GAP – Groups, Algorithms, Programming
a System for Computational Discrete Algebra

GAP is a system for computational discrete algebra, with particular emphasis on Computational Group Theory.
GAP is used in research and teaching for studying groups and their representations, rings, vector spaces, algebras, combinatorial structures, and more.
GAP is developed by international cooperation. The system, including source, is distributed freely under the terms of the GNU General Public License. You can study and easily modify or extend GAP for your special use.
The current version is GAP 4, the older version GAP 3 is still available. See the GAP website

http://www.gap-system.org/

for many more details than can be given here.

1 Overview of GAP

GAP provides:

- Mathematical capabilities accessible through
  - a large library of functions (divided into 'Modules' under the responsibility of 'maintainers'), containing implementations of various algebraic algorithms,
  - presently (2005) nearly 50 separate 'Packages' of additional functions for specialized purposes which can be used like library functions, but remain under the responsibility of their authors,
  - data libraries containing large classes of various algebraic objects that are accessible by using GAP commands.
- A programming language, also called GAP, which is interpreted and can be compiled. It can be used interactively at the keyboard or to write programs to be saved and then executed. Such programs can easily be modified and rerun. The language features:
  - Pascal-like control structures,
  - automatic memory management including garbage collection,
  - built-in data types for key algebraic objects,
  - flexible list and record data types,
  - furthermore a mechanism for automatically choosing the highest ranked method for a certain operation, depending on the current state of all its arguments, so that GAP objects representing mathematical objects may gain knowledge about themselves during their lifetime resulting in better methods being chosen later on.
- An interactive environment that provides on-line help (i.e. on-line access to the manuals) and supports debugging and timing of GAP programs.
- Documentation, in particular
  - download and installation instructions.
  - a Tutorial and further learning and teaching material,

– the Reference Manual giving descriptions of library functions with examples of use,
– separate manuals for the packages,
– worked out higher level examples and a collection of preprints and talks,
– FAQ answers,
– advice for people writing GAP code,
– a mark-up language GAPDoc for writing GAP documentation,
– an archive of GAP Forum contributions,
– a Bibliography of presently (2005) over 700 papers quoting GAP.

2 Contacts and Cooperation

A central point in the philosophy of GAP is to further the cooperation of developers and users. To this end we provide:

- the GAP Forum for discussions and announcements of general interest to all users of GAP, which are kept in a Forum archive,
- an address `support@gap-system.org` to which users can send questions, complaints or suggestions of more local interest that will be answered by a GAP support team,
- the possibility to have new GAP packages refereed in a way similar to the refereeing of a paper submitted to a journal for publication (an international GAP Council serves as editorial board and gives general advice for the development of GAP).

We ask to be informed about downloading GAP, interesting use of GAP, and publications quoting use of GAP, that we collect in the bibliography mentioned above.

3 Structure of the GAP system

GAP has a kernel written in C. It implements the GAP language, the interactive environment for developing and using GAP programs, memory management, and fast versions of time critical operations for various data types.
All the rest of the library of functions is written in the GAP language. Packages are mainly written in the GAP language, but some also involve standalones. Some packages provide links to other systems.

4 Requirements and Availability

GAP can be installed on computers with UNIX/Linux, Windows or Macintosh operating systems (users of Mac OS X may consider a UNIX installation). The computer must have about 500 MB of free disc space and it should have at least 128 MB of main memory for smooth operation.
The current version is GAP 4, but GAP 3 is also available, so those GAP 3 packages that have not yet been converted to GAP 4 may still be used. Both GAP 4 and GAP 3 may be obtained at no cost by following the download instructions.

5 Mathematical Capabilities of GAP

The following description covers capabilities of the main GAP library, packages, and data libraries.

5.1 Basic Capabilities

GAP provides many ways of forming

- lists, which also covers sets, row vectors, matrices, strings, and Boolean lists,
- records,
- domains, i. e., objects representing sets with an algebraic structure.

