Snapshots of modern mathematics
from Oberwolfach

№ 5/2019

Algebra, matrices, and computers

A. S. Detinko ● D. L. Flannery
A. Hulpke

What part does algebra play in representing the real world abstractly? How can algebra be used to solve hard mathematical problems with the aid of modern computing technology? We provide answers to these questions that rely on the theory of matrix groups and new methods for handling matrix groups in a computer.

1 Introduction

Nowadays, computers enable us not just to calculate, but also to use mathematics in completely new ways. A real-life problem can sometimes be framed in mathematical terms, providing a mathematical model of the problem. After converting the model into a form that is understandable by a computer, we then program the computer with an *algorithm* (a sequence of instructions) to find a solution.[1]

In this snapshot we consider mathematical problems that can be solved by algebra with matrices, and focus on the hurdles that may be encountered in using a computer to obtain solutions.

[1] The word 'algorithm' derives from the name of the Persian mathematician Muhammad al-Khwarizmi (c. 780–c. 850); see http://www-history.mcs.st-and.ac.uk/Biographies/Al-Khwarizmi.html.

2 Basic ideas about matrix groups

2.1 Matrices

For complicated tasks, ordinary numbers (such as the real or complex numbers) are not enough. To deal with these tasks, we need generalizations of the notion of number. One such generalization is tables of numbers, called *matrices*. If the table has m rows and n columns, where each entry in the table belongs to a set $\mathbb{F}$, then we say that it is an $m \times n$ *matrix over* $\mathbb{F}$. If $m = 1$ or $n = 1$ then the matrix is a *vector*.

Matrices are exceptionally useful tools. As just one application, we note that the need to solve a system of linear equations arises frequently in real-world problems (for example, from engineering), and the system is naturally described as a matrix of the constants in the equations.

Moreover, $n \times n$ matrices have a geometrical interpretation as models of linear transformations of n-dimensional Euclidean space. Examples are the Cartesian (or x-y) plane when $n = 2$, or our surrounding three-dimensional space when $n = 3$. A transformation A of the space is *linear* if it respects addition and scalar multiplication of vectors within the space, that is, $A(x + y) = A(x) + A(y)$ and $A(sx) = sA(x)$ for all vectors x, y and any number (scalar) s; see Section 2.2 for the definition of matrix operations. If matrices are selected according to specific restrictions, then we can model linear transformations that preserve a metric (a measurement such as length, angle, volume; see Example 1 below), or preserve symmetry of a geometrical object (such as one of the Platonic solids: tetrahedron, cube, octahedron, dodecahedron, icosahedron).

2.2 Operations on matrices

Algebraic operations combine matrices over $\mathbb{F}$ to produce more matrices over $\mathbb{F}$. We can add, subtract, multiply, and divide numbers. Can we do the same with matrices? The answer is a qualified “yes”. Addition and subtraction of $n \times n$ matrices are done entry-by-entry, extending these operations on numbers. The multiplicative product of $n \times n$ matrices A and B is calculated differently. It may be defined as the result of applying the transformations represented by A and B one after the other. We write the product as AB. (More generally, we can multiply an $m \times n$ matrix by an $n \times r$ matrix. That is, AB is defined whenever the number of columns in A is equal to the number of rows in B.)

Some familiar properties of operations on numbers no longer hold for matrix multiplication. For example, multiplication of numbers is *commutative*, meaning that $ab = ba$ for all numbers a and b, whereas $AB = BA$ may not be true for arbitrary matrices A and B. Furthermore, any non-zero number r has a reciprocal $r^{-1} = \frac{1}{r}$ that is used to undo multiplication by r (to divide by r).

However, a matrix A need not have a multiplicative *inverse* A^{-1}: we may not be able to do division when working with arbitrary matrices.

There is a special number in $\mathbb{F}$ associated to each $n \times n$ matrix over $\mathbb{F}$, its *determinant*, that tells us whether it has an inverse. To be more precise, the matrix is invertible if and only if its determinant is invertible in $\mathbb{F}$. The set of all invertible $n \times n$ matrices over $\mathbb{F}$ is denoted $\mathrm{GL}(n, \mathbb{F})$. Therefore, each A in $\mathrm{GL}(n, \mathbb{F})$ has an inverse B in $\mathrm{GL}(n, \mathbb{F})$ such that $AB = BA = \mathbb{1}_n$. Here $\mathbb{1}_n$ is the $n \times n$ *identity matrix*, with 1s all down its main diagonal and zeros everywhere else. Note that $\mathbb{1}_n$ is an element of $\mathrm{GL}(n, \mathbb{F})$.

