

# Groups & Computation: A Brief History

Peter Brooksbank

Department of Mathematics

The Ohio State University

Columbus, OH 43210

`brooksbank@math.ohio-state.edu`

[www.math.ohio-state.edu/~brooksbank](http://www.math.ohio-state.edu/~brooksbank)

May 7, 2004

# Outline

- **Origins of Computational Group Theory (CGT)**
  - “An Introduction to CGT” Ákos Seress  
Notices of the AMS, June/July 1997.
  - “An Invitation to CGT” Joachim Neubüser  
Groups St Andrews/Galway, 1993.
- **Permutation Group Algorithms**
  - Examples of Problems
  - Methods and Constructions
- **Matrix Group Algorithms**
  - Obstacles
  - Strategies
  - Recent Progress

## Prehistory

- 1830 Solubility of polynomials by radicals. (Galois)
- 1860 Discovery of first sporadic simple groups. (Mathieu)
- 1911 **Word Problem** for finitely presented groups formulated. (Dehn)
- 1936 First systematic approach to deciding finiteness of a finitely presented group using **coset enumeration**. (Todd & Coxeter)
- 1951 Suggested use of computational and probabilistic methods to investigate groups of order 256. (Newman)

## Early History

- 1953** Partial implementation of the Todd-Coxeter algorithm on EDSAC II in Cambridge. (Haselgrove)  
Calculation of characters of symmetric groups on BARK in Stockholm. (Comet)
- 1959** Calculation of subgroup lattice of permutation groups. (Neubüser)
- 1967** Oxford conference:  
“Computational Problems in Abstract Algebra”.

# Decade of Discovery

## METHODS

- 1970 – Methods for handling large permutation groups. (Sims)
- Rewrite Systems for f.p. groups. (Knuth-Bendix)
- 1974 –  $p$ -Nilpotent-Quotient method. (Macdonald)
- Reidemeister-Schreier method. (Havas)

## APPLICATIONS

- 1973 Existence proof of Lyons' sporadic simple group. (Sims)
- 1974 Determination of the Burnside group  $B(4, 4)$  of order  $2^{422}$ .  
(Newman & Havas)

## SYSTEMS

- 1974 Aachen-Sydney Group System operational.
- 1976 Description of group theory language Cayley. (Cannon)

# Modern Times

- Existence of the **Baby Monster** of order  
4, 154, 781, 481, 226, 426, 191, 177, 580, 544, 000, 000  
as a permutation group of degree 13,571,955,000. (Sims)
- Computational techniques used to help make and verify the **Atlas of Finite Groups**.
- Classification of the 58,760 isomorphism classes of groups of order  $2^n$ ,  $n \leq 8$ . (O'Brien)
- Development of the group theory systems **GAP** and **MAGMA**.
- Development of polynomial-time theory for permutation groups.
- Improved methods in computational representation theory.
- Progress in matrix group algorithms.

# Permutation Group Algorithms

We assume that permutation groups are input concisely by means of a generating set of permutations of the set  $\{1, \dots, n\}$ . We write

$$G = \langle X \rangle \leq S_n, \text{ for given } X \subset S_n.$$

Extremely large groups may be input using small generating sets.

It is therefore natural to ask:

For any given set  $X \subset S_n$ , what properties of the group  $G = \langle X \rangle$  may be determined using “efficient” algorithms?

“Efficiency” may have a variety of meanings, but is measured in terms of the **input length**: in this case, as a function of  $|X|n$ .

# Complexity

In order to put efficiency issues for computational problems on a firm theoretical footing, we define the following two classes:

**P** a problem is in **P** if there is a deterministic algorithm that solves the problem, for any given input  $I$ , in time polynomial in the input length,  $|I|$ .

**NP** a problem is in **NP** if there is a deterministic, polynomial-time algorithm that will **verify** the correctness of any purported solution to the problem.