Notions and tools, modeled in some analogy to elementary set theory, help to handle these: distinction of objects and elements, Booleans, orderings, mappings (in particular mappings respecting structures which again can be elements of structures), and relations.
GAP can compute with

- arbitrary integers,
- rational numbers,
- cyclotomic numbers, in particular Gaussian numbers,
- elements of finite fields,
- p-adic numbers (in a rudimentary way),
- polynomials, including multivariate polynomials (however GAP has only a basic implementation of Buchberger's algorithm - an interface to SINGULAR is provided),
- rational functions as well as
- various kinds of group elements, e. g. permutations, matrices, and abstract words.

One can work with many algebraic structures. In addition to those dealt with in separate sections below there are e. g.

- finite fields,
- residue class rings of integers,
- abelian number fields,
- lattices and integral matrices.

Also there are

- various combinatorial functions,
- functions for elementary number theory as well as
- functions for prime number factorization.

5.2 Groups and Group Elements

Groups can be given in various forms: for example as permutation groups or matrix groups (by generating elements), as finitely presented groups or as polycyclicly presented groups. GAP knows how to construct a number of well-known groups such as symmetric and classical groups and to fetch concrete groups from group libraries that are part of GAP.
There is a wide variety of functions for the investigation of groups. Some of these functions just build on the concept of a group while others (usually the more efficient ones, for instance nearly linear methods for permutation groups) utilize the way in which a particular group is given. GAP automatically tries to select a good method, but the user can take over full control of this selection of methods. Also, if no deterministic method exists (e. g., for determining the order of a finitely presented group) GAP will try

to find an isomorphism to a group it can handle (in the above case it will try to find an isomorphism to a permutation group using the Todd-Coxeter method).
There are many functions to compute invariants of groups, e. g.:

- order (called “size” in GAP),
- conjugacy classes of elements,
- derived series,
- composition series (including identification of the composition factors),
- Sylow subgroups,
- certain characteristic subgroups,
- maximal subgroups,
- normal subgroups,
- subgroup lattice,
- table of marks,
- automorphism group,
- cohomology groups,
- ordinary character table.

There are also functions for

- group homomorphisms,
- group actions,
- group products, and
- group constructions.

Of course the range of applicability of the particular functions depends very much on the order and structure of the group. To give an idea of capabilities, GAP has (already in 1993) been used to find the composition series, Sylow subgroups and character table of a certain solvable subgroup of order 3,265,173,504 in the sporadic simple group Fi23, given as a permutation group of degree 31,671.

5.3 Permutation and Matrix Groups

For an overview of computational methods for permutation groups see the book by Ákos Seress.
The Schreier-Sims method is the basis of many functions implemented in GAP. Nearly linear time methods for permutation groups include functions to compute

- a stabilizer chain,
- p-core,
- radical,
- centre,
- composition series.

There are also tasks for which no polynomial time methods are known and for which GAP relies on partition backtrack methods, for example

- centralizer,
- normalizer, or
- intersection of subgroups.

The package MATRIXSS implements Schreier-Sims methods for matrix groups.
The package POLENTA allows to find polycyclic presentations for matrix groups.
Routines to recognize isomorphism types of matrix groups using the Aschbacher classification (which recently have been in the focus of attention in computational group theory) are still (2005) only available through the GAP 3 package MATRIX. Also the package CHEVIE for dealing with groups of Lie type and Chevalley groups still (2005) only exists with GAP 3.

5.4 Finitely Presented Groups

It follows from the well known theorems on the algorithmic unsolvability of the word problem and related problems that there are no deterministic methods to answer most questions about the structure of finitely presented groups. For these essentially two approaches are possible: On the one hand trial-and-error methods such as the Todd-Coxeter and Knuth-Bendix methods, on the other hand methods to find certain factor groups of described structure, most of which are based on the idea of "collection".
The main GAP library provides methods for handling finitely presented groups such as

- Todd-Coxeter,
- Reidemeister-Schreier,
- Low Index Subgroups, as well as
- certain quotient methods.

