COMPUTING WITH MATRIX GROUPS OVER INFINITE FIELDS

A. S. Detinko[1], B. Eick[2], and D. L. Flannery[3]

[1,3] School of Mathematics, Statistics and Applied Mathematics, National University of Ireland, Galway, Ireland
Email: alla.detinko@nuigalway.ie; dane.flannery@nuigalway.ie

[2] Institut Computational Mathematics, TU Braunschweig, 38106 Braunschweig, Germany
Email: beick@tu-bs.de

Abstract

We survey currently available algorithms for computing with matrix groups over infinite domains. We discuss open problems in the area, and avenues for further development.

1 Introduction

The subject of linear groups is one of the main branches of group theory. Linear groups provide a link between group theory and natural sciences such as physics, chemistry, and genetics; as well as other areas of mathematics, including geometry, combinatorics, functional analysis, and differential equations.

The significance of linear groups was realized at the very beginning of group theory, dating back to work by C. Jordan (1870). In the early twentieth century, major successes in linear group theory were achieved by Burnside, Schur, Blichfeldt, and Frobenius; their results continue to exert an influence up to the present day.

Linear groups arise in various ways in the theory of abstract groups. For instance, they occur as groups of automorphisms of certain abelian groups, and they play a central role in the study of solvable groups. Furthermore, linearity is a vital property for some classes of groups: polycyclic-by-finite groups and countable free groups are prominent examples. Linear groups are closely associated to Lie groups, algebraic groups, and representation theory. For extra background we refer to [16, 48, 49, 50].

Advances in computational algebra have motivated a new phase in linear group theory. Matrix representations of groups have the advantage that a large (even infinite) group can be defined by input of small size. An illustration of this is the explicit realization of some large sporadic simple groups, in particular the Monster ([18, Section 5] and [23, p. 4]).

Currently, a very active area in the algorithmic theory of matrix groups is the so-called 'Matrix group recognition project'. This considers groups over finite fields,

Supported in part by Science Foundation Ireland, grants 07/MI/007 and 08/RFP/MTH1331.

with a principal aim of determining a composition or chief series ([28, 36]).

Separate difficulties may arise when computing with matrix groups over infinite domains. For example, some basic algorithmic problems are undecidable for such groups. Complexity issues, such as the uncontrollable growth of matrix entries, also cause complications.

We now state some of the problems for matrix groups over infinite fields that we believe are particularly important. Throughout, let $\mathbb{F}$ be an infinite (commutative) field and let $G \leq \mathrm{GL}(n, \mathbb{F})$ be given by a finite set S of generators.

Finiteness-type problems

- *Is G finite? If so, determine the order of G.*
- *Is G finitely presentable? If so, determine a finite presentation for G.*

Deciding finiteness is one of the first problems encountered when dealing with a potentially infinite group. Knowing a finite presentation may be helpful for further computation with the group.

Membership and conjugacy problems

- *Given $g \in \mathrm{GL}(n, \mathbb{F})$, is g contained in G? If so, express g as a word in S.*
- *Given $g, h \in G$, does there exist $x \in G$ such that $g^x = h$?*

Many computations with groups are based on positive solutions of these two problems. Hence one seeks practical algorithms to solve these problems for linear groups, wherever possible.

Structural problems

- *Is G solvable or solvable-by-finite?*
- *Is G polycyclic or polycyclic-by-finite?*
- *Is G nilpotent or nilpotent-by-finite?*

The first problem is closely connected to the 'Tits alternative' (see Theorem 2.3 below). The interest in the second and third problems stems from the fact that the groups in question allow effective computations via special techniques, some of which rely on certain finite presentations.

Action problems

- *Is $\mathbb{F}^n$ irreducible as a G-module?*
- *Given $x, y \in \mathbb{F}^n$, does there exist $g \in G$ with $xg = y$?*

Reduction to the case of irreducible groups is an old and common technique in linear group theory. Likewise, computational problems for matrix groups can be handled efficiently via irreducibility testing and construction of irreducible modules. Algorithms for the orbit problem are recognized as useful tools for the structural investigation of groups.

Our objective in this paper is to discuss some of these problems at greater length, and give an insight into practical algorithms for their solution.

2 Theoretical preliminaries

In this section we outline fundamental properties of finitely generated linear groups that foreshadow techniques used for computing with these groups.

2.1 Finite approximation

Let R denote the subring of $\mathbb{F}$ generated by the entries of the matrices in $S \cup S^{-1}$ (here S^{-1} denotes the set of inverses of the elements of S). Thus $G \leq \mathrm{GL}(n, R)$. Without loss of generality we may assume that $\mathbb{F}$ is the field of fractions of R. If ρ is an ideal of R then it induces a *congruence homomorphism* $\varphi_\rho : \mathrm{GL}(n, R) \to \mathrm{GL}(n, R/\rho)$, which replaces every entry in an element of S or S^{-1} by its image in R/ρ. The corresponding kernel G_ρ is called a *congruence subgroup* of G.

Each quotient ring R/ρ for a maximal ideal ρ is a finite field, and the intersection of all maximal ideals of R is zero. So in some sense R can be 'approximated' by finite fields. As a consequence, the following holds.

Theorem 2.1 (Mal'cev, 1940) *The group G is residually finite. Moreover, G is approximated by finite groups, each of which has a faithful representation of degree n over some finite field.*

Theorem 2.1 is background for the method of finite approximation in linear group theory. The theorem implies that a finitely generated simple linear group is finite ([49, Chapter 4]). The next result pushes the finite approximation further.