Example 1. An $n \times n$ matrix that preserves the length of every Euclidean vector of length n is called *orthogonal.* It can be shown that an orthogonal matrix M has an inverse: the *transpose* $M^\top$ of M, whose entry in row r, column c is equal to the entry in row c, column r of M.

In dimension $n = 2$, the matrix

$$M = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

represents a rotation of the plane about the origin through an angle of θ radians. We have $MM^\top = M^\top M = \mathbb{1}_2$ because $\cos^2\theta + \sin^2\theta = 1$. The length $l(v) \geq 0$ of any vector $v = (x, y)^\top$ is defined by $l(v)^2 = v^\top v = x^2 + y^2$. We rotate v through θ radians by matrix multiplication with M, to get $w = Mv$. Then w and v have the same length: $l(w)^2 = w^\top w = v^\top M^\top M v = v^\top \mathbb{1}_2 v = v^\top v = l(v)^2$.

2.3 Matrix entries

We say a little more about the set $\mathbb{F}$ from which matrix entries are drawn. This symbol $\mathbb{F}$ will denote various generalizations of the natural numbers $0, 1, 2, \ldots$ Instances of $\mathbb{F}$ include:

- the integers $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$;
- the rationals $\mathbb{Q}$ (all fractions $\frac{a}{b}$ for integers a and b, $b \neq 0$);
- the real numbers $\mathbb{R}$ (all points on a straight line);
- the set $\mathbb{F}_p$ of residues in $\mathbb{Z}$ modulo a fixed prime p (recall that a prime p is an integer greater than 1 whose only positive integer divisors are 1 and p). Each integer m can be written uniquely as $m = pk + r$ where r, k are integers and $0 \leq r < p$. The number r is the *residue of m modulo p.* Clearly $\mathbb{F}_p$ is a finite set, of size p.

A common feature of each set $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{F}_p$ is that adding, subtracting, and multiplying its elements produces an element from the same set (for $\mathbb{F}_p$, this is the foundation of *modular arithmetic*). Apart from $\mathbb{Z}$, we can also divide by non-zero elements within each set. We call $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{F}_p$ *fields*, while $\mathbb{Z}$ is a *ring*. Any subset G of $\mathrm{GL}(n, \mathbb{F})$ containing $\mathbb{1}_n$ that is 'closed under multiplication and inverses' (contains AB and A^{-1}, for all A, B in G) is a *matrix group over* $\mathbb{F}$. We observe that $\mathrm{GL}(n, \mathbb{F})$ itself is a matrix group, the *general linear group of degree n over* $\mathbb{F}$. Matrix groups, fields, and rings are *algebraic structures*: sets equipped with algebraic operations (the set is closed under each operation) that satisfy given conditions. Modern algebra studies these structures in depth.

3 The advantages of matrix groups

Since matrix groups model transformations, they appear in science (physics, chemistry, biology), and throughout mathematics: in number theory, geometry, topology, differential equations; and, of course, in algebra, where matrix groups are used to represent other algebraic structures.

Representation of a mathematical object by matrices makes it more amenable to study. The theory of matrix groups began in the late 19th century; we see origins in work [6] by the French mathematician C. Jordan (1838–1922).[2] It is now a highly developed part of algebra.

Matrix groups also turn out to be a convenient format for handling by computer. Here we remark that a large (even infinite!) algebraic structure can correspond to input of much smaller size – as the following example shows.

Example 2. The set of all $n \times n$ matrices over $\mathbb{F}$ with determinant 1 is a matrix group, denoted $\mathrm{SL}(n, \mathbb{F})$. The abbreviation SL stands for 'special linear'. Note that $\mathrm{SL}(n, \mathbb{F})$ is finite if and only if $\mathbb{F}$ is finite.

Take $n = 5$ and $\mathbb{F} = \mathbb{Z}$. Although it is infinite, we can input $\mathrm{SL}(5, \mathbb{Z})$ to a computer as 50 bits. Explicitly, we only need to input the two matrices

$$a = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

in $\mathrm{SL}(5, \mathbb{Z})$, since a and b generate *all* of $\mathrm{SL}(5, \mathbb{Z})$ using the algebraic operations on matrices: see [7, Theorem VII.3] (here we have switched to lower case letters for group elements, as is customary). If we suppose that a and b are defined over $\mathbb{F} = \mathbb{F}_2$, then again we get all of $\mathrm{SL}(5, \mathbb{F})$ by taking products of these matrices and their inverses. The size of $\mathrm{SL}(5, \mathbb{F}_2)$ is 9999360, about ten million.

