Lecture 0: Introduction & Overview of Course

Rafael Oliveira

University of Waterloo
Cheriton School of Computer Science

rafael.oliveira.teaching@gmail.com

January 10, 2021

Overview

- Introduction
  - What is this course about?
  - What to expect from me?
  - What do I expect from you?
  - Guidelines
- Logistics
  - Structure of Lectures
  - Homework
  - Final Project
  - Student drop-in hours
  - Responses from survey
  - Words of Wisdom
  - Questions?

What is this course about?

Many natural problems and algorithms are arithmetic or algebraic in nature:

What is this course about?

Many natural problems and algorithms are arithmetic or algebraic in nature:

- Testing if a number is prime
- Multiplying two matrices
- Computing determinant of a matrix
- Evaluation of polynomial
- Factoring a number or a polynomial
- Decomposing algebraic varieties into irreducible components
- Solving polynomial equations
- many more...

What is this course about?

Many natural problems and algorithms are arithmetic or algebraic in nature:

- Testing if a number is prime
- Multiplying two matrices
- Computing determinant of a matrix
- Evaluation of polynomial
- Factoring a number or a polynomial
- Decomposing algebraic varieties into irreducible components
- Solving polynomial equations
- many more...

This course aims to:

What is this course about?

Many natural problems and algorithms are arithmetic or algebraic in nature:

- Testing if a number is prime
- Multiplying two matrices
- Computing determinant of a matrix
- Evaluation of polynomial
- Factoring a number or a polynomial
- Decomposing algebraic varieties into irreducible components
- Solving polynomial equations
- many more...

This course aims to:

- introduce you to *why* such problems are fundamental
- *why* symbolic approach is important

What is this course about?

Many natural problems and algorithms are arithmetic or algebraic in nature:

- Testing if a number is prime
- Multiplying two matrices
- Computing determinant of a matrix
- Evaluation of polynomial
- Factoring a number or a polynomial
- Decomposing algebraic varieties into irreducible components
- Solving polynomial equations
- many more...

This course aims to:

- introduce you to *why* such problems are fundamental
- *why* symbolic approach is important
- expand your algorithmic toolkit (symbolic algorithms, use of randomness, finite fields, etc.)

What will we learn?

This course is focused on the symbolic computation approach for algebraic and geometric problems, and their applications. So we will learn:

What will we learn?

This course is focused on the symbolic computation approach for algebraic and geometric problems, and their applications. So we will learn:

- Algorithms for manipulation of polynomials
- Discrete Fourier Transform
- Polynomial Division, GCD
- Computing determinant of a symbolic matrix
- Resultants and their geometric significance
- Factoring a polynomial
- Algorithm for ideal membership problem
- Solving polynomial equations
- Computational Invariant Theory
- Applications to other areas (coding theory and computer science in general)

What to expect from me?

Here is what you can expect from me:

- Give lectures
- Provide homework which helps you understand the material better
- Be present during student drop-in hours
- Help you with choosing your final project topic
- Give feedback on you final project mid way
- Be active on Piazza (to the extent that I will be able to, without hurting the points above)

What to expect from me?

Here is what you can expect from me:

- Give lectures
- Provide homework which helps you understand the material better
- Be present during student drop-in hours
- Help you with choosing your final project topic
- Give feedback on you final project mid way
- Be active on Piazza (to the extent that I will be able to, without hurting the points above)

I care much more about your **learning** rather than your **grades**. I also care much more about your **exploration** of this vast field of symbolic computation rather than **forcing on you** any opinion on what are the "important problems." So I am designing a course which reflects that.

What do I expect from you?

If you are taking the class, I expect that you:

- Do the homework
- Participate in class (asking questions, correcting me, etc.)
- Come to office hours - we are here to help you!
- Always ask yourself: "why is this important? Why should I care?"
- Explore the topics, and/or some area that fascinates you!
- Always keep an open mind!
- Be kind to your classmates, to the TAs and myself
- Participate on Piazza (asking question, answering your classmates' questions if you know the answer)
- Provide me feedback on how the course is going
- Let me know if any problems arise during the term, so we can help as soon as possible.

What do I expect from you?

If you are taking the class, I expect that you:

- Do the homework
- Participate in class (asking questions, correcting me, etc.)
- Come to office hours - we are here to help you!
- Always ask yourself: "why is this important? Why should I care?"
- Explore the topics, and/or some area that fascinates you!
- Always keep an open mind!
- Be kind to your classmates, to the TAs and myself
- Participate on Piazza (asking question, answering your classmates' questions if you know the answer)
- Provide me feedback on how the course is going
- Let me know if any problems arise during the term, so we can help as soon as possible.

There will be no participation points, as we are doing this online and people may not be available during the class times, etc.