Theorem 2.2 (Selberg, 1960; Wehrfritz, 1970) *The group G contains a normal subgroup N of finite index such that all torsion elements of N are unipotent. In particular, if $\mathrm{char}\,\mathbb{F} = 0$ then N is torsion-free.*

For more details about the method of finite approximation we refer to [49, Chapters 4, 10] and [16, Chapter 10].

In practice, a congruence subgroup G_ρ can take the place of N in Theorem 2.2. A key issue then is to select ρ so that the corresponding congruence subgroup G_ρ satisfies the conditions of Theorem 2.2 (see Section 4 below).

2.2 The Tits alternative

We conclude this section by recalling the celebrated Tits alternative.

Theorem 2.3 (Tits, 1972) *A finitely generated linear group is either solvable-by-finite, or it has a non-abelian free subgroup.*

So finitely generated linear groups divide into two very different classes: in the class of solvable-by-finite groups many problems are decidable and computations are often feasible; while the class of groups with non-abelian free subgroups is much wilder and less well investigated.

3 Decidability

Before attempting to design an algorithm to solve a problem for any class of groups, one should ask whether such an algorithm even exists. In other words: is the problem decidable?

Results of Michailova (1958) imply that the membership problem is undecidable in general for subgroups of $\mathrm{GL}(4,\mathbb{Z})$. The same situation occurs for the conjugacy problem and the orbit problem, or, more generally, the orbit-stabilizer problem ([32, p. 42], [17, p. 239]). Furthermore, it is known that finitely generated linear groups may not be finitely presentable (see [49, p. 66]).

On the positive side, each finitely generated linear group has solvable word problem (Rabin; [49, p. 71]). There are also many positive results for special classes of linear groups. Kopytov [27] proved that the following problems for a finitely generated group over an algebraic number field are decidable: solvability testing, finiteness testing, and the membership problem for solvable groups. This implies decidability of the orbit problem for finitely generated completely reducible solvable groups over number fields ([17]).

Most computational problems are known to be decidable for polycyclic matrix groups over number fields. These groups are finitely presentable, and an algorithmically useful finite presentation can be computed for them [2, 37]. The word and membership problems can be solved [2], and using the finite presentation, many further structural problems have a practical solution [23, Chapter 8]. Also, the orbit-stabilizer problem is decidable in the class of polycyclic subgroups of $\mathrm{GL}(n,\mathbb{Z})$ ([17, 20]). Many of these results for polycyclic groups extend in principle to polycyclic-by-finite groups. However, practical algorithms are often available only for polycyclic groups.

4 The congruence homomorphism

In this section we discuss how to apply congruence homomorphism techniques in practice. The basic idea is to select an ideal ρ of R so that the torsion elements of G_ρ are unipotent (cf. Theorem 2.2). The essence of this method dates back to Minkowski.

As before, let R denote the subring of $\mathbb{F}$ generated by the entries of the matrices in $S \cup S^{-1}$, and assume that R has field of fractions $\mathbb{F}$. Then $\mathbb{F}$ is a finitely generated extension of its prime subfield $\mathbb{E}$. So for some $m \geq 0$ there exist algebraically independent indeterminates $x_1, \ldots, x_m$ over $\mathbb{E}$ such that $\mathbb{F}$ is a finite extension of $\mathbb{E}(x_1, \ldots, x_m)$. Replacing the elements of $\mathbb{F}$ with matrices over $\mathbb{E}(x_1, \ldots, x_m)$, we obtain an isomorphism of G onto a subgroup of $\mathrm{GL}(ne, \mathbb{E}(x_1, \ldots, x_m))$, where e is the degree of $\mathbb{F}$ over $\mathbb{E}(x_1, \ldots, x_m)$. Thus, without loss of generality, we can now assume that $\mathbb{F} = \mathbb{P}(x_1, \ldots, x_m)$, where $\mathbb{P}$ is either a number field or a finite field. If $\mathbb{F}$ is a number field then R is a Dedekind domain; otherwise, R is a UFD (unique factorization domain). Consequently the following lemma indicates how to select a suitable ideal ρ in all necessary cases (see [48, Chapter 3, Section 12] and [13, Section 3]).

Lemma 4.1

(i) *Let R be a UFD, $q \in R$ be irreducible, and $\rho = qR$. If* $\operatorname{char} R = 0$, *$q \nmid 2$, and $q^2 \nmid p$ for any prime $p \in \mathbb{Z}$, then G_ρ is torsion-free. If* $\operatorname{char} R > 0$ *then each torsion element of G_ρ is unipotent.*

(ii) *Let R be a Dedekind domain with prime ideal ρ. If* $\operatorname{char} R = 0$, *$2 \notin \rho$ and $p \notin \rho^2$ for all primes $p \in \mathbb{Z}$, then G_ρ is torsion-free. If* $\operatorname{char} R > 0$ *then each torsion element of G_ρ is unipotent.*

Let $m \geq 1$. With a slight abuse of notation we take $R = \frac{1}{\mu}\mathbb{P}[x_1, \ldots, x_m]$, where $\mu = \mu(x_1, \ldots, x_m)$ is a common multiple of the denominators of the entries of the matrices in $S \cup S^{-1}$. Define ρ to be the ideal $\langle (x_1 - \alpha_1), \ldots, (x_m - \alpha_m) \rangle$, where $\mu(\alpha_1, \ldots, \alpha_m) \neq 0$. Then ρ satisfies the conditions of Lemma 4.1. If $\mathbb{F} = \mathbb{P}$ is a number field, then see [13, Section 3] for construction of ρ as in Lemma 4.1. In particular, if $R \subseteq \mathbb{Q}$ and $\rho = qR$, $q > 2$ a prime in $\mathbb{Z}$, then G_ρ is torsion-free.

