'Computational Group Theory': reference number 1631; 31 July - 6 August 2016.

Recent advances in computing with infinite linear groups

A. Detinko, D. Flannery

We report on recent developments in computing with linear groups given by a finite set of generating matrices over an infinite field. In previous research [1] we developed effective methods for computing in this class of groups, used those methods to solve a number of computational problems, and designed software for practical computing. The problems solved include finiteness testing and testing (virtual) solvability over an arbitrary field, as well as structural investigation of solvable, nilpotent, and finite groups.

Further developments have occurred in two directions: (i) computing in solvable-by-finite groups; (ii) algorithms for groups containing a free non-abelian subgroup.

1. Algorithms for (virtually) solvable groups

Motivation. The theory of infinite solvable groups has played a central role in group theory over the past seventy years. Furthermore, solvable linear groups constitute a major component in the investigation of (abstract) solvable groups.

Challenges (solvable vs polycyclic). In contrast to (virtually) polycyclic groups, solvable groups may not be finitely presentable, they may contain subgroups that are not finitely generated, and they do not satisfy the maximal condition on subgroups. The failure of these properties poses severe difficulties in the design of algorithms for solvable groups.

Method. We initiated and developed a new approach to computing with (virtually) solvable linear groups, based on rank restrictions. Notice that finitely generated linear groups have finite Prüfer rank if and only if they are solvable-by-finite and $\mathbb{Q}$-linear.

Algorithms. Given a finitely generated solvable-by-finite subgroup G of $\mathrm{GL}(n, \mathbb{F})$, the following algorithms have been developed.

- Computing the torsion-free rank of G and bounds on its Prüfer rank [5, Section 4.4] when $\mathbb{F}$ is a number field.
- If H is a finitely generated subgroup of G, we can test whether $|G : H|$ is finite.
- Construction of a generating set of the completely reducible part of G. This includes testing whether G itself is completely reducible, and whether G is unipotent (i.e., is upper unitriangular in some basis) [5, Section 4.2].

Software. The algorithms are implemented in the package [6].

This research is joint with Eamonn O'Brien.

Applications. Applying the above results, we obtained a practical algorithm for arithmeticity testing of finitely generated subgroups of solvable algebraic $\mathbb{Q}$-groups. This involves a new efficient algorithm for testing integrality of a finitely generated solvable subgroup of $\mathrm{GL}(n, \mathbb{Q})$.

This is joint with Willem de Graaf.

2. Algorithms for semi-simple arithmetic groups

Most finitely generated linear groups are not virtually solvable, and comprise a broad variety of different types of groups. At this stage we restrict our attention to arithmetic subgroups of a semi-simple algebraic $\mathbb{Q}$-group $\mathcal{G}$.

Motivation.

- The class of arithmetic groups is an important class of finitely generated linear groups; moreover, computing with arithmetic subgroups is currently in high demand, especially due to the connections with number theory, topology, and physics.

- Fundamental algorithmic problems are known to be decidable (for *explicitly given* arithmetic groups, as defined by Grunewald & Segal, 1980).

We consider $\mathcal{G} = \mathrm{SL}$ or Sp, $n > 2$. These are prominent examples of groups with the congruence subgroup property (CSP): i.e., each arithmetic subgroup H of $\Gamma_n := \mathrm{SL}(n,\mathbb{Z}), \mathrm{Sp}(n,\mathbb{Z})$ contains a principal congruence subgroup (PCS) $\Gamma_{n,m}$ of level m, which is the kernel $\Gamma_{n,m}$ of the reduction modulo m homomorphism on $\mathcal{G}(\mathbb{Z})$.

Method. We developed methods for practical computing with arithmetic subgroups based on the congruence homomorphism technique. The two main components are computing the level M of the maximal principal congruence subgroup of an arithmetic group H; and computing with congruence images of H, which are matrix groups over the finite ring $\mathbb{Z}_m$.

Algorithms. Let H be an arithmetic subgroup of Γ_n given by a finite set S of generating matrices. We list below the functions designed to handle these groups via computer.

