Mathematics 392 – Seminar in Computational Commutative Algebra
Final Projects – Topics and Schedule
February 25, 2019

General Information

Recall from the course syllabus that one of the assignments for this seminar will be a final project leading to a paper of about 15 pages and an oral presentation of about 25 minutes duration to the class. Several suggested topics are given below. All of them take material we have learned in this course and extend it in new directions.

I will also be happy entertain any ideas you might have about designing a project topic of your own. If there is some subject you are interested in where solving polynomial equations or Gröbner basis techniques might come up, please do not hesitate to discuss it with me and see if there is a possible project there.

You will work *in pairs*, or *individually* if you prefer, on these projects. If you need help putting a group together, be sure to talk to me *well before March 15.* Once you identify which topic you want to work, please immediately set to work to identify sources, starting from the suggestions below. Ms. Merolli, our Science Librarian will be happy to help you obtain materials through inter-library loan if you need that.

In working on this paper, you should follow the same procedures you would follow in preparing a research paper for any other course, and of course the College Policy on Academic Honesty applies here, as it does to all of your work. Your grade will depend on the thoroughness of your research, the degree of independent thought about the subject revealed through your work, the organization of the paper, and the quality of your writing and oral presentation.

Your papers should use one side only of the sheet, double-spaced. Equations can be entered by hand if necessary. You can also use the TeX/LaTeX mathematical typesetting system (which is what I use for the handouts). See me for a tutorial on its use. Your paper should include a bibliography listing all the sources you consulted. Direct quotations should be identified with foot- or end-notes.

Schedule

Here are some important dates for the projects:

1. *Friday, March 15 (or before)* — Please inform me which topic you have chosen to work on and who you will be working with. You can do this by sending me a short email message, or by talking to me in person.
2. *April 12 - April 19* — During this stretch of the semester, I would like to meet with each group (during office hours, or whenever is convenient for you) to discuss your progress on the project. Of course, you're always welcome at other times too if you need help.
3. *April 29, May 1, May 3, May 6* — Oral presentations. We will schedule who goes which day later in the semester.
4. *Monday, May 6* — Final project papers due.

Possible Project Topics

Important Note: I would prefer that only one group works on any one of the following topics, unless the groups end up specializing in really different aspects of a topic with several different possible directions.

1. Buchberger's Criterion and Improvements to Buchberger's Algorithm

For this project, the main goals would be:

- To learn and present the proof of Buchberger's Criterion for Gröbner bases – that is the statement that G is a Gröbner basis if and only if

$$\overline{S(g_i, g_j)}^G = 0$$

for all pairs $g_i, g_j \in G$ with $i \neq j$.
- Then, to study some of the strategies that people have used to improve the rudimentary form of the Buchberger algorithm that we discussed in class.

Some of the improvements that people have developed include strategies for ordering the list of polynomials so that S-polynomials are more likely to reduce to zero, criteria for detecting when remainder calculations are *unnecessary* (which can be a big time saver – the most computationally intensive part of Buchberger's algorithm is the remainder calculations), and other types of "tweaks" and modifications.

If you want to delve even more deeply into this circle of ideas, another possible topic to consider would be the recent work of other mathematicians such as Jean-Charles Faugère, who have developed entirely different ways of computing Gröbner bases that can outperform any form of Buchberger (the "F4" and "F5" algorithms).

References

a) Start with Sections 2.6 and 2.9 of *Ideals, Varieties, and Algorithms* for the basics. A number of references to original articles are given in Section 2.9
b) J.-C. Faugère, "A new efficient algorithm for computing Grobner bases (`F4`)." *Journal of Pure and Applied Algebra*, **139** (1999), 61–88 (in Science Library)
c) The `F4` algorithm is also discussed in Chapter 10 of *Ideals, Varieties, and Algorithms.*

2. The FGLM Gröbner Basis Conversion Algorithm

Because *lex* Gröbner bases are so useful for elimination of variables, having efficient methods to compute them is a topic of major interest. Unfortunately, the very properties that make *lex* Gröbner bases so useful also make them very difficult to compute in many cases. So, instead of trying to compute them directly via Buchberger's algorithm, an alternate strategy is to compute a Gröbner basis with respect to some "easier" order first (usually *grevlex*), then *convert* the the *grevlex* Gröbner basis to the desired *lex* Gröbner basis by some other transformations. The first published Gröbner basis conversion algorithm was described by Faugère, Gianni, Lazard, and Mora and described in a joint paper by those four authors. As a result, it is called the "FGLM" algorithm. Mathematicians always name things after their inventors (or is it discoverers?).