5 Deciding finiteness

Deciding finiteness is a fundamental problem in any class of potentially infinite groups. Recent progress [14, 15] proves that finiteness is decidable for linear groups over fields; moreover, the corresponding algorithms are practical. We summarize the methods here, and also discuss some other algorithms.

5.1 Finiteness testing over number fields

First, we note that a finite subgroup G of $\mathrm{GL}(n, \mathbb{Q})$ is conjugate to a subgroup of $\mathrm{GL}(n, \mathbb{Z})$. Section 3 of [7] gives a polynomial-time algorithm for testing whether G is conjugate to a group of integral matrices. If so, the algorithm constructs an appropriate conjugating matrix. The algorithm is based on manipulation with lattices, and estimates, in terms of input matrices, of the common denominator $D \in \mathbb{Z}$ such that $DG \subseteq \mathrm{GL}(n, \mathbb{Z})$. See [3, Section 5] and [38] for related results.

The papers [5, 7] contain several algorithms for deciding infiniteness of matrix groups over $\mathbb{Q}$. The most efficient algorithms feature random walk techniques ([4], [7, Section 4]). A key idea is that if G is infinite, then by a theorem of Schur, G contains an element of infinite order ([48, Chapter VI, Section 23]). The random walk method is used to search for an element of infinite order in G.

A subgroup G of $\mathrm{GL}(n, \mathbb{Z})$ is finite if and only if there exists a positive definite symmetric matrix B such that $gBg^T = B$ for all $g \in G$ ([34, p. 178]). The Monte-Carlo algorithm in [7, Section 6] for verifying finiteness first takes a special set of generators of G constructed by the random walk method, and then attempts to calculate B by an 'averaging trick' due to Babai and Friedl; see [7, Section 8.3] for various modifications. In practice, to get a definite answer to the question of whether G is finite, both of the above randomized algorithms run in parallel.

Two deterministic polynomial-time algorithms are proposed in [7]; they rely on testing whether G preserves a positive definite quadratic form. One algorithm is based on the observation that the set of all G-invariant quadratic forms is a vector space U, while a positive definite quadratic form f preserved by G is contained

in a compact convex subset of U. A search for f can be performed using the ellipsoid method described by Grötschel, Lovasz, and Schrijver ([7, Section 7.1]). The other algorithm uses some facts from representation theory of finite groups to compute the quadratic form $\frac{1}{|G|}\sum_{g\in G} gg^T$ (assuming G is finite), and then checks positive definiteness to verify the finiteness assumption ([7, Section 7.2]). Although both algorithms are polynomial-time, the above algorithms based on random walk techniques have proved to be more efficient.

So far the standard method for deciding finiteness of groups over an arbitrary number field begins by 'blowing up' the dimension of generating matrices to obtain a group over $\mathbb{Q}$ (cf. the discussion before Lemma 4.1).

5.2 Finiteness testing over function fields

The main idea here is to use the methods of Section 4 for $\mathbb{F} = \mathbb{P}(x_1, \ldots, x_m)$, $m \geq 1$. We first determine an ideal ρ of R as in Section 4, and test finiteness of $\varphi_\rho(G)$. The group $\varphi_\rho(G)$ is a subgroup of $\mathrm{GL}(n, \mathbb{P})$. If $\mathbb{P}$ is finite, then obviously this subgroup is finite. If $\mathbb{P}$ is a number field, then the methods of Section 5.1 can be used to decide finiteness of $\varphi_\rho(G)$.

Further, G is finite only if G_ρ is trivial ($\mathrm{char}\,\mathbb{F} = 0$), or unipotent ($\mathrm{char}\,\mathbb{F} > 0$). To check these properties, the finiteness testing algorithms of [14, 15] proceed by comparing the dimensions of the enveloping algebras $\langle G\rangle_{\mathbb{P}}$ and $\langle \varphi_\rho(G)\rangle_{\mathbb{P}}$. The special method for this purpose does not require computing a basis of $\langle G\rangle_{\mathbb{P}}$. Indeed, almost all the computation is carried out with matrices over the coefficient field $\mathbb{P}$ rather than the ground field $\mathbb{F}$.

There are other approaches to the finiteness problem over function fields, as in [42, Section 2] and [10, 25]. However, these approaches involve extensive computing over the ground field. Thus, in practice, they may work only for input of reasonably small size (including small degrees).

5.3 Computing the order

After we have recognized that a group is finite, the next problem is to find its order. The methods of Section 4 furnish a way to compute the order of a finite matrix group G over an infinite field, by constructing an isomorphic copy of G in some $\mathrm{GL}(n, q)$ and then computing the order of that image group. Algorithms to construct an isomorphic copy for groups over number fields or function fields (of zero and positive characteristic) are given in [13, Section 3], [14, Section 3.2.1], and [15, Section 3].

Notice that although algorithms for computing the order of a matrix group over a finite field are available, the problem of improving the efficiency of those algorithms is still open ([36], [44, Section 3]). Special methods for computing orders of elements of $\mathrm{GL}(n, q)$ were obtained by Celler and Leedham-Green ([36, Section 2]).

6 Computing with nilpotent matrix groups

This section focuses on algorithms for computing with nilpotent matrix groups over infinite fields.