There is a tremendous amount of research aimed at the design of practical algorithms for computing with matrix groups over finite fields. Our concern

[2] http://www-groups.dcs.st-and.ac.uk/history/Biographies/Jordan.html

is matrix groups over an infinite field $\mathbb{F}$ (say, $\mathbb{Q}$ or $\mathbb{R}$). These are the matrix groups that are most prevalent, with applications in crystallography, error-correcting codes, theoretical physics, and elsewhere. They also pose the strongest computational challenges, which we examine below.

4 Computing with matrix groups

The increasing power and capabilities of computers have initiated a fresh burst of activity in the classical subject of matrix groups. This development falls under the heading of *computational group theory* (CGT for short), in turn a part of *computational algebra*. These are exciting and growing areas situated at the interface between mathematics and computer science.

Several requirements must be fulfilled in order to compute with a group. First, we have to represent the group in a computer: we need to input the group. Second, we must supply the computer with an algorithm to solve each assigned problem. Third, we should be able to gauge how good an algorithm is – how fast a computer will complete all steps in the algorithm, depending on the input (say, depending on how 'big' the input is), and how much memory we expect a computation to consume. Finally, we require a suitable environment in which to write our algorithms. This will be provided by a *computer algebra system*, containing all the ingredients needed for our computations.

In particular, to compute with matrix groups, the system will allow us to define objects and calculate in $\mathrm{GL}(n, \mathbb{F})$ for various $\mathbb{F}$, choosing from a menu of functions (that may have been contributed by many authors). The system also incorporates a programming language for implementation. Algorithms are written as programs in the system's language, understandable by any computer on which the system has been installed. Currently the two main systems for CGT are GAP [4] and MAGMA [1].

5 How to input a matrix group

If the matrix group G is infinite, how can it be input? Obviously we cannot feed all elements of G into the computer, one at a time. However, it might be possible to designate G by a finite set $S = \{g_1, \ldots, g_r\}$ of its elements. What this means is that every element of G is expressible as a 'word in S', that is, a product $g_{i_1}^{m_1} \cdots g_{i_k}^{m_k}$ where $m_1, \ldots, m_k$ are integers. Such groups G are *finitely generated*, and can be input using the set S of generators g_i. This ploy may not work for an arbitrary group G in $\mathrm{GL}(n, \mathbb{F})$: not all matrix groups are finitely generated.

Example 3. It is known that $\mathrm{GL}(n,\mathbb{Q})$ is not finitely generated. On the other hand, $\mathrm{GL}(n,\mathbb{Z})$ and $\mathrm{SL}(n,\mathbb{Z})$ *are* finitely generated, as illustrated in Example 2.

Even if a matrix group is not finitely generated, it could still be defined by a finite set, say, a finite set of polynomials in the entries of the group elements. In this latter case we would be dealing with *(linear) algebraic groups.* For example, $\mathrm{GL}(n,\mathbb{Q})$ and $\mathrm{SL}(n,\mathbb{Q})$ are algebraic groups.

There are other issues related to inputting matrix groups. How do we specify the field $\mathbb{F}$ of the matrix entries? This is not difficult to manage if $\mathbb{F}=\mathbb{Q}$, but what about $\mathbb{F}=\mathbb{R}$? Computing with real numbers often involves *floating point representation*, which approximates the numbers by truncated fractions. In contrast, group-theoretical computer algebra systems use *symbolic* representations to get 'exact' solutions. Here finiteness of our input is crucial: a finite generating set of G defines G over a ring inside $\mathbb{F}$ that is itself finitely generated (as a ring). The next example demonstrates this fact.

Example 4. Each matrix in the group $G \subseteq \mathrm{GL}(2,\mathbb{Q})$ generated by

$$\begin{pmatrix} 1 & \frac{1}{2} \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 \\ \frac{1}{3} & 1 \end{pmatrix}$$

has all entries of the form $m/6^k$. The collection of such rational numbers is a ring, denoted $\frac{1}{6}\mathbb{Z}$, and we have $G \subseteq \mathrm{GL}(2,\frac{1}{6}\mathbb{Z})$. The ring $\frac{1}{6}\mathbb{Z}$ is finitely generated: its elements are obtained from repeated addition and multiplication with the single element $\frac{1}{6}$.