The Million Dollar Question

$P \neq NP ?$

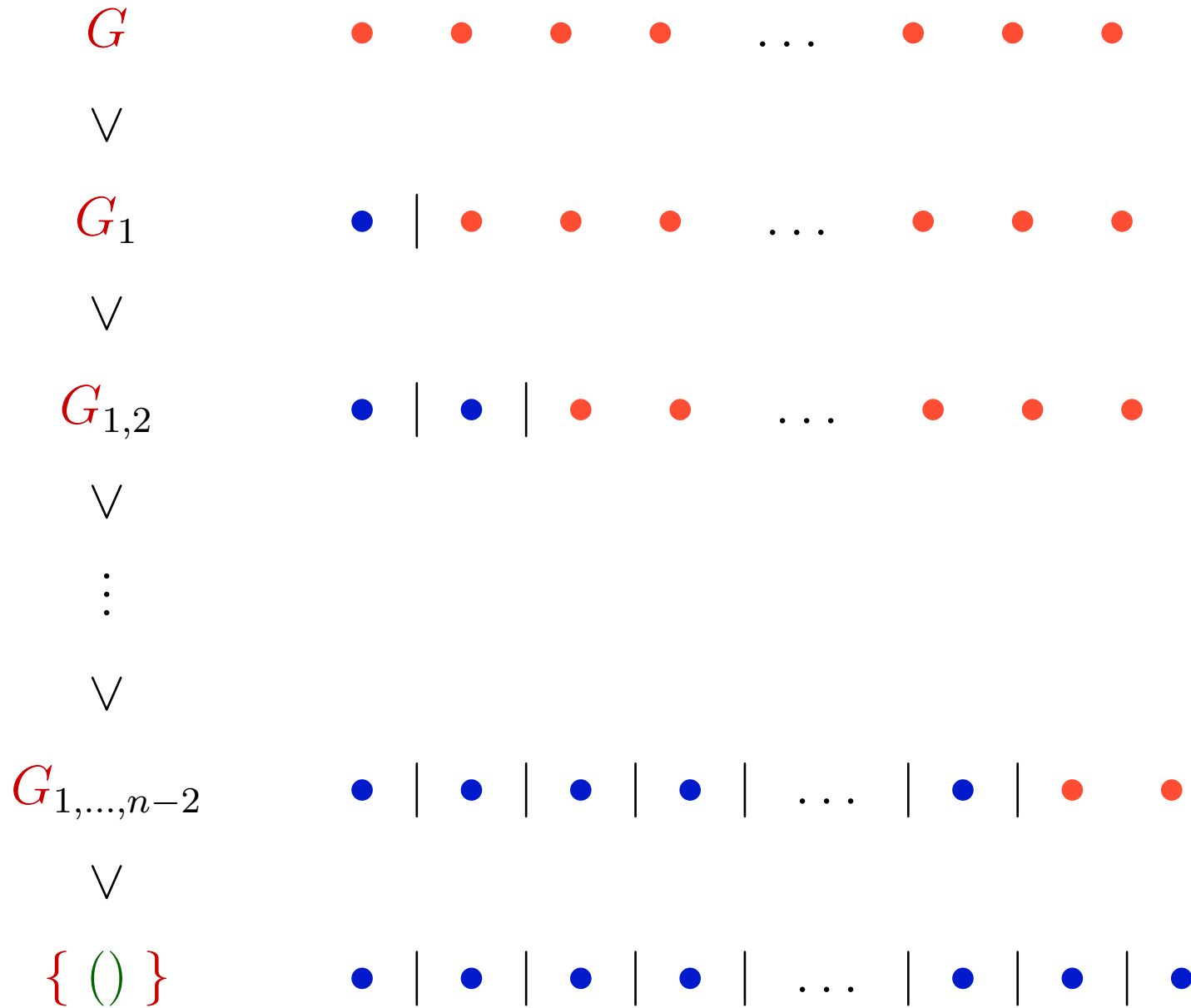
# Computational Problems

Let  $G = \langle X \rangle \leq S_n$  be a given permutation group.

1. Find  $|G|$ .
2. Given  $x \in S_n$ , decide whether  $x \in G$ .
3. Given  $\alpha \in \{1, \dots, n\}$ , find (gens for)  $G_\alpha = \{g \in G \mid \alpha^g = \alpha\}$ .  
[Problems 1-3 are in P: Furst, Hopcroft, Luks (1980).]
4. Construct a composition series for  $G$ . [In P: Luks (1987)]
5. Given  $H = \langle Y \rangle \leq S_n$ , find (gens for)  $G \cap H$ .
6. Given  $\Delta \subset \{1, \dots, n\}$ , find  $G_\Delta = \{g \in G \mid \Delta^g = \Delta\}$ .

[The last two problems have connections with the famous Graph Isomorphism Problem, one of the leading candidates for a problem in a complexity class lying in between P and NP.]