Let $g = g_s g_u$ be the Jordan decomposition of $g \in \mathrm{GL}(n, \mathbb{F})$. That is, g_s, g_u are the unique matrices in $\mathrm{GL}(n, \overline{\mathbb{F}})$ such that g_s is diagonalizable, g_u is unipotent, and $g = g_s g_u = g_u g_s$. If $\mathbb{F}$ is perfect then $g_s, g_u \in \mathrm{GL}(n, \mathbb{F})$.

Define $G_s = \langle g_s \mid g \in S \rangle$ and $G_u = \langle g_u \mid g \in S \rangle$. These groups are straightforward to compute (see e.g. [6, Appendix A]). Moreover, as the following lemma shows, they determine nilpotency of G.

Lemma 6.1 *The group G is nilpotent if and only if G_s and G_u are nilpotent and $[G_s, G_u] = \langle 1_n \rangle$.*

This lemma is the basis of an effective nilpotency test. It also enables a reduction of several computational problems to the case of completely reducible groups.

Lemma 6.2 *Suppose that G is nilpotent.*

(i) *Let $\mathbb{F}$ be perfect. Then G is completely reducible if and only if $G = G_s$.*
(ii) *If G is completely reducible, then G is central-by-finite, and every normal torsion-free subgroup of G is central.*

Lemma 6.2 (ii) implies that completely reducible nilpotent linear groups are reasonably 'close' to finite matrix groups. Consequently, the methods of Section 4 are efficient for computing with nilpotent matrix groups.

For groups over finite fields, the paper [12] develops algorithms for nilpotency testing, as well as constructing presentations, computing Sylow subgroups, and calculating orders of nilpotent groups. Those algorithms form a background for computing with nilpotent groups over infinite fields via the methods of Section 4.

In the remainder of this section, $\rho \subseteq R$ is an ideal such that the torsion elements of G_ρ are unipotent.

6.1 Nilpotency testing

A nilpotency testing algorithm for groups defined over an infinite field is given in [13, Section 4.6]. The main steps are as follows.

(a) Construct G_s and G_u. Test whether G_u is nilpotent (that is, unipotent), and whether $[G_s, G_u] = \langle 1_n \rangle$.
(b) Construct $\varphi_\rho(G_s) \leq \mathrm{GL}(n, q)$ and test nilpotency of $\varphi_\rho(G_s)$.
(c) Test whether the congruence subgroup $(G_s)_\rho$ of G_s is central.

Step (a) is performed by testing whether G_u is conjugate to a subgroup of $\mathrm{UT}(n, \mathbb{F})$ (see [13, Section 4.1]). Step (b) is based on nilpotency testing as in [12]. For step (c) we take a presentation of $\varphi_\rho(G_s)$ and apply the 'normal subgroup generators' method (see [13, Section 4.2]).

6.2 Deciding finiteness and computing orders

In Section 5 we described algorithms for deciding finiteness and computing the order of a matrix group over an infinite field. In the special case of nilpotent groups, simpler and more efficient algorithms for these tasks are given in [13, Section 4.3] and [15, Section 4].

Suppose that $\operatorname{char}\mathbb{F} = 0$. If G_u is non-trivial then G is infinite. If $G = G_s$ then to decide finiteness of G it suffices to test whether the congruence subgroup G_ρ is trivial. This can be done readily using a presentation of $\varphi_\rho(G)$, since $\varphi_\rho(G) \leq \mathrm{GL}(n, q)$ is nilpotent.

Let $\operatorname{char}\mathbb{F} = p$, $\mathbb{F} = \mathbb{F}_q(x_1, \ldots, x_m)$, and set $\gamma = \lceil \log_p n \rceil$. The group G is finite if and only if the completely reducible group $H = \langle g_1^{p^\gamma}, \ldots, g_m^{p^\gamma} \rangle$ is finite. Finiteness of H can be decided by a special version of the general algorithm in Section 5.2 for completely reducible groups ([15, Section 3]).

Computing $|G|$ can be done as in Section 5.3, but at the final stage we compute $|\varphi_\rho(G)|$ using the algorithms from [11, 12].

6.3 Other algorithms: constructing presentations and Sylow subgroups

Since finitely generated nilpotent groups are polycyclic, finding presentations of nilpotent linear groups is an important problem. The following approach to this problem was proposed in [13, Section 4.4].

(a) Construct a presentation of $G_u \leq \mathrm{UT}(n, \mathbb{F})$.
(b) Construct a presentation of $\varphi_\rho(G) \leq \mathrm{GL}(n, q)$.
(c) Construct a presentation of the completely reducible abelian group $(G_s)_\rho$.

A solution of problem (b) was obtained in [11, 12]. The methods of [2] allow one to solve (a) and (c) over number fields. Note that this approach is simpler than the more general method for polycyclic groups (cf. Section 8.2).

Algorithms from [12, 13] provide structural information about a nilpotent group G. If G is finite then [12] gives an algorithm that constructs a series of G with small abelian factors. That algorithm is used further to find the Sylow system of G (see [11]). If G is infinite, then by computing a certain adjoint representation of G we can find the p-primary subgroups of G (see [13, Section 4.5]). Also, if $\mathbb{F}$ is perfect, then one can decide whether G is completely reducible (Lemma 6.2 (i)).

