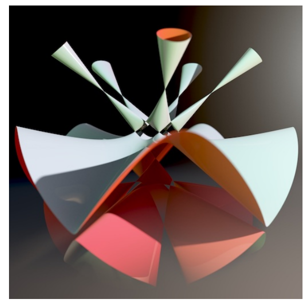
# Math 748 — Computational Algebra — Fall 2020

Instructor: Alexander Duncan

This course investigates systems of polynomial equations from a computational point of view. The main idea is the use of Buchberger's algorithm to compute Gröbner bases. This generalizes both the Euclidean algorithm, which computes polynomial GCDs, and Gaussian elimination, which computes the echelon form of a matrix. Using these tools one can determine whether *any* polynomial system in *any* number of variables has a complex solution.



A. Merzouk (CC BY-SA 3.0)

For example, with ad hoc methods and some patience, you can see that the following system of polynomial equations has complex solutions:

$$\begin{aligned} x^3 - z^3 - 2xy &= 4 \\ 6xz^2 + 4y^2 &= 1 \\ 2x^2 + 3z &= 4. \end{aligned}$$

These kinds of systems are familiar to anyone who did Lagrange multipliers problems in their multivariate calculus class. However, to have computers solve these systems, we require a more systematic approach. The tools developed in this course are used in modern symbolic algebra packages such as Maple, Mathematica, and Sage.

### What's in the course?

The course textbook is *Ideals, Varieties, and Algorithms ($4^{th}$ Edition)* by D. A. Cox, J. Little, and D. O'Shea. I aim to cover chapters 1–5, 8, 9 from the text, in addition to a few weeks of supplementary material. Topics covered include:

- affine varieties and ideals in polynomial rings
- Gröbner bases and Buchberger's algorithm
- elimination theory and resultants
- quotient rings and coordinate rings
- homogeneous ideals and projective varieties.

Supplementary topics will be determined based on the backgrounds and interests of the class, but possibilities include: advanced algorithms, invariant theory, real algebraic geometry, and numerical algebraic geometry.

## Who <u>can</u> take this course?

The textbook only assumes a background in linear algebra and mathematical proof. However, I will assume you have at least a passing acquaintance with fields, rings, and ideals. The standard undergraduate algebra curriculum should be more than sufficient. In particular, the graduate algebra sequence 701/702 is *not* a prerequisite for this course (although they are synergistic). While we will be using computer algebra systems extensively, no knowledge of computer programming is required or expected. If you are unsure of your background, then feel free to send me an email.

## Who <u>should</u> take this course?

*First-year mathematics graduate students:* Incoming graduate students make up one of the primary intended audiences for this course. The course has minimal background requirements and should be accessible to a wide range of students. In particular, I do not assume only "algebra students" will take this course. Knowing your way around a computer algebra system and being conversant in the basic language of algebraic geometry are skills that any mathematician will find useful.

*Algebra graduate students:* Another likely audience for this course consists of those working with, or intending to work with, the algebra group of the department. Many of the concepts in this course are absolutely foundational for work in algebraic geometry or commutative algebra. For those who already have some expertise, there's still a lot to learn. Maybe you already know about schemes and Koszul complexes, but do you know a practical algorithm for computing colon ideals?

*Undergraduate mathematics students:* I expect that there will be some undergraduate mathematics students enrolled in the class. However, this course is designed for graduate students and comes with different expectations. This course is a graduate version of the undergraduate Math 548, which only covers chapters 1–4 from the same text at a relatively superficial level. A student who has taken Math 547 is more than prepared for this course (and if they took that class with me, then they should expect a moderate increase in workload).

*Graduate computer science students:* It is also possible that graduate computer science students may be interested in this class. They are certainly welcome. However, this is a mathematics course, so any interested students should probably have a mathematics degree. Moreover, while many algorithms will be discussed, the emphasis will be on the mathematical questions that they allow us to answer. For example, we will not discuss the computational complexity of these algorithms in any detail, except possibly in the supplementary material at the end of the course.

## Nuts and Bolts

Your final grade will be determined wholly by assignments, which will be assigned roughly every two weeks. We will be using `Sage` as our computer algebra system both in class and on assignments.