# Point-Stabilizer Chain

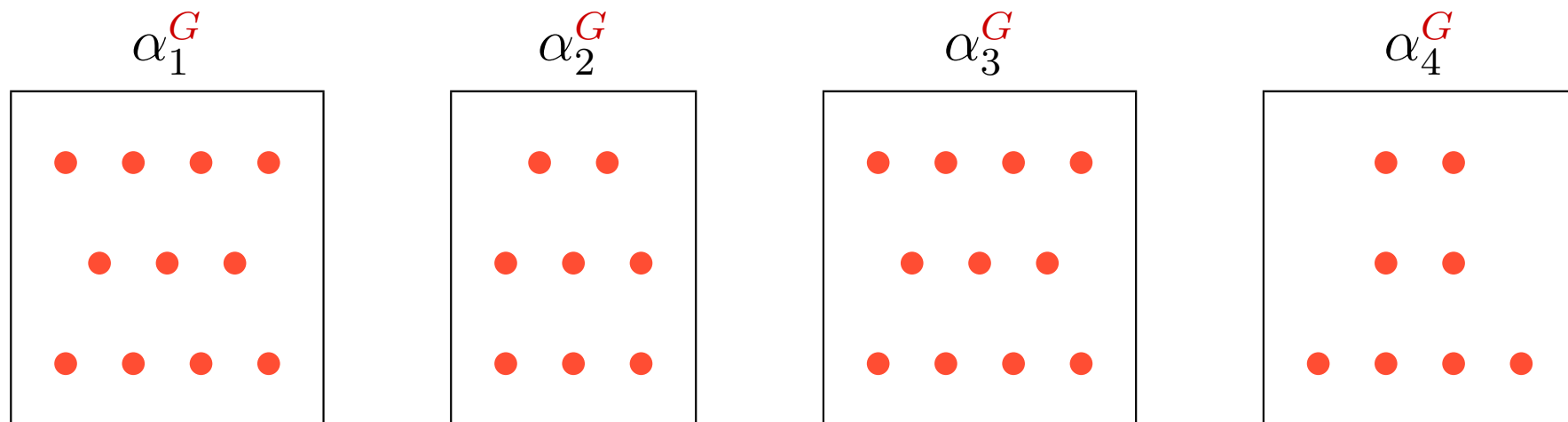


# Orbits and Transitivity

For  $\alpha \in \{1, \dots, n\}$ , define the orbit of  $G$  containing  $\alpha$  to be

$$\alpha^G = \{\beta \mid \beta = \alpha^g \text{ for some } g \in G\}.$$

We obtain a partition of  $\{1, \dots, n\}$  into  $G$ -orbits:



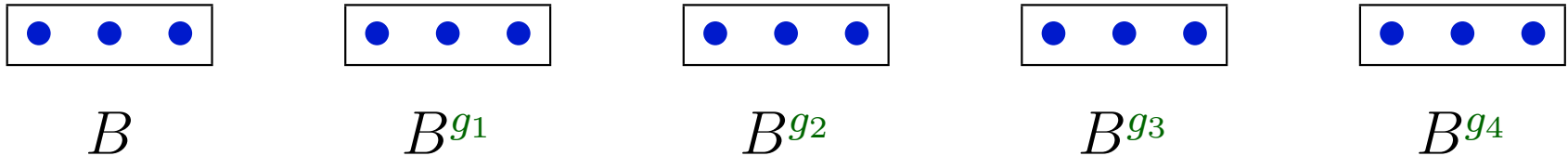
If  $G$  has only one orbit, then it is said to act **transitively**.

## Blocks and Primitivity

Suppose  $G$  is transitive on  $\{1, \dots, n\}$ . A subset  $B \subseteq \{1, \dots, n\}$  is called a **block** for  $G$  if, for all  $g \in G$ , either  $B^g = B$  or  $B \cap B^g = \emptyset$ .

If a block  $B$  is nontrivial (i.e.  $1 < |B| < n$ ), the translates of  $B$  under the action of  $G$  give a **block system** upon which  $G$  acts:

$G$



If  $G$  has no nontrivial blocks, it is said to act **primitively**.

# Divide and Conquer

Suppose we wish to find a composition series of a given permutation group  $G$ . A good starting point is to find a **proper normal subgroup**:

1. Compute the action of  $G$  induced on each of its orbits.
  - If such an action has a nontrivial kernel, we're done.
  - If not, we have a faithful, transitive action of  $G$  on the orbit.
2. For each of the transitive actions in step 1, find a minimal block system, as well as the action of  $G$  induced on this system.
  - If such an action has a nontrivial kernel, we're done.
  - If not, we have a faithful, primitive action of  $G$  on the block system.
3. If we failed to find a proper normal subgroup in steps 1 and 2, then we may now assume that  $G$  is acting primitively.

# The O’Nan-Scott Theorem

This celebrated theorem classifies primitive permutation groups in terms of their socles. (The **socle** of a group is the subgroup generated by all of its minimal normal subgroups.)