7 Solvable groups

The first algorithms for computing with finite solvable matrix groups were designed by E. Luks [31]. Drawing on those results, Beals [8] gave a Monte-Carlo algorithm for solvability testing of (infinite) matrix groups over number fields. This uses a reduction to finite fields via a congruence homomorphism, and relies on the fact that the derived length of a solvable linear group is bounded by a function of n ([48, Chapter V, Section 19]).

A practical deterministic algorithm for solvability testing of linear groups over number fields was obtained in [2], and implemented in [1]. In contrast to [8], the

algorithm of [2] investigates not only a congruence image of the group, but also the congruence subgroup. So it produces a definite answer. The analysis of the congruence subgroup is based on the following result of J. Dixon (see [17, Section 6] for a proof).

Theorem 7.1 (Dixon, 1985) *Let $G \leq \mathrm{GL}(n, R) \leq \mathrm{GL}(n, \mathbb{Q})$, and $\rho = pR$, $p \in \mathbb{Z}$ a prime greater than 2. Then G_ρ is connected in the Zariski topology. Therefore, if G is solvable then G_ρ is unipotent-by-abelian.*

7.1 Solvability testing of matrix groups

Algorithms testing solvability over $\mathbb{Q}$, and related procedures, were obtained in [2]. Solvability testing proceeds as follows.

(a) Compute $\rho = \langle p \rangle \subseteq R$, $p > 2$, and construct $\varphi_\rho(G) \leq \mathrm{GL}(n, p)$.
(b) Test solvability of $\varphi_\rho(G)$.
(c) Construct generators for the congruence subgroup G_ρ.
(d) Test whether G_ρ is unipotent-by-abelian.

Step (b)—testing solvability of a matrix group over a finite field—is performed in [2] by means of a procedure that draws on [45]. The implementation [1] provides the first practical solution of the solvability testing problem over finite fields. It additionally returns a (polycyclic) presentation of $\varphi_\rho(G)$ if this group is found to be solvable.

Step (c) uses the normal subgroup generators method, and the presentation found in step (b).

The key step of the algorithm is step (d). We outline the method used in [2] for testing whether G_ρ is unipotent-by-abelian. Let $V = \mathbb{Q}^n$, and $V = V_1 > \cdots > V_k > V_{k+1} = \{0\}$ be a semisimple series of V as a G_ρ-module. Clearly, G_ρ is unipotent-by-abelian if and only if G_ρ acts as an abelian group on every factor V_i/V_{i+1}. If G_ρ is non-abelian then $u = gh - hg \neq 0$ for some $g, h \in G_\rho$, and u defines a non-trivial G_ρ-module $W \leq V$ which yields a reduction of the problem to groups of smaller degrees; that is, $G_\rho|W$ and the restriction to V/W of the stabilizer $\mathrm{Stab}_{G_\rho}(V/W)$. In finitely many steps the procedure recognizes whether G_ρ is unipotent-by-abelian. For details see [1, 2].

7.2 Other algorithms

Similarly to Section 6.2, results from [2] imply an algorithm for deciding finiteness of solvable subgroups of $\mathrm{GL}(n, \mathbb{Q})$ via testing whether G_ρ is non-trivial. Also, construction of an isomorphic congruence image of G in $\mathrm{GL}(n, p)$, together with methods for computing orders of finite polycyclic groups ([45]), provide an efficient way to compute the order of a finite solvable subgroup of $\mathrm{GL}(n, \mathbb{Q})$.

As a by-product of the main algorithms in [2], we gain a simple test of whether a solvable group G is completely reducible: it is enough to decide whether G_ρ is a completely reducible abelian group. In turn, this can by done by testing whether the unipotent parts of the generators of G_ρ are trivial (cf. Lemma 6.2 (ii)).

Note that while membership testing for solvable matrix groups over number fields is decidable ([27]), no practical algorithms for this problem are known.

8 Polycyclic matrix groups

In this section we consider algorithms for computing with polycyclic matrix groups over $\mathbb{Q}$.

Although polycyclic groups are a partial case of solvable groups, they possess significant computational advantages. For instance, a polycyclic group has a finite presentation of a very useful form; namely, a polycyclic presentation. A broad variety of algorithms based on polycyclic presentations can be applied to computing with polycyclic groups (see [23, 46]). So the determination of such a presentation is of high interest. The following is a variation of Theorem 7.1.

Theorem 8.1 *Let $G \leq \mathrm{GL}(n, R) \leq \mathrm{GL}(n, \mathbb{Q})$, and $\rho = pR$, $p \in \mathbb{Z}$ a prime greater than 2. Then G is polycyclic if and only if G_ρ is (finitely-generated unipotent)-by-abelian.*

Using the solvability testing method of Section 7.1, we obtain a homomorphism $\lambda : G_\rho \to G_1 \times \cdots \times G_k$, where G_i is induced by action of G_ρ on the quotient V_i/V_{i+1} of a semisimple series. The group G is solvable if $\varphi_\rho(G) \leq \mathrm{GL}(n, \mathbb{F}_p)$ is solvable, λ has abelian image, and $\ker \lambda$ is finitely generated. Confirming the third condition is the most difficult step. In the next subsection we summarize the method of [3] for this problem.

8.1 Mal'cev correspondence

Let U denote the kernel of λ. As the image of λ can be explicitly computed, we can find a polycyclic presentation on generators $g_1, \ldots, g_m$, say, for the image. We use this to obtain normal subgroup generators for U. Note that, by construction, U is a subgroup of the group $\mathrm{Tr}_1(n, \mathbb{Q})$ of upper unitriangular matrices.