The basic form of the algorithm works for a *zero-dimensional* ideal I. For ideals I in $\mathbf{Q}[x_1, \ldots, x_n]$, for example, this condition means that the set of points in $\mathbf{V}(I)$ is *finite*, even if we allow solutions that have components in the algebraically closed field $\mathbf{C}$. If you choose this project, you would learn how and why this method works, and present some examples. Then, you could either:

a) Implement the FGLM algorithm (converting to a *lex* order) in the Maple programming language, test it on examples, see how your implementation performs relative to the `fglm` command in the `Groebner package`, etc. *or*
b) Think about whether FGLM can be extended to ideals that are more general than 0-dimensional ideals - for instance, might there be some additional information about the monomial order or the ideal that would allow the same kind of approach to be used, even if the ideal is not 0-dimensional?

References

a) Chapter 5, section 3 of "IVA" for background about what it means for an ideal to be zero-dimensional.
b) ("straight from the horse's mouth") Faugère, J.C., Gianni, P., Lazard, D., and Mora, T. "Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering," *Journal of Symbolic Computation* **16** (1993), 329-344 (in Science Library).
c) Chapter 2, §3 of Cox, Little, O'Shea *Using Algebraic Geometry*
d) Becker, T. and Weispfenning, V. *Gröbner Bases*, Chapter 9, §1.

3. The Gröbner Fan of an Ideal

This topic would be best for a group who wanted to gain a deeper theoretical understanding of the different Gröbner bases for a given ideal (i.e. what happens when you change the monomial ordering, and what all of the possibilities are). One way to make this precise is to consider the "weight orders" $>_{u,\sigma}$ from Problem 10 in Section 2.4 of "IVA." Given a vector u and another monomial order $>_\sigma$ to break ties, we can define

$$x^\alpha >_{u,\sigma} x^\beta \Leftrightarrow \begin{cases} u \cdot \alpha > u \cdot \beta & \text{or} \\ u \cdot \alpha = u \cdot \beta & \text{and } x^\alpha >_\sigma x^\beta \end{cases}$$

In fact, given an ideal I, *every* possible Gröbner basis can be obtained using one of these(!) Suppose we are interested in studying a particular ideal I and all of its different possible Gröbner bases. The basic idea here is that "most" weight vectors u will pick out a unique leading term of highest weight in each of the elements of the given ideal I. The set of weight vectors that select the same leading terms (for all elements of I) forms a *polyhedral cone* in $\mathbf{R}^n$ – a set closed under positive scalar multiples, and with boundary defined by a finite collection of hyperplanes. The collection of all these cones is called the *Gröbner fan* of the ideal. The first main goal of this project would be to work through and present a proof that the Gröbner fan of every ideal consists of a *finite* number of these cones. Then you could consider one or more of the following open-ended questions: What does the Gröbner fan of the ideal of the twisted cubic in k^3 look like? What about the ideals of the parametric curves $\alpha(t) = (t, t^n, t^m)$ for $m > n \geq 2$? How do you determine the Gröbner fan of an ideal I in general? Is it possible to find a finite set of polynomials that is a Gröbner basis for an ideal I with respect to *all* monomial orders simultaneously? How?

References

a) ("straight from the horse's mouth" again) Mora, T. and Robbiano, L. "The Gröbner Fan of an Ideal", in: *Computational Aspects of Commutative Algebra*, L. Robbiano, ed. (in Science Library)
b) Also see Chapter 8, section 4 of Cox, Little, O'Shea, *Using Algebraic Geometry*, 2nd ed.

4. An Application – Conformations of Cyclic Molecules

This topic would introduce you to an application of Gröbner bases, resultants, etc. in the area of computational chemistry. A simplified model for cyclic molecules like cyclohexane: C_6H_{12} (and "cyclo-n-ane" C_nH_{2n} more generally) is to ignore the hydrogen atoms attacted to the cyclic "backbone" of the molecule and translate the minimum energy constraints that would describe the physically observable forms of the molecule into geometric constraints on the *lengths* of the bonds between the carbons and the bond angles at each carbon atom. This leads to a system of algebraic equations that describe the possible *conformations* or geometric forms for molecules of the given type. (For example, cyclohexane comes in both "boat" and "chair" conformations; the difference between these is described by different values for two angles between planes formed by triples of carbon atoms.) So no specific knowledge of chemistry is necessary to work on this topic – the chemistry is converted into questions in pure geometry and algebra!