Guidelines for the course

Please read

`https://cs.uwaterloo.ca/~r5olivei/courses/2021-winter-cs487/guidelines/`

- Introduction
  - What is this course about?
  - What to expect from me?
  - What do I expect from you?
  - Guidelines

- Logistics
  - Structure of Lectures
  - Homework
  - Final Project
  - Student drop-in hours
  - Responses from survey
  - Words of Wisdom
  - Questions?

Structure of lectures

Lectures will be live, but in (somewhat) asynchronous format (to benefit everyone). This means:

- Each lecture will have the same duration as a regular lecture
- However, each lecture will be divided into 2-3 parts (not necessarily same duration) with short breaks in between
- Videos will be posted on youtube shortly after lecture (please let me know if you **cannot** access youtube nor zoom!)
- We will use **zoom** for lectures, and links will be provided on Piazza

Homework

We will have 5 problem sets for this class. See *assignments* section in

`https://cs.uwaterloo.ca/~r5olivei/courses/2021-winter-cs487/`

Homework

We will have 5 problem sets for this class. See *assignments* section in

`https://cs.uwaterloo.ca/~r5olivei/courses/2021-winter-cs487/`

Due dates (tentative):

- January 25th
- February 12th
- March 5th
- March 19th
- April 5th

Homework

We will have 5 problem sets for this class. See *assignments* section in

`https://cs.uwaterloo.ca/~r5olivei/courses/2021-winter-cs487/`

Due dates (tentative):

- January 25th
- February 12th
- March 5th
- March 19th
- April 5th

Each homework will have n questions, where $n \sim 6$ and you will be required to turn in only $n - 1$ of them. Please list collaborators.

Homework

We will have 5 problem sets for this class. See *assignments* section in

`https://cs.uwaterloo.ca/~r5olivei/courses/2021-winter-cs487/`

Due dates (tentative):

- January 25th
- February 12th
- March 5th
- March 19th
- April 5th

Each homework will have n questions, where $n \sim 6$ and you will be required to turn in only $n - 1$ of them. Please list collaborators.

In addition, I will post practice problems (won't be graded/don't turn in) which are selected so that you can get a better understanding of the material. I am a strong believer that we only *learn by doing*.

Final Project

`https://cs.uwaterloo.ca/~r5olivei/courses/2021-winter-cs487/final-project/`

- Topic of your choice (see page above for suggestions)
- Undergraduates can pair up for final project (not mandatory to pair though)
- Goal: write code, work on an open problem and/or present a survey on a problem or area of your choice within symbolic computation.
- To turn in: project report (around 10 pages - see LaTeX template)
- If you are programming something, then report has to be documentation and reasoning/explanation for correctness of your program.
- Individual work: each student will be required to do a 20 min presentation on their project, and then there will be a 10 min period for questions (by me and the TA)

Student drop-in hours

(For now) I will hold student drop-in hours on Mondays from 4pm-5pm and on Wednesdays from 9pm-10pm. This should comfortably cover all of the time zones that the students from the class are in. Other times could be set by appointment.

You are always welcome to attend the drop-in hours to ask questions about the course, about the final project, and about research in general. My drop-in hours will be hosted via zoom, and a link for the will be posted on Piazza.

For the TA student drop-in hours, please take the Piazza poll on preferences for timing, and we will do our best to accommodate what suits everyone best.

Make sure to attend office hours, they are an invaluable resource!

Response to questions from survey

- This course will require some familiarity with linear algebra and with writing proofs.
- Is this a CS course or a PMATH course?
  - This is a course at the intersection of CS and PMATH, as we talk about the analysis of symbolic algorithms, their applications (which are also in PMATH questions, as well as more practical places)
  - In this course we will understand why the algorithms we will see work, and that requires mathematics, for the formal understanding of algorithms
- Additional examples which I won't have time to cover in class will be given as practice problems or homework problems.
- Some homework problems will also be intended for you to extend your knowledge on a particular topic.
- The programming part of this course will be done most likely in Macaulay 2, and familiarity with this language will not be required.
- Written and scanned responses to homework problems will be accepted, so long as it is *clearly legible*.

Words of Wisdom

- This course will require you to spend time thinking about the concepts presented to you
- Some of the concepts are quite sophisticated, so make sure you save time to think about them, try examples, and understand them.
- You are the person who knows best about yourself. But in general, if you are planning to take six courses, and this is one of them, I will say that this is not a great idea.
- The two main applications that we will see in class are in coding theory and invariant theory, and how the latter connects to many areas of mathematics and computer science.
  - this will require us to learn a bit more concepts, which I will introduce in the course
  - means that class is content heavy, but this content is necessary to give you breadth and depth on where symbolic computation is used
- Make sure to use the office hours as an invaluable resource for your learning (and this will also help others, since your questions may be questions that everyone has as well)

Questions?