Next, we define $\mathcal{L}(U) = \mathbb{Q}\log(U)$ where $\log : \mathrm{Tr}_1(n, \mathbb{Q}) \to \mathrm{Tr}_0(n, \mathbb{Q})$ is the logarithm of unitriangular matrix groups. This has the structure of a Lie algebra over $\mathbb{Q}$. Using the normal subgroup generators of U we can determine a basis B of $\mathcal{L}(U)$.

The group G acts on $\mathcal{L}(U)$ via the conjugation action of G on its normal subgroup U. Let $\overline{g}_1, \ldots, \overline{g}_m$ denote the action with respect to B of the images of the generators $g_1, \ldots, g_m$ for G/U on $\mathcal{L}(U)$. Let χ_i denote the minimal polynomial of $\overline{g}_i$. The following is proved in [3].

Lemma 8.2 *U is finitely generated if and only if $\chi_i \in \mathbb{Z}[x]$ and χ_i has constant term ± 1 for $1 \leq i \leq m$.*

This lemma allows one to readily check whether U is finitely generated, and hence facilitates an effective test for polycyclicity of matrix groups over number fields.

8.2 Construction of polycyclic presentations

Once G is found to be a polycyclic group, it is straightforward to compute a polycyclic presentation for G following the setup of the previous section. It only remains to determine a polycyclic presentation for the finitely generated unipotent group U, and then combine it with the already determined polycyclic presentation of G/U.

8.3 The orbit-stabilizer problem

For linear groups over $\mathbb{Q}$, the orbit-stabilizer problem is restricted naturally to polycyclic-by-finite groups (see [17]).

A first method to compute orbits and stabilizers for the action of a nilpotent-by-finite subgroup G of $\mathrm{GL}(n, \mathbb{Q})$ on $\mathbb{Q}^n$ was presented in [17]. This method applies a congruence homomorphism to G and uses the property of the congruence subgroup noted in Theorem 7.1.

The more recent, practical approach in [20] also employs congruence homomorphisms. That paper relies on the same construction as the one used to compute a polycyclic presentation, and methods of algebraic number theory.

8.4 Membership testing

A polynomial-time algorithm to test membership in abelian subgroups of $\mathrm{GL}(n, \mathbb{F})$, where $\mathbb{F}$ is a number field, was developed in [6] (for another algorithm dealing with this problem see [37]). The results of [6] depend heavily on the special algorithm for the case $n = 1$ in [21]. Membership testing for abelian-by-finite subgroups of $\mathrm{GL}(n, \mathbb{F})$ is considered in [8]. For polycyclic subgroups of $\mathrm{GL}(n, \mathbb{Q})$, the problem can be solved using membership testing algorithms for polycyclic groups ([19]), and the algorithms in [1, 3] for constructing polycyclic presentations (see Section 8.2).

9 Testing virtual properties; a computational analogue of the Tits alternative

Once we have algorithms that handle nilpotent, polycyclic, and solvable linear groups, the next step is to devise algorithms for virtually nilpotent, virtually polycyclic, and virtually solvable linear groups. Algorithms testing whether $G \leq \mathrm{GL}(n, \mathbb{F})$ is solvable-by-finite would give us a computational analogue of the Tits alternative: a verification of whether G contains a non-abelian free subgroup.

Let $\mathbb{F}$ be a number field. The first algorithms for testing virtual solvability over $\mathbb{F}$ appeared in [8], and were subsequently improved in [9]. The algorithms of [8] use computing with matrix algebras [43] and adjoint representations. This latter feature leads to extensive computation with subgroups of $\mathrm{GL}(m, \mathbb{F})$, $m \leq n^2$, rendering the algorithms impractical.

A different approach to the problem, based on methods from [17], was proposed in [37]. To test whether G is solvable-by-finite, by Theorem 7.1 it is enough to test whether a congruence subgroup G_ρ is unipotent-by-abelian. Although that problem

was solved in [1, 2] (see Section 7.1), construction of the congruence subgroup G_ρ is a computational challenge. As a result, the problem of testing virtual solvability so far remains open. The same is true for testing virtual nilpotency and virtual polycyclicity.

One advantage of testing virtual properties via congruence homomorphisms is that having G_ρ, which is a unipotent-by-abelian subgroup of finite index, we can tackle other computational problems: for example, orbit-stabilizer and membership problems (cf. [17]).

The next two related open problems are as follows: (i) given that G is not solvable-by-finite, construct a free non-abelian subgroup of G; and (ii) given $G = \langle g, h\rangle \leq \mathrm{GL}(n, \mathbb{F})$, test whether G is a free group.

Testing virtual solvability over fields other than number fields has not been considered at all.

10 Irreducibility testing and related problems

We now discuss irreducibility testing and construction of irreducible modules. Over finite fields, a solution of these problems based on the Meataxe procedure is a vital part of computing with subgroups of $\mathrm{GL}(n, q)$ ([28, 36]). One might seek a Meataxe for (at least finite) matrix groups over infinite fields. Research in that direction is reported in [22, 24, 38].

Notice that in contrast to groups over finite fields, structural analysis of finite groups over infinite fields can be done without computing irreducible modules, but rather via direct construction of an isomorphic copy in $\mathrm{GL}(n, q)$—see Section 4. Further, each infinite finitely generated linear group is non-simple and, moreover, has normal subgroups of finite index. Thus, for infinite groups (especially solvable-by-finite groups), it is natural to test complete reducibility (and more) by use of congruence homomorphisms; cf. Section 7.2.

