruyter, Berlin, 2001.

[46] The GAP Group. GAP – Groups, Algorithms, and Programming, Version 4.4.12; 2008. `www.gap-system.org`.

[47] Mark Giesbrecht. Nearly optimal algorithms for canonical matrix forms. PhD thesis, University of Toronto, 1993.

[48] S.P. Glasby, C.R. Leedham-Green and E.A. O'Brien. Writing projective representations over subfields. *J. Algebra* **295**, 51–61, 2006.

[49] S.P. Glasby and Cheryl E. Praeger. Towards an efficient Meat-axe algorithm using f-cyclic matrices: The density of uncyclic matrices in $\mathrm{M}(n, q)$. *J. Algebra* **322**, 766–790, 2009.

[50] Daniel Gorenstein, Richard Lyons and Ronald Solomon. The classification of the finite simple groups. Number 3. American Mathematical Society, Providence, RI, 1998.

[51] Robert Guralnick, Tim Penttila, Cheryl E. Praeger and Jan Saxl. Linear groups with orders having certain large prime divisors. *Proc. London Math. Soc.* **78**, 167–214, 1999.

[52] R.M. Guralnick and F. Lübeck. On p-singular elements in Chevalley groups in characteristic p. In *Groups and Computation, III (Columbus, OH, 1999)*, volume 8 of *Ohio State Univ. Math. Res. Inst. Publ.*, pages 169–182, De Gruyter, Berlin, 2001.

[53] R.M. Guralnick, W.M. Kantor, M. Kassabov and A. Lubotzky. Presentations of finite simple groups: a quantitative approach. *J. Amer. Math. Soc.* **21**, 711–774, 2008.

[54] R.M. Guralnick, W.M. Kantor, M. Kassabov and A. Lubotzky. Presentations of finite simple groups: a computational approach. To appear *J. European Math. Soc.*, 2010.

[55] G. Hiss and G. Malle. Low-dimensional representations of quasi-simple groups. *LMS J. Comput. Math.*, 4:22–63, 2001. Also: Corrigenda *LMS J. Comput. Math.* **5**, 95–126, 2002.

[56] P.E. Holmes, S.A. Linton, E.A. O'Brien, A.J.E. Ryba and R.A. Wilson. Constructive membership in black-box groups. *J. Group Theory* **11**, 747–763, 2008.

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

[58] 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**, 818–838, 1996.

[59] Derek F. Holt, Bettina Eick and Eamonn A. O'Brien. *Handbook of computational group theory.* Chapman and Hall/CRC, London, 2005.

[60] Derek F. Holt and Colva M. Roney-Dougal. Constructing maximal subgroups of classical groups. *LMS J. Comput. Math.* **8**, 46–79, 2005.

[61] Derek F. Holt and Colva M. Roney-Dougal. Constructing maximal subgroups of orthogonal groups. To appear *LMS J. Comput. Math.* 2010.

[62] Alexander Hulpke and Ákos Seress. Short presentations for three-dimensional unitary groups. *J. Algebra* **245**, 719–729, 2001.

[63] I.M. Isaacs, W.M. Kantor and N. Spaltenstein. On the probability that a group element is p-singular. *J. Algebra* **176**, 139–181, 1995.

[64] Gábor Ivanyos and Klaus Lux. Treating the exceptional cases of the MeatAxe. *Experiment. Math.* **9**, 373–381, 2000.

[65] William M. Kantor and Ákos Seress. Black box classical groups. *Mem. Amer. Math. Soc.*, **149** (708):viii+168, 2001.

[66] William M. Kantor and Ákos Seress. Computing with matrix groups. In *Groups, Combinatorics & Geometry (Durham, 2001)*, 123–137, World Sci. Publishing, River Edge, NJ, 2003.

[67] William M. Kantor and Ákos Seress. Large element orders and the characteristic of Lie-type simple groups. *J. Algebra* **322**, 802–832, 2009.

[68] William M. Kantor and Kay Magaard. Black box exceptional groups of Lie type. Preprint 2009.

[69] W. Keller-Gehrig. Fast algorithms for the characteristic polynomial. *Theoret. Comput. Sci.* **36**, 309–317, 1985.

[70] Peter Kleidman and Martin Liebeck. The subgroup structure of the finite classical groups. London Mathematical Society Lecture Note Series, **129**. Cambridge University Press, Cambridge, 1990.

[71] Vicente Landazuri and Gary M. Seitz. On the minimal degrees of projective representations of the finite Chevalley groups. *J. Algebra* **32**, 418–443, 1974.

[72] C.R. Leedham-Green. The computational matrix group project. In *Groups and Computation, III (Columbus, OH, 1999)*, 229–248. De Gruyter, Berlin, 2001.

[73] C.R. Leedham-Green and E.A. O'Brien. Constructive recognition of classical groups in odd characteristic. *J. Algebra* **322**, 833–881, 2009.

[74] Martin W. Liebeck. On the orders of maximal subgroups of the finite classical groups. *Proc. London Math. Soc.* (3) **50**, 426–446, 1985.

[75] Martin W. Liebeck and Aner Shalev. The probability of generating a finite simple group. *Geom. Ded.* **56**, 103–113, 1995.

