CAS CS 132

Geometric Algorithms

Spring 2020

Meeting Place: STO B50

Meeting Time: TR 3:30 pm – 4:45 pm

Instructor: Prof. Mark Crovella

- **Office:** MCS-101B
- **Office Hours:** XXX, XXX
- **Email:** crovella@bu.edu

Teaching Fellow 1: TBD

- **Office Hours:** TBD
- **Office Hours Location:** TBD
- **Lab Tutoring Hours:** TBD.
- **Email:** TBD

Teaching Fellow 2: TBD

- **Office Hours:** TBD
- **Office Hours Location:** TBD
- **Lab Tutoring Hours:** TBD.
- **Email:** TBD

Overview of the Course

This course will introduce you to linear algebra from a Computer Science standpoint. Linear algebra is such a useful tool that it is crucially important to a number of areas in Computer Science. For example, if you study optimization, the starting point is linear algebra. If you study computer graphics, the language you use every day is linear algebra. If you study the performance of computer systems, you need linear algebra. If you study algorithms – especially graph algorithms – you will absolutely need linear algebra. If you study data mining, you will use linear algebra all the time.

The dominance of linear algebra arises because it is so fundamental, and in some ways, very simple. It deals with objects that almost always can be interpreted geometrically. So often we can use linear algebra in a very intuitive manner – so much so that many times it is actually the best way to think about geometric problems. But it is also rigorous – and so it captures situations that sometimes we might overlook if we were proceeding purely intuitively. So the advantage of being basic and fundamental is that it can be used and applied in so many ways.

As mentioned already, this course is taught from a computer science perspective. As such, it seeks to develop (simultaneously) three different modes of thinking: algebraic, geometric, and computational. To support this, the course makes heavy use of interactive technology and you will be expect to do significant programming. There is more information about the teaching philosophy of the course in the "CS 132 Teaching Philosophy" document in the course repository.

Getting Set Up

You will need to set up access to the following online materials. Instructions for how to do all of those setups are below. Note the first five are **required;** Using Github and DiagramAR are **not** required but **highly recommended.**

Required The online textbook ("Linear Algebra and Its Applications"),

Required Python on your laptop for homeworks,

Required Tophat for inclass participation,

Required Piazza for discussion of assignments and course material,

Required Gradescope for assignment submission,

Recommended the course `github` repository, for study lecture notes and for experimentation, and

Recommended the visualization app `DiagramAR`, for studying and interacting with figures via augmented reality.

Textbook

The text for the course is *Linear Algebra and Its Applications* (LAA), 5th edition, by David C. Lay, Steven R. Lay, and Judi J. McDonald.

Lectures are closely tied to book sections, and you will get the most out of lecture if you read the corresponding book section beforehand. You will also want to refer to the book in completing homeworks. See the schedule at the end of the syllabus.

You do **not** need to buy a hardcopy of this textbook! We will use an online textbook for this course, via a product called *MyMathLab*. This online text is being provided free of charge by the publisher, on a trial basis. We only ask that you provide short feedback in the form of a digital survey at the end of the course.

***Setup**: Step-by-step instructions for accessing MyMathLab are on Piazza. In brief: go to* `https://www.pearson.com/mylab`*, select Register as a Student. The instructor's course ID is crovella29991. Create an account if you don't have one, then select the access code option, and enter the access code WSCMMV-PASHM-ILMEN-COMET-DOLBY-VEXES. To view the text, you will also need to install the Wolfram CDF viewer on your local machine (available from the Pearson site).*

Programming Environment

We will use `python` as the language for teaching and for assignments that require coding. You are expected to know python.

***Setup**: Instructions for installing and using Python are on Piazza.*

Tophat

We will be using "Peer instruction" as part of the lectures. This requires you to answer questions during lecture, sometimes after discussion with your classmates.

To support this, we will use Tophat (`www.tophat.com`) for student feedback during lecture. You will be able to submit answers to in-class questions using Apple or Android smartphones and tablets, laptops, or through text message.

Should you require assistance with Top Hat at any time, due to the fact that they require specific user information to troubleshoot these issues, please contact their Support Team directly by way of email (`support@tophat.com`), the in-app support button, or by calling 1-888-663-5491.

You will also want to bring pencil and paper to lecture. This isn't absolutely critical, but you will find it easier if you can jot a note or two while responding to in-class questions.