Example 4 suggests how to specify the ring of entries for our input matrix group G: replace the original field $\mathbb{F}$ by a smaller ring, determined by the entries of all matrices in the finite generating set of G. This approach has added bonuses, which we will say more about in Section 7.

6 Algorithms

Having decided how to input a matrix group, we proceed with the design of algorithms. Users will typically want to extract concrete information from an input matrix group. For example, they might want to identify it as being of a certain type, describe its internal composition, and so on. We would like to have a library of algorithms for matrix groups over infinite fields that matches the breadth and sophistication of the libraries for other classes of groups that are available in GAP and MAGMA.

Nevertheless, we keep in mind that not every computational problem has a solution. This does not reflect our inability to design an algorithm to solve the problem, but rather that no algorithm exists. Such problems are said to be *undecidable.*

In the next section we discuss two fundamental group-theoretical properties. One of these will supply us with a technique for computing, and the other will guide our overall strategy.

7 Matrix groups in more detail

Modular arithmetic cropped up in Section 2, when we talked about finite fields. For a fixed integer $m \geq 2$, this arithmetic partitions the infinite ring $\mathbb{Z}$ into m congruence classes modulo m (two integers a and b belong to the same congruence class if their difference is exactly divisible by m). We extend modular arithmetic to matrix groups. Suppose that $G \subseteq \mathrm{GL}(n, \mathbb{Q})$ has generating set $S = \{g_1, \ldots, g_r\} \subseteq \mathrm{GL}(n, \frac{1}{t}\mathbb{Z})$, and select a prime p that does not divide t. Replace each entry in each g_i by its residue modulo p (do the reduction on numerators and denominators). The matrix group with generating set consisting of all reduced g_is is defined over $\mathbb{F}_p$. Denote this (finite) matrix group $\varphi_p(G)$. Remarkably, for every element g of the (perhaps infinite) matrix group G, there will be a prime p such that $\varphi_p(g) \neq \mathbb{1}_n$ in $\mathrm{GL}(n, \mathbb{F}_p)$. Moreover, this is true for all but a finite set of primes p. We say that G is *finitely approximated* by the $\varphi_p(G)$.

Finite approximation was pioneered by the Russian mathematician A. Malcev (1909–1967).[3] This method captures enough of G for our purposes. It transfers computation over an infinite ring largely to the context of matrix groups over the finite field $\mathbb{F}_p$, for which established and efficient algorithms are available. We also avoid the unfortunate possibility that calculations cause the entries in matrices over $\mathbb{Q}$ to become enormous; in $\mathrm{GL}(n, \mathbb{F}_p)$ the size of matrix entries is limited by the size p of $\mathbb{F}_p$.

As a consequence, our computation splits into two parts: computing with the matrix group $\varphi_p(G)$ over $\mathbb{F}_p$, and computing with the elements of G that map to $\mathbb{1}_n$ under reduction modulo p. The subset of these elements in G is the *kernel*, denoted G_p.

Example 5. Let G be as in Example 4. If p is a prime greater than 3 then $\varphi_p(G) = \mathrm{SL}(2, \mathbb{F}_p)$, and G_p is all matrices in G of the form $\mathbb{1}_2 + px$ where x is a 2×2 matrix over $\mathbb{Q}$.

[3] http://www-groups.dcs.st-and.ac.uk/history/Biographies/Malcev.html

While matrix multiplication is not commutative, occasionally the order of multiplication in a matrix group is irrelevant. In that case, the group is called *abelian* (after the Norwegian mathematician N. H. Abel (1802–1829)[4]). A profound theorem due to the Belgian mathematician J. Tits (b. 1930)[5] states that each finitely generated matrix group over a field $\mathbb{F}$ is one of two very different kinds:

1. it can be built up from finite and abelian components;
2. it contains a *free* group (in which nontrivial relations – for example, $x^r = 1$ or $xy = yx$, signifying that an element has finite order, or that elements commute – do not hold).

This prompts a basic question, that drives our computational strategy: to which class (1. or 2.) does a given matrix group belong? In other words, we must 'decide the Tits alternative'.

8 How things work

On the way to the Tits alternative we meet a simpler preliminary question. If $\mathbb{F}$ is infinite then so too is $\mathrm{GL}(n, \mathbb{F})$, and vice versa. However, a matrix group $G \subseteq \mathrm{GL}(n, \mathbb{F})$ for infinite $\mathbb{F}$ could well be finite.

