Course Announcement

Algorithms for Elementary Algebraic Geometry

Math 191, Fall Quarter 2007
MWF 1–1:50pm, Dodd Hall 162

Instructor. Matthias Aschenbrenner

E-mail. `matthias@math.ucla.edu`

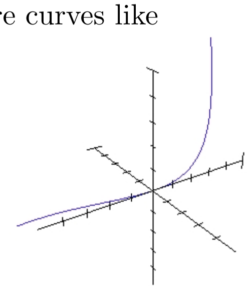
Course webpage. `http://www.math.ucla.edu/~matthias/191.1.07f`

Office. Mathematical Sciences Building 5614

Office phone. (310) 206-8576

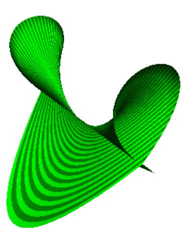
Office hours. M 2:30-4pm, W 2:30-4pm, or by appointment.
(I will *not* hold virtual office hours.)

Description. Geometric objects can often be described as the solution sets of algebraic equations. Simple examples in three-dimensional space are curves like



(the set of triples (x, y, z) of real numbers satisfying the equations $y = x^2$, $z = x^3$)

and surfaces like



(the set of triples (x, y, z) of real numbers solving the equation $x^2 - y^2z^2 + z^3 = 0$).

In this course, we will investigate questions such as:

> How can one compute the equations for the intersection or union of two such objects? How can one determine whether two systems of algebraic equations describe the same geometric object?

These are basic questions at the foundations of *algebraic geometry.* This course is intended as an introduction to this subject, which occupies a central place in modern mathematics. We will learn techniques for translating (certain) geometric problems into algebraic ones. Once they are reformulated in algebraic language, one may unleash the power of (commutative) algebra on them. Sometimes they even become (at least in principle) amenable to treatment by a computer.

However, only fairly recently (since the 1970s) have algorithms (and the computers powerful enough to run them!) become available to actually carry out the necessary computations. The engine behind these is *Buchberger's algorithm,* which is based on the notion of *Gröbner basis.*

The advent of these programs has enabled mathematicians to study complicated examples which previously couldn't be investigated by hand, in this way inspiring a wealth of new mathematics. It has also made the subject interesting for computer scientists and engineers, since many practical questions (e.g., in robotics) can be stated as problems in algebraic geometry.

Prerequisites. A good foundation in linear algebra (at the level of Math 115A) and the ability to formulate mathematical proofs. Some knowledge of abstract algebra would be useful, but is *not* strictly necessary. You should also be able to use (though *not* necessarily to program) a computer.

Course text. In this course we will discuss systems of polynomial equations (*ideals*), their solution sets (*varieties*), and how these objects can be effectively manipulated (*algorithms*). We will try to cover at least the first four chapters of the book

> *Ideals, Varieties, and Algorithms,* An Introduction to Computational Algebraic Geometry and Commutative Algebra, Third Edition, by David Cox, John Little, and Donal O'Shea, Springer, New York, 2007.

We'll try to cover as much of the first four chapters of this book as possible.

Homework. There will be a problem set assigned on a semi-regular basis, handed out in class, and also posted on this website. The problems will range in difficulty from routine to more challenging. Completed solutions are to be handed in at the beginning of class on the due date specified on the respective homework set.

No late homework will be accepted.

However, your lowest homework score will be dropped when computing your grade. You are encouraged to work together on the exercises, but any graded assignment should represent your own work. More detailed instructions about the homework will be provided on a separate handout.

Some of the homework problems (and the midterm exam) will involve the use of computer algebra systems. No previous experience with computer programming is assumed, but I expect that you are able and willing to familiarize yourself with the use of the program of your choice. For overall user-friendliness, I recommend the general-purpose program Maple (which can do algebra, calculus, graphics, and so on). Information on how to use Maple for computations with Gröbner bases may be found in Appendix C of the textbook.

If you prefer, you may also use Macaulay 2, a software system written by Mike Stillman and Dan Grayson, and explicitly designed to support computations in algebraic geometry and commutative algebra. Both systems are available for most platforms (Unix, Linux, Mac OS X, Window$, etc.). While Macaulay 2 is freely downloadable, Maple is not free. (Student licenses for Version 11 run at $99.) However, Maple will be accessible to students in the Program in Computing Lab (2817 Boelter Hall). Other (free) algebraic geometry software includes CoCoA and Singular.

Exams and paper. There will be a take-home Midterm examination, due at the beginning of class on

> **Monday, November 5.**

It will be handed out at the previous class meeting.

> **Students with conflicts with the Midterm Exam in this course are responsible for discussing makeup examinations with me no later than two weeks prior to the exam.**

There will be no Final examination. However, students are required to work on an independent project throughout the quarter. The project will involve studying a class-related topic, and writing a short summary paper on this subject, which will go through several stages of revision. Your paper should be self-contained and accessible to the other participants in the class. Achieving this should take approximately 10 pages. At the end of the course, you will read a referee report written by another student in the class, and you will also write such a report about the paper of another student. More detailed instructions about the project, including a list of possible topics and corresponding references, will be announced in the first week of the quarter.

Grading policy. Homework: 30%. Midterm Exam: 30%. Paper: 40%. All scores and final grades will be available on the MyUCLA gradebook.