The equations are sufficiently complicated, though, that some clever Gröbner techniques are necessary to solve them. For this project, you would first work through the geometry of the cyclic 6-atom molecules, investigate their possible conformations using "whatever methods work" to solve the systems of equations. Then the main focus of the project would be to study the analogous questions for 7-atom cyclic molecules (cycloheptane, for instance).

Reference

a) Emiris, I. and Mourrain, B. "Computer Algebra Methods for Studying and Computing Molecular Conformations" *Algorithmica*, Special Issue on Algorithms for Computational Biology, **25** (1999).
b) also see: von zur Gathen, J. and Gerhard, J. *Modern Computer Algebra*, section 24.4.

5. A "Pure" Topic – Invariant Theory of Finite Groups and Molien's Theorem

Note: This topic is suitable only for people who have taken or are taking Modern Algebra 1.

A finite matrix group is a finite subgroup of the group $GL(n,k)$ of invertible $n \times n$ matrices with entries in the field k. Each element A of a matrix group G also acts on the polynomial ring $k[x_1,\ldots,x_n]$ via $f(x) \mapsto f(A\cdot x)$. We say that $f \in k[x_1,\ldots,x_n]$ is an *invariant* of G if $f(A\cdot x) = f(x)$ for all $A \in G$. For example, the 6 matrices

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad S = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \qquad S^2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$
$$T_{12} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad T_{13} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \qquad T_{23} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

form a subgroup G of $GL(3,\mathbf{Q})$ (isomorphic to the symmetric group on 3 letters). The polynomials

$$\sigma_1 = x_1 + x_2 + x_3,\ \sigma_2 = x_1x_2 + x_1x_3 + x_2x_3,\ \sigma_3 = x_1x_2x_3$$

are invariants of G, and indeed, *every* polynomial invariant of G can be expressed as a polynomial expression $g(\sigma_1,\sigma_2,\sigma_3)$. Invariant theory is the study of the structure of the invariants of matrix groups such as this. It was a very "hot topic" in 19th century mathematics and it has a number of important applications. But eventually the computations that people wanted to carry out essentially became too difficult to do by hand, and at the same time mathematics as a whole moved in a much more abstract direction. Gröbner basis methods (and other techniques from computational algebra) have made possible a resurgence of interest and renewed progress in invariant theory.

For this topic, you would learn about the basic ideas involved (topics: Gröbner basis test for subring membership, the Reynolds operator and Noether's theorem which shows finite generation of rings of invariants, Molien's theorem) and study the rings of invariants in some interesting cases. The goal would be to present a proof of Molien's theorem, which gives a truly beautiful and wonderful formula for computing the dimension of the vector space of invariants of G in the homogeneous polynomials of degree t, for all t *simultaneously*. We write S_t^G for this space of invariants ($S = k[x_1,\ldots,x_n]$ is the polynomial ring G acts on; S_t is the vector subspace of homogeneous polynomials of degree t). Then Molien's theorem says:

$$\sum_{t=0}^{\infty} \dim_{\mathbf{C}}(S_t^G)u^t = \frac{1}{|G|}\sum_{g\in G}\frac{1}{\det(I-ug)}.$$

References

a) IVA, Chapter 7.
b) Sturmfels, *Algorithms in Invariant Theory*, Chapters 1 and 2.

c) Cox, Little, O'Shea, *Using Algebraic Geometry*, Chapter 6 for background needed for Molien's Theorem.

6. A More Applied Topic – Mechanical Linkages

In mechanical engineering, robotics, etc. an important area of the study is the motions of *mechanical linkages* – collections of rigid segments joined by joints of various types. (For example, simple planar linkages would have rigid segments of fixed length, joined by revolute joints.) We will have looked at a few examples of how varieties and Gröbner bases can be used to study questions about linkages already in the seminar. Several further questions could form the basis of interesting projects here.

A) To start, the students would work out the trajectory of a marked point on a general *"four-bar"* linkage (these are called Watt curves, after James Watt, the inventor of the steam engine – see below) to investigate questions like: What kinds of curves can you get here by changing the lengths of the segments in the linkage, etc.? Then, a 3-RPR parallel manipulator is a planar robot linkage consisting of three "arms" each fixed at one end with a revolute joint; each containing a prismatic joint, and each attached via a second revolute joint to one common end-effector. A question for which no really good general answer is known at present is: How can we determine the *maximal workspace* of the robot – the set of all points reachable by the end-effector, given the locations of the fixed ends of the arms, the minimum and maximum lengths of the prismatic joints, etc.? This is a good problem, and it can be attacked by Gröbner basis methods!