Example 6. A matrix in $\mathrm{GL}(n, \mathbb{Z})$ is *monomial* if each row and column contains exactly one non-zero entry, and that entry is 1 or -1. The set of all monomial matrices in $\mathrm{GL}(n, \mathbb{Z})$ is a finite group, of size $2^n \cdot n!$.

So, we should be able to solve the 'finiteness problem': given a finitely generated matrix group $G \subseteq \mathrm{GL}(n, \mathbb{F})$, $\mathbb{F}$ infinite, determine whether G is finite. We might attempt to do this by listing the elements of G. But if G really is infinite then this job will never terminate. If G is finite, but has a huge number of elements, then the job will take too long. More astute methods, harnessing the power of a computer, are needed.

To explain the method used in one of our algorithms to test whether G is finite (and compute the size of G if it happens to be finite), we cite a result proved more than a century ago by the German mathematician H. Minkowski (1864–1909).[6]

[4] http://www-history.mcs.st-and.ac.uk/Biographies/Abel.html

[5] http://www-history.mcs.st-andrews.ac.uk/Biographies/Tits.html

[6] http://www-history.mcs.st-andrews.ac.uk/Biographies/Minkowski.html

Theorem 1 (Minkowski). Suppose that $G \subseteq \mathrm{GL}(n, \mathbb{Z})$, and g is a non-identity element of the kernel G_p where p is an odd prime. Then $g^m \neq \mathbb{1}_n$ for all positive integers m.

Minkowski's theorem yields a finiteness test: G is infinite if and only if it has an element $g \neq \mathbb{1}_n$ in G_p for $p = 3$ (say). Detecting such a g is the main computational part of the test.

If G is finite then φ_p is a one-to-one map on G, and therefore G will have the same number of elements as $\varphi_p(G) \subseteq \mathrm{GL}(n, \mathbb{F}_p)$. We can find this number not by a brute force count, but by using the established algorithms for matrix groups over finite fields.

9 What else can we do?

The problems (and their solutions) discussed above constitute merely a sample of the many things that we are now able to do computationally with matrix groups. We round out our introduction to this area by touching on the *orbit-stabilizer problem.* Algorithms to solve this problem are vital in CGT, as they can be adapted to solve other problems in disparate settings.

The central concept is a matrix group $G \subseteq \mathrm{GL}(n, \mathbb{F})$ *acting* on a vector space V of dimension n over the field $\mathbb{F}$: according to a defined *action*, G maps vectors in V to vectors in V. The most natural action is via matrix multiplication, that is, g in G multiplies on the left or right of an $n \times 1$ or $1 \times n$ vector in V respectively. Now, take two vectors u and v in V. Is there an element g of G that maps u to v, under the action being considered? If so, u and v are said to be in the same *orbit.* Note that there could be more than one g mapping u to v. When $u = v$, the set of such g is the *stabilizer* of u, and it is a matrix group.

Example 7. Let $\mathrm{SL}(2, \mathbb{Z}) \subseteq \mathrm{GL}(2, \mathbb{Q})$ act via ordinary matrix multiplication on the 2-dimensional vector space over $\mathbb{Q}$. The stabilizer of $(1, 1)^\top$ in $\mathrm{SL}(2, \mathbb{Z})$ consists of all matrices

$$\begin{pmatrix} 1+a & -a \\ a & 1-a \end{pmatrix}$$

where a ranges over $\mathbb{Z}$, hence is infinite.

We mention a more interesting action of $\mathrm{SL}(2, \mathbb{Z})$. Let $\mathbb{C}$ be the field of complex numbers: this is the field of all numbers $x + y\mathrm{i}$ where x, y are in $\mathbb{R}$, and $\mathrm{i}^2 = -1$. We view $\mathbb{C}$ as a 2-dimensional vector space over $\mathbb{R}$, and depict it as the x-y plane. Going by the name of *modular group*, $\mathrm{SL}(2, \mathbb{Z})$ acts on the set

$\mathcal{H}$ of complex numbers $x + y\mathrm{i}$ with $y > 0$ (the *upper half plane*) as follows:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} (x + y\mathrm{i}) = \frac{a(x + y\mathrm{i}) + b}{c(x + y\mathrm{i}) + d}\,.$$

The orbits of points on the unit semicircle centered at $(0, 0)$ are formed from the blue curves in Figure 1 (which was drawn by computer using an orbit-stabilizer algorithm!).

The orbit of any point in $\mathcal{H}$ intersects the shaded region; so all of $\mathcal{H}$ is obtained as images of this shaded region under the modular group action.