Also there are functions to handle Tietze transformations.
In addition there are packages

- FGA providing algorithms for free groups,
- ACE linking to particular powerful implementations of the Todd-Coxeter method,
- ITC providing an interactive Todd-Coxeter mainly for teaching and learning,
- ANUPQ for finding p-quotients of fp groups, standard presentations and descendants of p-groups,
- NQ for finding infinite nilpotent quotients, and
- KBMAG with Knuth-Bendix and automatic groups methods.

The packages Double Coset Enumerator, Vector Enumerator, ANU Soluble Quotient, and Polycyclic Quotient are still (2005) only available through GAP 3.

5.5 Polycyclic Groups

Finite soluble groups and infinite polycyclic groups have special generating systems chosen in accordance with a subnormal series with cyclic factors. These allow highly efficient computation in such groups using collection of words in these generators. The main GAP library contains a wide variety of functions utilizing these methods and being able to compute efficiently

- conjugacy classes,
- centralizers,
- normalizers,
- intersections and complements of subgroups as well as
- cohomology groups.

The two packages FORMAT and CRISP give access to the highly developed theory of the subgroup structure of finite soluble groups related to notions such as formations, Schunck classes, and homomorphs. The package POLYCYCLIC extends the possibility of detailed structure analysis to infinite polycyclic groups allowing to compute e. g. Hirsch length and torsion subgroup. It utilizes an interface to the KANT / KASH system for algebraic number theory.

5.6 Representations and Characters of Groups

Group representations over fields of characteristic zero are mainly investigated via their characters. GAP provides methods for computing the irreducible characters of a given finite group, either automatically or interactively by character theoretic means. It also provides many functions for deducing group theoretic properties from character tables.
Efficient methods are available for special classes of groups, e. g. in the GAP 3 share package CHEVIE which allows one to work with generic characters of Hecke algebras and groups of Lie type (using a link to the Maple system).
Modular representations (i. e., over fields whose characteristic divides the group order) can be studied via Brauer characters or by explicit calculations with matrices representing the generators of the group in question, using MeatAxe methods, Vector Enumeration (in GAP 3), and condensation techniques.
Computing (irreducible) representations themselves (which are of interest e.g. for applications to signal processing) is possible using the GAP 3 package AREP (for not too large groups).
Several GAP data libraries (in particular CTBLLIB, TOMLIB and ATLASREP) are related to representations and characters.

5.7 Graphs, Codes, and Designs

The package GRAPE allows one to construct graphs (with or without using groups), to determine many graph invariants, to classify complete subgraphs with various properties, to determine automorphism groups of graphs, and to test graph isomorphism (the last two via an interface to B.D. McKay's nauty package).
The package GUAVA can construct many classes of codes, derive invariants such as minimal distance and weight distribution and find their automorphism groups.
The package DESIGN provides classification, partitioning, and study of block designs, including the determination of automorphism groups and the testing of isomorphism.

5.8 Vector Spaces, Modules and Algebras

Vector spaces over fields and modules over rings can be defined when the coefficient domain is available in GAP.
There are algorithms for the efficient calculation of Hermite and Smith normal forms over the integers.
Computations concerning special modules arising in representation theory are possible. However the package SPECHT for dealing with Specht modules is still (2005) only available in GAP 3.
Lie algebras can be given by structure constants, by generating matrices or by a finite presentation. There are routines for computing the structure of finite dimensional Lie algebras, in particular Cartan subalgebras, the direct sum decomposition, a Levi decomposition, the solvable radical and nil radicals.
The package SOPHUS deals with nilpotent Lie algebras over prime fields allowing to construct central extensions and to automorphism groups.