2.1. **Computing the level and related procedures.**

- `LevelMaxPCS`(H) computes the level M of the maximal principal congruence subgroup $\Gamma_{n,M}$ of H. More generally, `LevelMaxPCS` takes as input a generating set of a dense subgroup and returns the level of its minimal arithmetic overgroup.
- `Index`(Γ_n, H) returns the index of H in Γ_n. As an application, we can test whether $H = \Gamma_n$.
- `IsIn`(H, g) tests membership of $g \in \Gamma_n$ in H. More generally, `IsSubgroup`(H, H_1) returns `true` if and only if the finitely generated subgroup H_1 of Γ_n is in H.
- `Intersect`(H, H_1) returns a generating set of the intersection of H and an arithmetic subgroup H_1 of Γ_n.

2.2. **Investigating subgroup structure.** The structure of an arithmetic group is defined to some extent by its (sub)normal subgroups (e.g., its PCS).

- `IsSubnormal`(H): tests whether H is subnormal in Γ_n.
- `Normalizer`(H) returns a generating set of $N_{\Gamma_n}(H)$.
- `NormalClosure`(H) returns a generating set of $\langle H \rangle^{\Gamma_n}$; here H is an arbitrary finitely generated subgroup of Γ_n.

Method: All algorithms are based on `LevelMaxPCS`(H) and our library of functions for subnormal subgroups of matrix groups over $\mathbb{Z}_m$; see [2, Section 3.1]. The algorithms also involve computing the ideal generated by the entries of the matrices in S [2, Section 1.5, 3.2].

2.3. **Orbit-stabilizer problem.** Let H be an arithmetic subgroup H of Γ_n given by a finite generating set of matrices, and let u, v be vectors in $\mathbb{Q}^n$.

- `Orbit`(H, u, v) tests whether $\exists\ g \in H$ such that $g(u) = v$, and returns such an element if such exists.
- `Stabilizer`(H, u) returns a generating set of $\mathrm{Stab}_H(u)$.

N.B.: $\mathrm{Stab}_H(u)$ is a finitely generated group.

Method: solution of the orbit-stabilizer problem for $\varphi_m(H)$ acting on $\mathbb{Z}_m^n$ and for a PCS $\Gamma_{n,m}$ of H; see [2, Section 4].

2.4. **Application.** We extended our methods and algorithms to the wider class of Zariski dense subgroups of $\mathcal{G}(\mathbb{C})$. This includes computing the 'arithmetic closure' (i.e. the minimal arithmetic overgroup) of a finitely generated subgroup $H \leq \mathcal{G}(\mathbb{Z})$ dense in $\mathcal{G}(\mathbb{C})$; here $\mathcal{G} = \mathrm{SL}$ or Sp. Using our GAP implementation of the algorithms, we solved various problems for classes of groups which have emerged recently in areas of mathematics and its applications [3].

The results of Section 2 are joint work with Alexander Hulpke.

We also present a number of open problems that are important for further development of the area.

REFERENCES

1. A. Detinko, D. Flannery *Computing with matrix groups over infinite fields*, Oberwolfach Reports, Volume 8, Issue 3, 2011, 2118 - 2121.
2. A. S. Detinko, D. L. Flannery, A. Hulpke, *Algorithms for arithmetic groups with the congruence subgroup property*, J. Algebra **421** (2015), 234–259.
3. ———, *Zariski density and computing in arithmetic groups*, preprint.
4. A. Detinko, D. Flannery, W. de Graaf *Integrality and arithmeticity of solvable linear groups*, Journal of Symbolic Computation, 68 (2015) 138-145.
5. A. Detinko, D. Flannery, E. O'Brien *Algorithms for linear groups of finite rank*, Journal of Algebra, 393 (2013), 187-196.
6. ———, *Infinite—Computing with matrix groups over infinite fields*; `http://magma.maths.usyd.edu.au/magma/handbook/matrix_groups_over_infinite_fields` (2012).