B) Another, more mathematically-oriented, question here is as follows. The original interest in linkages started in the 19th century with the invention of steam engines for farm implements, manufacturing, locomotives, etc. One of the early questions in the subject was: How can a linkage be constructed to "turn circular motion into straight-line motion"? This was eventually solved by a French engineer named Peaucellier. Another well-known mathematician of the time, A. Kempe, studied linkages in great detail. (He also published an incorrect proof of the 4-color theorem!) Kempe sketched a proof of a theorem that says that *every* bounded portion of every variety $\mathbf{V}(f(x,y))$ in $\mathbf{R}^2$ can be "drawn" by following the trajectory of some point in a suitable1 mechanical linkage (!) For this topic, the students would learn the ideas behind Kempe's proof, which gives (in principle) a method to synthesize a linkage to draw an arc of any given variety, given the polynomial equation $f(x,y) = 0$. One unfortunate aspect of Kempe's approach is that the linkages he would find to construct even simple curves like ellipses are extremely complicated. Try one to see what I mean! His approach yields an *existence proof* for the linkage. It is far from a practical construction of a minimal (or close-to-minimal) linkage for a given curve. Another possible part of the project would be to try to address the following questions: Are there simplifications that are possible here? Are there better upper bounds for the number of segments in a linkage one would need to draw a general curve of a given degree.

References

1) IVA, Chapter 6 for general information on applications of Gröbner bases to questions in geometry of robots, etc.

For Topic A:

2) Merlet, J.-P. "Some Algebraic geometry problems arising in the field of mechanism theory, in: Algorithms in Algebraic Geometry and Applications Birkhäuser Progress in Mathematics 143.

For Topic B (unfortunately, neither of these are especially easy to read):

3) King, H. "Planar Linkages and Algebraic Sets", preprint 1998
4) Kapovich, M. and Millson, J. "Universality Theorems for configuration spaces of planar linkages", *Topology* **41** (2002), 1051–1107.

Very brief introductions to the general ideas can be found in:

5) Courant, R. and Robins H. What is Mathematics?
6) Hilbert, D. and Cohn-Vossen, S. Geometry and the Imagination.

7. A More Applied Topic – Algebraic Statistics

The rapidly developing field of algebraic statistics is based on the idea that many statistical models (i.e. families of probability distributions) for discrete data can be seen as algebraic varieties. Moreover the geometry of those varieties determines the behavior of parameter estimation and statistical inference procedures. A typical example is the family of binomial distributions. The probability that a binomial random variable X (based on n trials with success probability θ on each trial) takes value $k \in \{0, 1, \ldots, n\}$ is

$$p_k = P(X = k) = \binom{n}{k} \theta^k (1 - \theta)^{n-k}.$$

Viewing these as components of a curve parametrized by real θ satisfying $0 \leq \theta \leq 1$, we have a subset of the real points of a rescaled rational normal curve (the natural generalization of a twisted cubic curve to higher dimensions) of degree n lying in the hyperplane defined by the equation $p_0 + \cdots + p_n = 1$. The recent book by Seth Sullivant, *Algebraic Statistics* (AMS, 2018) is a good general reference. There are several possible ways to think about pursuing this topic:

(A) Given some number of observations we might want to estimate θ using maximum likelihood estimation, and this leads to a constrained optimization problem involving polynomial equations. Thinking about practical methods for doing this leads to some interesting questions. A good introduction to the basics of model construction and experimental design can be found in `Pistone 2001` (see the bibliography of IVA starting on page 627 for this and the other references here). A discussion of algebraic techniques for maximum likelihood estimation appears in Chapter 2 of `Drton 2009`.

(B) One of the main applications of these ideas so far has been in *genomics* and *bioinformatics*—the study of the information contained in DNA sequences. For students with the requisite background (a basic familiarity with DNA structure and some elementary genetics would suffice), the Jukes-Cantor models studied in Part I of `Pachter 2005` could form the basis of a more extensive project.

(C) A different sort of application to design of experiments can be found in Chapter 4 by Kehrein, Kreuzer, and Robbiano in `Dickenstein 2005`. This draws on material on *border bases* as well (see p. 625 in IVA) – a sort of complement to the theory of Gröbner bases for zero-dimensional ideals.