Much of the support for Lie algebras is based on more general methods using an implementation of the arithmetic operations via structure constants, which works for any finite dimensional algebra. In particular, associative algebras (e. g., group rings, cf the manual chapter 'Magma Rings') are also supported. For example, the package LAGUNA allows to investigate unit groups of the modular group ring of a p-group and Lie algebras associated with associative algebras.
Presentations are currently restricted to Lie algebras (if one uses the package FPLSA). There is a package QUAGROUP to work with Quantum Groups.

5.9 Semigroups, Monoids, and other Generalisations of Groups

These include

- functions for calculating with transformations,
- functions for investigating semigroups, in particular transformation semigroups, monoids, and finitely presented semigroups and monoids,
- some special algorithms for commutative semigroups,
- sets of basic functions for magmas and additive magmas,
- a package SONATA for investigating small semigroups and near rings,
- a package XMOD for crossed modules and cat-1 groups,
- a (forthcoming) package GRAPHGPD for computation of finite groupoids,
- a (forthcoming) package LOOPS for various types of loops.

5.10 Words, Rewriting, and Automata

These include

- functions for calculating with words, in particular with associative words,
- functions dealing with rewriting systems,
- the package KBMAG providing the Knuth-Bendix method on Monoids and functions for Automatic Groups,
- the package AUTOMATA allowing to generate finite state automata and investigate its states.

5.11 Further Capabilities

These include

- functions for the determination of Galois groups of rational polynomials,
- packages CRYST and CARAT for crystallographic groups,
- a package PARGAP for parallel computations with GAP,
- a graphical user interface XGAP that permits, for example, the display of subgroup lattices.

5.12 Interfaces to Other Systems

- to the KANT / KASH system for algebraic number theory,
- to the SINGULAR system for algebraic geometry,
- to the program DISCRETA for the construction of t-designs with prescribed automorphism group.

5.13 Data Libraries

- The package Small Groups contains all groups of order up to 2000 (except 1024) and some infinite series of groups characterised by the prime factorisation of their orders,
- Basic Groups: Cyclic, abelian, dihedral, extraspecial, alternating, symmetric, Mathieu, Suzuki, and Ree groups,
- Classical Groups: Linear, unitary, symplectic, and orthogonal groups,
- Almost all perfect groups of order up to a million,
- All transitive Permutation Groups of degree up to 30,
- All primitive permutation groups of degree less than 256 and non-affine primitive permutation groups of degree less than 1000,
- Irreducible solvable subgroups of GL(n,q) for q^n up to 243,
- The package IRREDSOL containing a library of all irreducible solvable subgroups of GL(n,q) with q^n less than 2^{16},
- Irreducible maximal finite integral matrix groups of dimension up to 31,
- The package CRYSTCAT contains the Crystallographic groups up to dimension 4,
- The package ACLIB contains a library of Almost Crystallographic Groups as well as algorithms for these,
- The package TOMLIB: Tables of marks for many almost simple groups,
- The package CTBLLIB: The GAP Character Table Library, containing i.a. all tables of the 'Atlas of Finite Groups' as well as the 'Atlas of Brauer Tables',
- The package ATLASREP providing a GAP interface to the 'Atlas of Group Representations',
- Lie Algebras: Free, Full Linear, and simple Lie Algebras.

6 Contributors and Acknowledgements

GAP has been and is developed by international cooperation of many people, including user contributions. We gratefully acknowledge all this help as well as some funding. For some details on both see

http://www.gap-system.org/Contacts/People/people.html

For each person involved with GAP you can use the search facility for GAP from

http://www.gap-system.org/search.html

to find all places in the GAP documentation (website, manuals, Forum archive) in which the name of the person appears.
GAP was started at Lehrstuhl D für Mathematik, RWTH Aachen in 1986. After 1997 the development of GAP was coordinated in St Andrews, at present (February 2005) the GAP centers in Aachen, Braunschweig, Fort Collins and St Andrews have agreed to coordinate jointly the development and maintenance of GAP.