One more approach to irreducibility testing and construction of modules is by applying algorithms for matrix algebras. Here the central problems are computing the radical of the enveloping algebra, finding the Wedderburn decomposition of a semisimple algebra, and expressing a simple algebra as a direct sum of minimal left ideals (see [43, Section 6] for details). The latter problem is equivalent to computing irreducible components of the group. This implies that, over a number field, finding irreducible components is at least as difficult as factorizing square-free integers ([43, Section 6]). On the other hand, for abelian groups over number fields, one can test irreducibility and construct irreducible modules ([2, Section 5.2]).

A number of algorithms for the above problems (particularly computing the radical, and Wedderburn decomposition) over various fields are now available; see [26, 43]. For recent advances dealing with groups over $\mathbb{Q}$, see e.g. [33, 47].

Along with irreducibility testing, one would like to be able to test primitivity and construct systems of imprimitivity; over infinite fields so far those problems have not been explored.

11 Linearity of groups and representation theory

An area closely related to computing with linear groups is construction of faithful representations of (abstract) groups. That work really lies in the province of computational representation theory (see e.g. [39] and [41]), which is beyond the scope of this survey. Nevertheless, this final section gives a brief account of results for groups which have linearity as a crucial property (see [49, Chapter 2] for a treatment of such groups).

The paper [35] contains an efficient algorithm for constructing a representation of a finitely generated torsion-free nilpotent group in $\mathrm{GL}(n, \mathbb{Z})$. Algorithms for constructing representations of polycyclic groups in $\mathrm{GL}(n, \mathbb{Z})$ are proposed in [30]. However, no practical algorithms for polycyclic or polycyclic-by-finite groups are currently available.

The wider problem of constructing representations of finitely presented groups is discussed in [40]. Special attention is paid in [40] to the impact of that problem on deciding finiteness of finitely presented groups. One example of relevant existing algorithms is the procedure, analogous to Todd-Coxeter coset enumeration, given in [29].

References

[1] B. Assmann and B. Eick, Polenta—Polycyclic presentations for matrix groups. A refereed GAP 4 package; see http://www.gap-system.org/Packages/polenta.html (2007).

[2] ———, *Computing polycyclic presentations for polycyclic rational matrix groups*, J. Symbolic Comput. **40** (2005), no. 6, 1269–1284.

[3] ———, *Testing polycyclicity of finitely generated rational matrix groups*, Math. Comp. **76** (2007), 1669–1682.

[4] L. Babai, *Local expansion of vertex-transitive graphs and random generation in finite groups*, Proceedings of the twenty-third annual ACM symposium on Theory of computing (New Orleans, LA, 1991) (New York), ACM, 1991, pp. 164–174.

[5] ———, *Deciding finiteness of matrix groups in Las Vegas polynomial time*, Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms (Orlando, FL, 1992) (New York), ACM, 1992, pp. 33–40.

[6] L. Babai, R. Beals, J. Cai, G. Ivanyos, and E. M. Luks, *Multiplicative equations over commuting matrices*, Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms (Atlanta, GA, 1996), ACM, New York (1996), 498–507.

[7] L. Babai, R. Beals, and D. N. Rockmore, *Deciding finiteness of matrix groups in deterministic polynomial time*, Proc. of International Symposium on Symbolic and Algebraic Computation ISSAC '93 (ACM Press), 1993, pp. 117–126.

[8] Robert Beals, *Algorithms for matrix groups and the Tits alternative*, J. Comput. System Sci. **58** (1999), no. 2, 260–279, 36th IEEE Symposium on the Foundations of Computer Science (Milwaukee, WI, 1995).

[9] ———, *Improved algorithms for the Tits alternative*, Groups and computation, III (Columbus, OH, 1999), Ohio State Univ. Math. Res. Inst. Publ., vol. 8, de Gruyter, Berlin, 2001, pp. 63–77.

[10] A. S. Detinko, *On deciding finiteness for matrix groups over fields of positive characteristic*, LMS J. Comput. Math. **4** (2001), 64–72 (electronic).

[11] A. S. Detinko, B. Eick, and D. L. Flannery, Nilmat—Computing with nilpotent matrix groups. A refereed GAP 4 package; see http://www.gap-system.org/Packages/nilmat.