Simplifying matters a great deal, if  $G$  acts primitively, and  $H$  is its socle, then we have the following dichotomy:

- $H$  is an elementary abelian  $p$ -group, and  $G$  is a subgroup of  $AGL(n, p)$ ; or
- $H = T_1 \times \dots \times T_k$  is a direct product of isomorphic nonabelian simple groups, whose action is explicitly described.

Within our divide-and-conquer mechanism for finding a proper normal subgroup, the classification is used to construct actions that are either unfaithful (have a nontrivial kernel), or involve sets of smaller size.

## Matrix Group Algorithms

We assume that matrix groups are input by means of a generating set of invertible matrices with entries in a finite field. We write

$$G = \langle X \rangle \leq GL(d, q).$$

Each generating matrix has input length  $d^2 \log q$ .

(Roughly  $\log q$  bits are required for each of the  $d^2$  matrix entries).

The input length of a matrix group is therefore  $|X|d^2 \log q$ .

# Principal Difficulties

Matrix groups are very concisely represented.

- If one attempts to regard the matrix group  $G \leq GL(d, q)$  instead as a permutation group on its nonzero vectors, the degree would be roughly  $q^d$ , whereas its original input length is roughly  $d^2 \log q$ : there would be an exponential blow-up.
- Unlike the permutation group case, where one is always able efficiently to construct a proper subgroup of index at most  $n$  (for example, a point-stabilizer subgroup), it may be the case that a given matrix doesn't possess a proper subgroup of polynomial index.

# The Aschbacher Classification

In a celebrated theorem, Aschbacher proves that each maximal subgroup of  $GL(d, q)$  belongs to at least one of a number of explicitly defined classes. Roughly speaking we have the following dichotomy:

- (Geometric Case) The maximal subgroup is the normalizer of some geometric object.
- (Nearly Simple Case) The maximal subgroup,  $M$ , is almost simple modulo scalars: that is, if  $\overline{M}$  denotes  $M$  modulo scalars, then there exists a simple group  $T$  with  $T \leq \overline{M} \leq \text{Aut}(T)$ .

# Divide and Conquer Based on Aschbacher

An international project has been underway for some time, whose goal is to construct a “composition tree” for any given  $G \leq GL(d, q)$ . In principle it should proceed somewhat as follows:

- If  $G$  is geometric:
  1. Determine the Aschbacher category of  $G$ .
  2. Find a geometric structure preserved by  $G$ .
  3. Compute the group  $G_1$  induced by the action of  $G$  on this structure, as well as the kernel,  $G_0$ , of this action.
  4. Recursively compute a composition tree for  $G_0$  and  $G_1$ .
- If  $G$  is nearly simple:

Solve the “recognition problem” for  $G$ .

# The Geometric Classes

Let  $G = \langle X \rangle \leq GL(d, q)$  be a given matrix group that is not nearly simple. Then one of the following holds:

1.  $G$  is reducible. (Holt & Rees)
2.  $G$  is semilinear over an extension field. (Holt et al)
3.  $G$  may be represented over a subfield. (Glasby & Howlett)
4.  $G$  is imprimitive. (Holt et al)
5.  $G$  is tensor factorisable. (Leedham-Green & O'Brien)
6.  $G$  is tensor induced. (Leedham-Green & O'Brien)
7.  $G$  normalises a symplectic-type  $p$ -group. (B, Niemeyer, Seress)
8.  $G$  normalises a classical group in its natural representation. (B)

## The Nearly Simple Case

The “leaves” of the composition tree correspond, more or less, to the composition factors of the matrix group. Here we need to be able to solve the

**Recognition Problem:** Let  $G = \langle X \rangle$  be a given simple group.

- **(Nonconstructive Recognition):** Determine the standard “name” (e.g. “ $A_7$ ” or “ $PSL(2, 11)$ ”) of the simple group to which  $G$  is isomorphic.
- **(Constructive Recognition):** Construct an explicit isomorphism  $\Psi: T \rightarrow G$ , where  $T$  is the standard copy of the simple group.

# The State of Play

- O'Brien and Leedham-Green have the architecture for the divide-and-conquer mechanism in place in MAGMA.
- An alternative to their “geometric” approach is currently being developed by Seress in GAP.

The following are important tasks for the immediate future:

1. Good implementations, and new ideas, for certain important special cases of constructive recognition (e.g. the **natural representation**; B (2003)).
2. Implementations of constructive recognition algorithms for **black box** simple groups (Kantor-Seress (2001), B (2003), B-Kantor (2004)).
3. An implementation of an algorithm to recognize symplectic-type groups (B, Niemeyer, Seress (2004)).