The stabilizer of any point is finite. In fact, unless the point is an image of i, $\frac{1}{2} + \frac{\sqrt{3}}{2}\mathrm{i}$, or $-\frac{1}{2} + \frac{\sqrt{3}}{2}\mathrm{i}$, its stabilizer is $\{\mathbb{1}_2, -\mathbb{1}_2\}$.

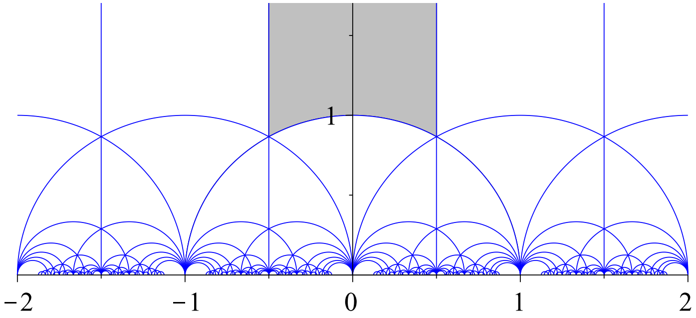

Figure 1: Orbits of the modular group acting on the upper half plane.

10 Afterword

Computer solutions of the problems highlighted in this snapshot have become possible only within the past few years (see [2] for a comprehensive survey written at the specialist level). The area will continue to expand, as other important problems await solution. For more general accounts of CGT, and computing with matrix groups over finite fields, see [3, 5, 8, 9]. For an exposition of classical theory of matrix groups, see [7].

Acknowledgments

Dr A. S. Detinko was funded by a Marie Skłodowska-Curie Individual Fellowship (Horizon 2020, European Commission). Prof. A. Hulpke was funded by Simons Foundation Collaboration grant 244502.

We are most grateful to Mathematisches Forschungsinstitut Oberwolfach for its hospitable support of our work.

Image credits

All images were created by the authors.

References

[1] W. Bosma, J. Cannon, and C. Playoust, *The Magma algebra system. I. The user language*, Journal of Symbolic Computation **24** (1997), 235–265.

[2] A. S. Detinko and D. L. Flannery, *Linear groups and computation*, Expositiones Mathematicae (2018), https://doi.org/10.1016/j.exmath.2018.07.002.

[3] B. Eick, *Computational group theory*, Jahresbericht der Deutschen Mathematiker-Vereinigung **107** (2005), 155–170.

[4] The GAP Group, *GAP – Groups, Algorithms, and Programming*, www.gapsystem.org, 2013.

[5] D. F. Holt, B. Eick, and E. A. O'Brien, *Handbook of Computational Group Theory*, Discrete Mathematics and its Applications (Boca Raton), Chapman & Hall/CRC Press, 2005.

[6] C. Jordan, *Traité des substitutions et des équations algébriques*, Les Grands Classiques Gauthier-Villars, Éditions Jacques Gabay, 1989.

[7] M. Newman, *Integral Matrices*, Academic Press, 1972.

[8] E. A. O'Brien, *Towards effective algorithms for linear groups*, Finite Geometries, Groups, and Computation, Walter de Gruyter, 2006, pp. 163–190.

[9] Á. Seress, *An introduction to computational group theory*, Notices of the American Mathematical Society **44** (1997), 671–679.

A. S. Detinko *is Marie Curie Fellow in the School of Computer Science at University of St Andrews.*

D. L. Flannery *is Professor in the School of Mathematics, Statistics and Applied Mathematics at National University of Ireland, Galway.*

A. Hulpke *is Professor in the Department of Mathematics at Colorado State University.*

Mathematical subjects
Algebra and Number Theory

Connections to other fields
Computer Science

License
Creative Commons BY-SA 4.0

DOI
10.14760/SNAP-2019-005-EN

Snapshots of modern mathematics from Oberwolfach provide exciting insights into current mathematical research. They are written by participants in the scientific program of the Mathematisches Forschungsinstitut Oberwolfach (MFO). The snapshot project is designed to promote the understanding and appreciation of modern mathematics and mathematical research in the interested public worldwide. All snapshots are published in cooperation with the IMAGINARY platform and can be found on www.imaginary.org/snapshots and on www.mfo.de/snapshots.

ISSN 2626-1995

Mathematisches Forschungsinstitut
Oberwolfach gGmbH
Schwarzwaldstr. 9 – 11
77709 Oberwolfach
Germany

Director
Gerhard Huisken