[76] Martin W. Liebeck and E.A. O'Brien. Finding the characteristic of a group of Lie type. *J. Lond. Math. Soc.* **75**, 741–754, 2007.

[77] S.A. Linton. The art and science of computing in large groups. *Computational Algebra and Number Theory* (Sydney, 1992), pp. 91–109, 1995. Kluwer Academic Publishers, Dordrecht.

[78] F. Lübeck. Small degree representations of finite Chevalley groups in defining characteristic. *LMS J. Comput. Math.* **4**, 135–169, (electronic), 2001.

[79] F. Lübeck, K. Magaard and E.A. O'Brien. Constructive recognition of $\mathrm{SL}_3(q)$. *J. Algebra* **316**, 619–633, 2007.

[80] Frank Lübeck, Alice C. Niemeyer and Cheryl E. Praeger. Finding involutions in finite Lie type groups of odd characteristic. *J. Algebra* **321**, 3397-3417, 2009.

[81] Eugene M. Luks. Computing in solvable matrix groups. In *Proc. 33rd IEEE Sympos. Foundations Comp. Sci.*, 111–120, 1992.

[82] Kay Magaard, E.A. O'Brien and Ákos Seress. Recognition of small dimensional representations of general linear groups. *J. Aust. Math. Soc.* **85**, 229–250, 2008.

[83] E.H. Moore. Concerning the abstract groups of order $k!$ and $\frac{1}{2}\,k!$. *Proc. London Math. Soc.* **28**, 357–366, 1897.

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

[85] Max Neunhöffer and Ákos Seress. A data structure for a uniform approach to computations with finite groups, ISSAC 2006, ACM, New York, 2006, pp. 254–261.

[86] Max Neunhöffer. Constructive Recognition of Finite Groups. Habilitationsschrift, RWTH Aachen, 2009.

[87] Max Neunhöffer and Cheryl E. Praeger. Computing minimal polynomials of matrices. LMS JCM **11**, 252-279, 2008.

[88] Alice C. Niemeyer and Cheryl E. Praeger. A recognition algorithm for classical groups over finite fields. *Proc. London Math. Soc.*, 77:117–169, 1998.

[89] Alice C. Niemeyer. Constructive recognition of normalisers of small extra-special matrix groups. *Internat. J. Algebra Comput.*, **15**, 367–394, 2005.

[90] E.A. O'Brien and M.R. Vaughan-Lee. The 2-generator restricted Burnside group of exponent 7. *Internat. J. Algebra Comput.*, **12**, 575–592, 2002.

[91] E.A. O'Brien. Towards effective algorithms for linear groups. *Finite Geometries, Groups and Computation*, (Colorado), pp. 163-190. De Gruyter, Berlin, 2006.

[92] Igor Pak. The product replacement algorithm is polynomial. In *41st Annual Symposium on Foundations of Computer Science (Redondo Beach, CA, 2000)*, 476–485, IEEE Comput. Soc. Press, Los Alamitos, CA, 2000.

[93] Christopher W. Parker and Robert A. Wilson. Recognising simplicity of black-box groups. To appear *J. Algebra*, 2010.

[94] Cheryl E. Praeger. Primitive prime divisor elements in finite classical groups. In *Groups St. Andrews 1997 in Bath, II*, 605–623, Cambridge Univ. Press, 1999.

[95] Alexander J.E. Ryba. Identification of matrix generators of a Chevalley group. *J. Algebra* **309**, 484–496, 2007.

[96] Gary M. Seitz and Alexander E. Zalesskii. On the minimal degrees of projective representations of the finite Chevalley groups. II. J. Algebra **158**, 233–243, 1993.

[97] Ákos Seress. *Permutation group algorithms*, volume 152 of *Cambridge Tracts in Mathematics.* Cambridge University Press, Cambridge, 2003.

[98] Igor E. Shparlinski. *Finite fields: theory and computation. The meeting point of number theory, computer science, coding theory and cryptography.* Mathematics and its Applications, 477. Kluwer Academic Publishers, Dordrecht, 1999.

[99] Charles C. Sims. Computational methods in the study of permutation groups. In *Computational problems in abstract algebra*, pages 169–183, Oxford, 1970. (Oxford, 1967), Pergamon Press.

[100] V. Strassen. Gaussian elimination is not optimal. *Numer. Math.* **13**, 354–356, 1969.

[101] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*, Cambridge University Press, 2002.

[102] G.E. Wall. On the conjugacy classes in the unitary, symplectic and orthogonal groups. *J. Austral. Math. Soc.* **3**, 1–62, 1963.

[103] Robert A. Wilson. Standard generators for sporadic simple groups. *J. Algebra* **184**, 505–515, 1996.

[104] R.A. Wilson *et al.* Atlas of Finite Group Representations. `brauer.maths.qmul.ac.uk/Atlas`.

[105] R.A. Wilson. Computing in the Monster. In *Groups, Combinatorics & Geometry (Durham, 2001)*, 327–335, World Sci. Publishing, River Edge, NJ, 2003.

[106] Şükrü Yalçınkaya. Black box groups. *Turkish J. Math.* **31**, 171–210, 2007.

[107] K. Zsigmondy. Zur Theorie der Potenzreste. *Monatsh. für Math. u. Phys.* **3**, 265–284, 1892.