Setup: *If you registered before the semester start, you should have gotten an email with instructions for how to create your account and sign in. If you are adding the class late, you can register using the join code: 794726; if you have problems, please contact a TF.*

Piazza

We will be using Piazza for class discussion. The system is well tuned to getting you help fast and efficiently from classmates, TFs, CAs, and myself. Rather than emailing questions to the teaching staff, please post your questions on Piazza. Our class Piazza page is at: `piazza.com/bu/spring2020/cs132/home`. We will also use Piazza for distributing materials such as homeworks and solutions.

When someone posts a question on Piazza, if you know the answer, please go ahead and post it. However pleased *don't* provide answers to homework questions on Piazza. It's OK to tell people *where to look* to get answers, or to correct mistakes; just don't provide actual solutions to homeworks.

Setup: *If you registered before the semester start, you should have been automatically enrolled. If you are adding the class late, go to Piazza at the link above, and enroll yourself. If you have any problems, please contact a TF.*

Gradescope

Assignments will be submitted via Gradescope (`https://www.gradescope.com/`). Graded assignments will be returned to you via Gradescope as well. If you have any questions about the grading you receive on Gradescope, please contact a TF.

Setup: *If you registered before the semester start, you should have been automatically enrolled. If you are adding the class late, go to Gradescope at the link above, and enroll yourself using the entry code 9KBJ2P. If you have any problems, please contact a TF.*

Lecture Slides and Code

The lecture slides I use are actually executable python scripts, using the `Jupyter notebook`. You can download and execute the examples on your own computer, and you can modify them any way you'd like, play around with them, experiment, etc.

The slides I use in lecture are published on `github`. If you are new to Github, check out one of the short online tutorials, eg, `https://rogerdudler.github.io/git-guide/`. From the commandline, all you really need is "git clone" and "git pull" to stay up-to-date. If you are more experienced with Github, I recommend you fork this repo so that it moves to your own Github account.

These notebooks are updated on a regular basis, so it is advisable to pull with each use. (However, if you make local edits, you'll want to work out a system to make sure you don't accidentally overwrite any edits you've made when pulling. One example is to copy each file and add a '-YourName' extention to it, leaving the original untouched.)

I invite pull requests, but I'll only merge them if you can present and explain the learning value of your changes (i.e., how might your change represent concepts or code that best exemplifies the underlying linear algebra or python programming?)

Setup: *The repository is*
`https://github.com/mcrovella/CS132-Geometric-Algorithms`.

DiagramAR

This semester we are trying out a new teaching tool to help you visualize geometric figures. It is called `DiagramAR`. It is an iPhone/iPad app (Android is in development, but not ready yet - sorry!).

Use of `DiagramAR` is optional – it's intended to augment your understanding of the existing lecture material. We developed `DiagramAR` right here at BU, and this is the first class to try it out!

The purpose of the app is to help you get a concrete sense of the geometry of various figures that are used in the course. It presents 3-D figures in augmented reality (AR). As a result, you can look at the figures "in space" and can move around them to look at them from different angles, rotate them, or zoom in/out.

The app works in two different ways. First, it responds to QR codes in the lecture notes. Whenever you see a figure in the lecture notes with a QR code next to it, then pointing the app at that code will pop up a 3D representation of the figure in AR. Second, you can create your own figures by inputting the equations for lines, planes, and other surfaces.

The primary developer for `DiagramAR` is Dennis Henneman (BU CS) and the designer is Xiqiao (Lily) Chen (BU CFA).

Setup: *DiagramAR is in the Apple App Store at*
`https://apps.apple.com/us/app/diagramar/id1484987191`.

Reading and Homeworks

1. You have about 10-12 pages of reading for each class. Class will be more understandable if you do the reading first. A good plan is to set aside time on Mondays and Wednesdays to do readings.

2. Homeworks will be assigned on Thursdays.

3. Homeworks are due at 11 am on the following Thursdays. This means they are due before the start of Thursday's class.

4. You can discuss homeworks in section meeting on Mondays. But don't expect that TFs will be going into detail – instead, they will answer specific questions!

5. Homeworks will be submitted via *gradescope*. **XXX FILL IN INSTRUCTIONS XXX.**

Submitting Homework

For showing your analytical / mathematical work, there are three options, in increasing order of quality:

1. You can scan handwritten notes into PDF. Note that these must be **clear** and **neat** because the grader will simply read them as best s/he can.

2. You can write up your work in Word, using the built-in equation editor for the mathematics. Then save as PDF.

3. You can learn and use LaTeX. This is the tool that produces a professional, publishable document. It is what serious computer scientists use. You can learn to use it in a few hours, starting from `http://www.latex-tutorial.com/`. Eventually you will find it useful for lots of your coursework, so it makes sense to learn it now.

