Groups, Algorithms, Programming and Free Software

Steve Linton

Centre for Interdisciplinary Research in Computational Algebra

University of St Andrews

Groups

- The mathematician's handle on symmetry
- Key objects in mathematics
- Applications in physics, chemistry, comp. sci.,...
- One of the first areas of pure mathematics to be computerised
- here finitely generated and discrete

“There will be positively no internal alterations to be made even if we wish suddenly to switch from calculating the energy levels of the neon atom to the enumeration of groups of order 720.”

Alan Turing (1945)

Algorithms – History

- Todd & Coxeter, 1936: “A Practical Method for Enumerating Cosets of a Finite Abstract Group”
  - formalised a type of computation going back to 1900
- 1950s – First digital computer experiments
- 1970s – First software systems: Aachen-Sydney Group System, Cayley, CAS, SOGOS, SPAS, Meataxe
  - Very many algorithms – permutation groups, fp groups, repn theory
- Since then – increasing sophistication in software and algorithms – GAP and MAGMA

Algorithms
Examples of the State of the Art 1

- Given a presentation we can
  - enumerate millions of cosets of a subgroup
    - giving a usable homorphism to a permutation group
  - search for a confluent rewriting system or a simpler presentation
  - find homomorphic images as
    - p-groups of composition length in the thousands
    - nilpotent groups of Hirch length in the dozens
    - permutation groups of degree in the dozens
    - many types of simple groups

Algorithms
Examples of the State of the Art 2

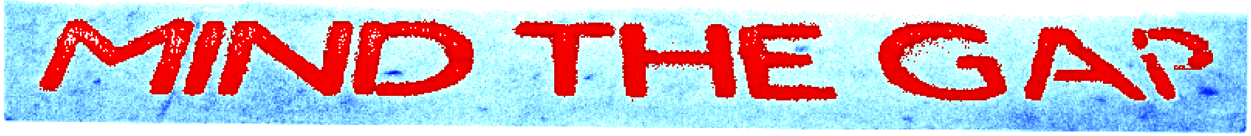
- We can compute freely with
  - Permutation groups on up to a million or so points
  - Finite polycyclicly presented groups with composition length in the thousands
  - Matrix groups and groups of automorphisms of reasonable size
- This means we can compute with elements, subgroups and homomorphisms:
  - centre, centralizer, normalizer, composition series, Sylow subgroups, conjugacy classes, image and kernel of mappings, automorphism group, coset reps, ...

Algorithms
Not Just Groups

- Modern group theoretic computations often need other things
  - polynomial arithmetic, Gröbner bases, etc.
  - Number theory – unit groups, etc.
  - algebras – Lie algebras for p-groups, associative algebras for representation theory
- Also the approaches and tools developed for group theory can be applied elsewhere
  - nearrings, semigroups and monoids
  - Lie algebras for their own sake

Programming About GAP

- Development began in Aachen, mid-80s
  - Neubüser, Schönert, others
- 1997, Neubüser retired – international project coordinated from St Andrews
- Free Software under GPL
- 165K lines of C, 380K lines of GAP, over 320MB of databases (including over 400 million groups), 2000+ operations, 12000+ methods, 30+ refereed contributed packages



GAP in Action

```
gap> AvgOrder :=
>  g->Sum(ConjugacyClasses(g),
>  c-> Size(c)*Order(Representative(c)))/
>   Size(g);
function( g ) ... end
gap> AvgOrder(MathieuGroup(11));
53131/7920
gap> ForAny(AllSmallGroups([2..100]),
> g->IsInt(AvgOrder(g)));
false
```

- Qn: is there a non-trivial group whose elements have integer average order?
- Define a function and run over some groups from the database
- Database includes polycyclically-presented and permutation groups
- Note generic operations like Size and ConjugacyClasses
- Partial answer at the end

