

WELCOME TO THE WORKSHOP

Steve Linton

Centre for Interdisciplinary Research in Computational
Algebra, St Andrews

August 29, 2003

Welcome

Welcome

- to St Andrews
- to Computational Algebra and Representation Theory
- to GAP

Welcome to St Andrews

- Conference locations
 - Here (Maths PC Lab) – lectures, lab sessions, coffee and tea outside
 - New Hall – lunches, Friday and Saturday night – next large building but one, going away from town
 - St Salvator's Hall – accommodation, conference dinner – North Street
- Computer access from here
- If you need a local UNIX account, use account workshop on wedderburn password netca
- Local people have coloured badges
- Please pay me for accommodation if you are due to do so and have not already done so

Programme

- 3 short courses of three lectures each by main speakers, plus a lab session
- 3 lectures on GAP, plus a lab session
- Contributed talks – 30 minutes, including questions.
Due to a few late withdrawals, we have some spare slots for these talks. If you'd like to talk, please contact me. This is an excellent chance to give an informal explanation of what you're working on, and how you have used computation, or hope to use it in the future, to a friendly audience.
- Wednesday afternoon has been left free. If you there is enough interest, a guided walking tour of St Andrews can be arranged, costing about £4 per person. Please sign up and pay by Monday lunchtime.

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 17 years. First in Lehrstuhl D für Mathematik, RWTH Aachen, Germany, then internationally, coordinated from St Andrews
 - Prof Joachim Neubüser
 - Martin Schönert
 - Cast of thousands, all over the world
- More info <http://www.gap-system.org/>
- 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 2000
- Documentation – LaTeX files, also browsable as on-line help and HTML – about 1600 pages in total
- Packages – user contributed and supported extensions
 - Pure GAP code – eg autpgrp
 - links to external programs – eg ACE coset enumerator – usually UNIX only

Capabilities

Just a small selection:

- In the library (and kernel)
 - Generic machinery for groups and other structures
 - Permutation groups
 - Finite soluble groups
 - Finitely-presented groups
 - Matrix algebras
 - Character tables
 - Generic finite groups
 - Finite fields, cyclotomic fields, algebraic number fields
 - Polynomials
- In packages
 - Graphs and finite geometries
 - Crystallographic groups
 - Infinite polycyclic groups
 - Coding theory
 - Quantum groups