For showing your computational work, you will often submit code and/or scripts showing your code runs. For the code, simply submit them as .py files. For the scripts showing your code executing, you can use the built-in logging system of ipython. I recommend:

1. `%logstart -ort hwk-file.txt backup`

2. run your code in the interpreter showing the output

3. `%logstop`.

At which point the file `hwk-file.txt` will contain a record of the inputs and outputs of your code. (Note that output from "print" statements will not show up however).

Course and Grading Administration

Assignments will be submitted using `Gradescope`. The TF will explain how to submit assignments.

NOTE: IMPORTANT: Late assignments **WILL NOT** be accepted. However, your final grade will be based on the top 10 homeworks submitted (out of 12).

Final grades will be computed based on the following:

50% Homework assignments. The top 10 homework grades (out of the 12 assigned) will be used to compute this score.

5% Attendance and in-class participation via TopHat.

20% Midterm

25% Final (Cumulative)

To get full credit for class participation by TopHat, you need to respond to 85% of the questions that are posed in lecture. So if you miss a question here or there, or forget your device one day, don't worry as long as you come to lecture consistently.

The exact cutoffs for final grades will be determined after the class is complete.

You need to consistently work the problem sets each week. Plan to set aside a regular time each week to do them.

Academic Honesty

You may discuss homework assignments with classmates, but you are solely responsible for what you turn in. Collaboration in the form of discussion is allowed, but all forms of cheating (copying parts of a classmate's assignment, plagiarism from books or old posted solutions) are NOT allowed. We – both teaching staff and students – are expected to abide by the guidelines and rules of the Academic Code of Conduct (which is at `http://www.bu.edu/dos/policies/student-responsibilities/`).

You can probably, if you try hard enough, find solutions for homework problems online. Given the nature of the Internet, this is inevitable. Let me make a couple of comments about that:

1. If you are looking online for an answer because you don't know how to start thinking about a problem, talk to a TF or myself, who may be able to give you pointers to get you started. Piazza is great for this – you can usually get an answer in an hour if not a few minutes.

2. If you are looking online for an answer because you want to see if your solution is correct, ask yourself if there is some way to verify the solution yourself. Usually, there is. You will understand what you have done *much* better if you do that.

3. If you are looking online for an answer because you don't have enough time and are getting close to the assignment deadline, think about this:

   (a) what you are doing is intellectually dishonest,

   (b) you are going to have to solve problems like this on the midterm and final, and

   (c) you can miss up to two homeworks without penalty.

   So ... it would be better to simply submit what you have at the deadline (without going online to cheat) and plan to allocate more time for homeworks in the future.

Course Schedule

IMPORTANT: It is important to do the reading *before* the class on which it is based. All readings are from LAA.

Date	Topics	Reading	Assigned	Due
1/21	1: Linear Equations	1.1		
1/23	1: Linear Equations (wrapup), 2: Numerics	notes only	H1	
1/28	3: Row Reduction	1.2		
1/30	4: Vector Equations	1.3	H2	H1
2/4	5: $A\mathbf{x} = \mathbf{b}$	1.4, 1.5, 1.6		
2/6	6: Linear Independence	1.7	H3	H2
2/11	7: Linear Transformations (Guest Lecture)	1.8		
2/13	8: Matrices of Linear Transformations	1.9	H4	H3
2/18	**Substitute Monday; No Class**			
2/20	9: Matrix Operations	2.1	H5	H4
2/25	10: Matrix Inverse	2.2, 2.3		
2/27	11: Markov Chains	4.9		H5
3/3	12: Matrix Factorizations	2.5		
3/5	**Midterm**		H6	
3/10	**Spring Break**			
3/12	**Spring Break**			
3/17	13: Computer Graphics	2.7		
3/19	14: Subspaces of $\mathbb{R}^n$	2.8	H7	H6
3/24	15: Dimension and Rank	2.9		
3/26	16: Eigenvectors and Eigenvalues	5.1	H8	H7
3/31	17: The Characteristic Equation	5.2		
4/2	18: Diagonalization	5.3, 5.4	H9	H8
4/7	19: PageRank	notes only		
4/9	20: Inner Product, Length	6.1	H10	H9
4/14	21: Orthogonal Sets	6.2		
4/16	22: Least Squares	6.3, 6.5	H11	H10
4/21	23: Linear Models	6.6		
4/23	24: Symmetric Matrices	7.1, 7.2, 7.3	H12	H11
4/28	25: Singular Value Decomposition	7.4		
4/30	26: Applications of SVD			H12