Introduction

Up to the current academic year the UG4 course Computer Algebra has used Maple as the system for practical work (and as part of the study for the course itself). This has now become rather expensive and as there is a good free alternative it is proposed to switch to that from 2015-16 onwards. The proposed system is Axiom (http://axiom-developer.org). This is a highly sophisticated system which in many ways is preferable to Maple; the main downside is the rather basic interface but it is not a real problem. Switching to this also has the advantage that students can install it on their own machines at no cost. Axiom has been built successfully and tested under DICE.

Changes to DRPS entry

The course descriptor needs some changes to take account of the new situation. All entries that are not shown below stay the same as before. The changes take effect from the 2015-16 session.

Course description

Computer graphics uses various shapes such as ellipsoids for modelling. Consider the following problem: we are given an ellipsoid, a point from which to view it, and a plane on which the viewed image is to appear. The problem is to find the contour of the image as an equation (a numerical solution is not good enough for many applications). The problem does not involve particularly difficult mathematics, but a solution by hand is very difficult. This is an example of a problem that can be solved fairly easily with a computer algebra system. These systems have a very wide range of applications and are useful both for routine work and research. From a computer science point of view they also give rise to interesting problems in implementation and the design of algorithms. The considerations here are not only theoretical but also pragmatic. The design of efficient algorithms in this area involves various novel techniques.

The course addresses the underlying principles and supporting mathematics by considering a few key areas and algorithms. There is also emphasis on developing an intuitive understanding of techniques. There is a two way process in which the desire to solve mathematical problems motivates the design of new algorithms and in turn the process of designing algorithms motivates supporting mathematical material. The course uses the open source computer algebra system Axiom to support coursework and as an example of algorithms in practice.

Summary of Intended Learning Outcomes

1. Use the computer algebra system Axiom as an aid to solving mathematical problems.
2. Design and implement in Axiom appropriate algorithms from constructive mathematical solutions to problems.
3. Evaluate the results obtained from a computer algebra system and discuss possible problems.
4. Explain the gap between ideal solutions and actual systems (the need to compromise for efficiency reasons).
5. Describe and evaluate data structures used in the computer representation of mathematical objects.

6. Discuss the mathematical techniques used in the course and relate them to computational concerns.

7. Apply the mathematical techniques used in the course to related problem areas.

8. Discuss and apply various advanced algorithms and the mathematical techniques used in their design.

9. Use the techniques of the course to design an efficient algorithm for a given mathematical problem (of a fairly similar nature to those discussed in the course).

Syllabus

- Axiom: general design principles, user facilities, data structures, domains, etc.
- Brief comparison of systems.
- Algebraic structures: overview, basic concepts and algorithms.
- Arbitrary precision operations on integers, rationals, reals, polynomials and rational expressions.
- Importance of greatest common divisors and their efficient computation for integers and univariate polynomials (using modular methods).
- Multivariate polynomial systems: solution of sets of equations over the complex numbers; construction and use of Groebner bases; relevant algebraic structures and results.
- Reliable solution of systems of polynomial equations in one variable; Sturm sequences, continued fractions method.

Relevant QAA Computing Curriculum Sections: Data Structures and Algorithms, Simulation and Modelling, Theoretical Computing