html (2007).
[12] A. S. Detinko and D. L. Flannery, *Computing in nilpotent matrix groups*, LMS J. Comput. Math. **9** (2006), 104–134 (electronic).
[13] ———, *Algorithms for computing with nilpotent matrix groups over infinite domains*, J. Symbolic Comput. **43** (2008), 8–26.
[14] ———, *On deciding finiteness of matrix groups*, J. Symbolic Comput. **44** (2009), 1037–1043.
[15] A. S. Detinko, D. L. Flannery, and E. A. O'Brien, *Deciding finiteness of matrix groups in positive characteristic*, J. Algebra **322** (2009), 4151–4160.
[16] J. D. Dixon, *The structure of linear groups*, Van Nostrand Reinhold, London, 1971.
[17] ———, *The orbit-stabilizer problem for linear groups*, Canad. J. Math. **37** (1985), no. 2, 238–259.
[18] B. Eick, *Computational group theory*, Jahresbericht der DMV 107, Heft 3 (2005), 155–170.
[19] B. Eick and W. Nickel, Polycyclic—Computation with polycyclic groups. A refereed GAP 4 package; see http://www.gap-system.org/packages/polycyclic.html (2004).
[20] Bettina Eick and Gretchen Ostheimer, *On the orbit-stabilizer problem for integral matrix actions of polycyclic groups*, Math. Comp. **72** (2003), no. 243, 1511–1529 (electronic).
[21] G. Ge, *Algorithms related to multiplicative representations of algebraic numbers*, Ph.D. thesis, U. C. Berkeley, 1993.
[22] S. P. Glasby, *The Meat-Axe and f-cyclic matrices*, J. Algebra **300** (2006), no. 1, 77–90.
[23] D. F. Holt, B. Eick, and E. A. O'Brien, *Handbook of computational group theory*, Chapman & Hall/CRC Press, Boca Raton, London, New York, Washington, 2005.
[24] Derek F. Holt, *The Meataxe as a tool in computational group theory*, The atlas of finite groups: ten years on (Birmingham, 1995), London Math. Soc. Lecture Note Ser., vol. 249, Cambridge Univ. Press, Cambridge, pp. 74–81.
[25] G. Ivanyos, *Deciding finiteness for matrix semigroups over function fields over finite fields*, Israel J. Math. **124** (2001), 185–188.
[26] Gábor Ivanyos and Lajos Rónyai, *Computations in associative and Lie algebras*, Some tapas of computer algebra, Algorithms Comput. Math., vol. 4, Springer, Berlin, 1999, pp. 91–120.
[27] V. M. Kopytov, *The solvability of the occurrence problem in finitely generated solvable matrix groups over an algebraic number field*, Algebra i Logika **7** (1968), no. 6, 53–63 (Russian).
[28] C. R. Leedham-Green, *The computational matrix group project*, Groups and computation, III (Columbus, OH, 1999), Ohio State Univ. Math. Res. Inst. Publ., vol. 8, de Gruyter, Berlin, 2001, pp. 229–247.
[29] S. A. Linton, *On vector enumeration*, Linear Algebra Appl. **192** (1993), 235–248, Computational linear algebra in algebraic and related problems (Essen, 1992).
[30] Eddie H. Lo and Gretchen Ostheimer, *A practical algorithm for finding matrix representations for polycyclic groups*, J. Symbolic Comput. **28** (1999), no. 3, 339–360.
[31] E. Luks, *Computing in solvable matrix groups*, Proc. 33rd IEEE Symposium on Foundations of Computer Science, pp. 111–120, 1992.
[32] Charles F. Miller, III, *On group-theoretic decision problems and their classification*, Princeton University Press, Princeton, N.J., 1971, Annals of Mathematics Studies, No. 68.
[33] G. Nebe and A. Steel, *Recognition of division algebras*, J. Algebra **322** (2009), 903–909.
[34] Morris Newman, *Integral matrices*, Academic Press, New York, 1972.
[35] Werner Nickel, *Matrix representations for torsion-free nilpotent groups by Deep Thought*, J. Algebra **300** (2006), no. 1, 376–383.

[36] E. A. O'Brien, *Towards effective algorithms for linear groups*, Finite geometries, groups, and computation, Walter de Gruyter, Berlin, 2006, pp. 163–190.
[37] Gretchen Ostheimer, *Practical algorithms for polycyclic matrix groups*, J. Symbolic Comput. **28** (1999), no. 3, 361–379.
[38] Richard A. Parker, *An integral meataxe*, The atlas of finite groups: ten years on (Birmingham, 1995), London Math. Soc. Lecture Note Ser., vol. 249, Cambridge Univ. Press, Cambridge, 1998, pp. 215–228.
[39] W. Plesken, *Finite rational matrix groups: a survey*, The atlas of finite groups: ten years on (Birmingham, 1995), London Math. Soc. Lecture Note Ser., vol. 249, Cambridge Univ. Press, Cambridge, 1998, pp. 229–248.
[40] ———, *Presentations and representations of groups*, Algorithmic algebra and number theory (Heidelberg, 1997), Springer, Berlin, 1999, pp. 423–434.
[41] Wilhelm Plesken and Bernd Souvignier, *Constructing rational representations of finite groups*, Experiment. Math. **5** (1996), no. 1, 39–47.
[42] D. N. Rockmore, K.-S. Tan, and R. Beals, *Deciding finiteness for matrix groups over function fields*, Israel J. Math. **109** (1999), 93–116.
[43] Lajos Rónyai, *Computations in associative algebras*, Groups and computation (New Brunswick, NJ, 1991), DIMACS Ser. Discrete Math. Theoret. Comput. Sci., vol. 11, Amer. Math. Soc., Providence, RI, 1993, pp. 221–243.
[44] Ákos Seress, *A unified approach to computations with permutation and matrix groups*, International Congress of Mathematicians. Vol. II, Eur. Math. Soc., Zürich, 2006, pp. 245–258.
[45] Charles C. Sims, *Computing the order of a solvable permutation group*, J. Symbolic Comput. **9** (1990), no. 5-6, 699–705, Computational group theory, Part 1.
[46] ———, *Computation with finitely presented groups*, Encyclopedia of Mathematics and Its Applications, vol. 48, Cambridge University Press, New York, 1994.
[47] Bernd Souvignier, *Decomposing homogeneous module of finite groups in zero characteristic*, J. Algebra **322** (2009), 948–956.
[48] D. A. Suprunenko, *Matrix groups*, Transl. Math. Monogr., vol. 45, American Mathematical Society, Providence, RI, 1976.
[49] B. A. F. Wehrfritz, *Infinite linear groups*, Springer-Verlag, 1973.
[50] A. E. Zalesskiĭ, *Linear groups*, Russian Math. Surveys **36** (1981), no. 5, 63–128.