WELCOME TO THE GAP WORKSHOP

Steve Linton
Division of Computer Science, St. Andrews

July 7, 1997

Welcome

- to St Andrews
- to Computational Algebra and Discrete Mathematics
- to GAP

Welcome to St Andrews

- Three conference locations
  - Here (Maths lecture theatre C) – morning lectures
  - The physics foyer (down, over bridge) – coffee, tea
  - The CS 1st year lab (beyond foyer) – afternoon lab sessions
- Telnet from the lab
- If you need a local UNIX account, use account `workshop` on `chrystal` (Pentium Pro w. Linux) password `gap`
- Local people have coloured badges
- Please pay Werner for accommodation
- Workshop dinner – Thursday – please confirm/deny with Werner

Welcome to Computational Algebra

Why Compute?

- Exploration – to form or dismiss conjectures or gain insight
- Resolve finite problems – sporadic structures, small cases left over by general theorems
- As a research area in its own right – algorithms, algorithmics
- As a testbed for Computer Science – type systems, languages, database concepts

Approaching Problems in Computational Algebra

- Charlie Sims, some years ago, divided problems into three classes

  I Problems where many group elements fit into memory

  II Problems where just a few group elements fit

  III Problems where only a fraction of one element fits

- I would generalise this and speak of:

  I Problems that can be solved quickly by entirely automatic means

  II Problems that can be solved using standard low-level tools and some mathematical ingenuity

  III Problems that require special-purpose programming

- These boundaries are both fuzzy and moving
- Frustration arises when you tackle a problem in one class using tools suitable for another

Welcome to GAP

- Groups, Algorithms, Programming
- Free system for computational algebra and discrete mathematics
- Runs under UNIX, MSDOS and MacOS (at least) – easy to port
- Developed over 11 years in Lehrstuhl D für Mathematik, RWTH Aachen, Germany
  - Prof Joachim Neubüser
  - Martin Schönert
  - Cast of thousands, mainly in Aachen, some elsewhere
- More info `http://www-gap.dcs.st-and.ac.uk/~gap/`
- If you use GAP in research that you publish, please cite it

The parts of GAP

- C kernel
  - Critical functions
  - System interface
  - User interface
  - GAP language
- GAP library
  - Most of the real functionality
  - Examples of good GAP programming
- Data libraries
  Character tables and tables of marks, various types of group; (nearly) all groups of order up to 1000
- Documentation – LaTeX files, also browsable as on-line help and HTML – about 1600 pages in total
- Share packages – user contributed and supported
  - Pure GAP code – eg matrix
  - links to external programs – eg ANU pq program – usually UNIX only

Capabilities

Just a small selection:

- In the library (and kernel)
  - Permutation groups
  - Finite soluble groups
  - Finitely-presented groups
  - Matrix algebras
  - Character tables
  - Generic finite groups
  - Finite fields, cyclotomic fields, algebraic number fields
  - Polynomials
- In share packages
  - Coxeter groups, Hecke algebra, character tables of Chevalley groups
  - Matrix groups and algebras
  - Semigroups and near-rings
  - Graphs and finite geometries
  - Crystallographic groups
  - Finitely-presented monoids
  - Infinite polycyclic groups
  - Crossed modules