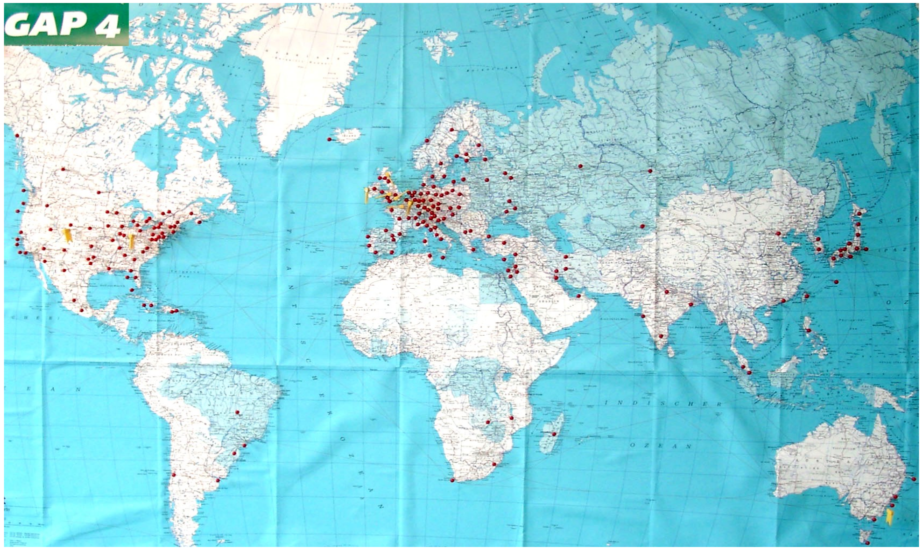
GAP System Features

- Available for UNIX, Windows, MacOS
  - some features, UNIX (and MacOS X) only
- www.gap-system.org
- Current release 4.4.7 (4.5 in winter 06/07, we hope)
- “Drop-in” packages
  - fully integrates code and documentation
- Fully Open ource

- “Read-eval-print” style UI
  - GAP programming language is interface
- On-line help
  - all manuals, Text, HTML and PDF views
  - Many examples checked automatically
- Read files, load packages, save workspace, write output to files, run other programs

Who Uses GAP?

- Estimated 1000+ sites, worldwide
- Math researchers
  - algebraists
  - and algebra consumers
- Teachers and students
- Other Researchers
- Government agencies



- 800+ publications citing GAP or reporting work where GAP is known to have been used

How is GAP Used?

- As a desk calculator
  - purely interactive use
- As a smart ATLAS
  - front-end to databases
- Via one-liners
  - Interactive programming
- As a programming language for single massive background jobs
  - rarely
- Via Web sites
  - experimental, so far
  - Algebra Interactive
- By extending the system and then interacting with it
  - my favourite paradigm

Extend and Interact!

- “What programs do your users want to run?”
- “What is GAP command to solve my problem?”
  - both of these are usually the wrong question
- For best results on a hard, or vague, question:
  - write and debug the necessary extensions to GAP
    - anything from a few simple functions to a large package
  - load them into GAP and explore your problem interactively (remember to keep a logfile)
  - iterate as needed
  - think whether your functions might be useful to others

Free Software
The GAP Community

- Developers
  - volunteers, driven by own research but willing to go “the extra mile”
  - a few people paid to do some jobs
    - hard to get funding for “infrastructure”
  - St Andrews, Aachen, Fort Collins, Braunschweig,......
  - CVS mailing lists, occasional meetings
- User support
  - mailing lists, Web site – support team
- Package authors, and referees, etc.

Sharing GAP code

- We run a mailing list, the GAP forum on which people can share experiences
- We will accept (almost any) GAP code for the “deposited contributions” section on the Website
- Packages that use the standard interface are easy to install, load and upgrade
- We operate a refereeing service for these packages
  - to offer a more thoroughly checked set of “accepted packages”
  - to provide recognition to authors for the work involved

Refereeing Software 1

- Refereeing GAP packages has proved valuable
  - gives authors a reason to write the manual
  - many packages improve dramatically
    - sometimes more like mentoring than refereeing
  - authors seem to value the recognition
    - less clear about authors “bosses”, RAE, etc.

Refereeing 2

- But refereeing is not always easy!
  - hard to find qualified referees
    - sometimes need a “subject” referee and a “GAP” referee
  - hard to define what is “acceptable”
    - a useful and competent, but not original, implementation of published algorithms?
    - an implementation of some new object with definitions and “obvious” methods, but no non-trivial algorithms?
    - an improved help or documentation mechanism?
    - a database and simple interface functions?

Other Package Issues

- Maintenance
  - we have lost touch with some package authors. At the moment we do minimum necessary maintenance.
- Licensing
  - Unclear whether packages MUST be GPL
  - Some authors didn't or don't make clear their licensing
- Namespace
  - increasing problem of collisions between packages

GAP and Other Software

- Many existing packages link GAP to other programs
  - specialized stand-alones for specific group theoretic computations
    - ACE, anupq, nq, carat, nauty
  - General packages in related fields
    - KANT – Algebraic number theory
    - Singular – Algebraic Geometry
  - OpenMath package support for future links
- We see this as increasingly important

SCIEnce Project

- European Research Infrastructure project
- Improve connections (software and human) between systems
- Explore symbolic computation on “Grids”
- Promote symbolic computation
- visitor program at RISC

SCIENCE
Symbolic
Computation
Infrastructure for
Europe

GAP and Free Software Components

- We'd like to avoid reinventing the wheel
  - use GMP, Jenkins hash, Born gc, etc.
- How to manage build process and version choice?
  - have users on Windows, MasOS, Linux and “legacy UNIX”