# Mathematical Methods in Linguistics

| Course | Info |
|---|---|
| Course# | lin539 |
| Time | MF 1:00-2:20pm |
| Location | Humanities 3020 |
| Website | lin539.thomasgraf.net |
| | |
| Instructor | Thomas Graf |
| Email | [coursenumber]@thomasgraf.net |
| Office hours | M 11:00-12:30 & 3:30-4:00 |
| | W 3:30-4:00 |
| | F 11:15-11:45 |
| Office | SBS N-249 |

**Attention:** To get access to the private course repository, you must email me your github username.

**A friendly plug:** If you are interested in this class, also consider attending the department's Mathematical Linguistics Reading Group.

## Course Outline

### Bulletin Description

An overview of the mathematical foundations of theoretical and computational linguistics. Topics covered include set theory, morphisms, logic and model theory, algebra, lattices, lambda calculus, probability theory, information theory, and basics of formal language theory. A strong emphasis is put on the linguistic application of the mathematical concepts in the study and analysis of natural language data.

### Full Description

This course is an introduction to mathematics in linguistics. It aims to help students familiarize themselves with mathematical concepts and applications that are widely relevant to theoretical and/or computational linguistics. This covers a wide range of topics, mostly from *discrete mathematics*. The course is also very different from what you did in high school, there's precious few numbers here and we don't care much about integrals or calculating compound interest. In contrast to a proper mathematics course, we also focus more on techniques and tools rather than theorems and proofs. This means that you will learn how to work with things like matrices, semirings, and lattices, but you won't have to prove things about them. So this is more like a CS methods course than a proper math class.

For more information about the content, see the Selected Topics section. You will see that the schedule for this class is very ambitious. It has to be: this class serves an integral function of our Computational Linguistics MA and must get students to a level where they can take courses and read textbooks on mathematically demanding topics such as machine learning. This is not your typical graduate-level course, it is a **boot camp**, so be prepared to invest a fair amount of blood, sweat, and tears.

### Mode of Instruction: Hybrid Class

The most important mathematics skill is the ability to learn mathematics on your own, without the help of an instructor. Whether you're doing computational linguistics in the industry or as an academic, sooner or later you will come across some tool or technique that builds on an area of math you have never encountered before. The most important thing at that point is that you can pick up a textbook or survey paper and teach yourself how this unfamiliar kind of math works.

Unfortunately, math is already a very challenging topic for most people, and learning it without somebody's help seems impossible to many. In order to teach you how to make the transition from listener to autonomous learner, this course is run as a **hybrid class**. This means that a lot of the learning takes place outside the class room, with physical meetings serving mostly as an evaluation that your learning process outside the class room was successful. In the context of this specific class, this will work as follows:

- Each week you are assigned several units from the lecture notes.
- You discuss the material with each other on github, using the issue tracker.
- Monday classes are *Q&A sessions*. These are **not** lectures. I will not present material from square one. The assumption is that you have worked through the assigned units and have focused questions on specific points. Attendance of Q&A sessions is not mandatory.
- Friday classes are *exercise sessions* where you present your solutions to the assigned exercises. Attendance is mandatory.

Expect the workload outside the class room to be much higher than usual, but on the flip side you can skip Monday class if you have no questions.

The lecture notes will be made available as Jupyter notebooks in the main github repository. A Jupyter notebook is a mixture of text and Python code, which allows for a more interactive learning environment. There are multiple ways you can view the notebooks:

1. Use Stony Brook's Virtual SINC site, which already has Jupyter installed.
2. Use our virtual machine image for VirtualBox, available at Stony Brook's Softweb.
3. Install Anaconda, a Python distro that also installs Jupyter.
4. If you already have a working Python setup, install Jupyter separately.
5. If you can live without the interactive Python demonstrations, you can just read the notebooks directly on github.

For all of them, you should use the supplied `start_jupyter.py` script to start the Jupyter server. Proceed as follows:

1. Clone or download the git repository (green button at the top of the page).
2. If you downloaded the repository as a zip archive, extract it.
3. Run the `start_jupyter.py` script. The Jupyter notebook server will start and open a new tab in your browser.
4. Navigate to the notebook you want to read. They are all in the notebooks folder.

**Prerequisites**

Students are expected to have some previous experience with phonology and syntax at the undergraduate or graduate level. Students who do not satisfy this requirement should be enrolled in Syntax 1 and either Phonetics or Phonology 1 at the same time. No prior mathematical or computational experience is required.

## Selected Topics

A brief selection of the topics to be covered (though we will proceed in a different order):

1. Basics of mathematics
    - Topics: sets, multisets, tuples, functions, relations
    - Applications: bag of words model of text, n-gram models of grammaticality
2. Types of infinity
    - Applications: is language infinite?
3. Relations and orders
    - Topics: properties of orders, posets, lattices, antimatroids
    - Applications: mereology, string extension learners, OT, feature systems
4. Linear algebra
    - Topics: vectors and vector spaces, matrices, tensor product
    - Application: vector space semantics, spatial semantics, inflectional morphology
5. Abstract algebra
    - Topics: monoids, groups, semirings
    - Application: violation semirings in OT, semiring parsing
6. Graph theory
    - Topics: (un)directed graphs, connectedness, components
    - Application: morphological paradigms, parse forest representation, autosegmental phonology
7. Automata theory
    - Topics: finite-state automata and transducers, regular expressions, push-down automata
    - Application: complexity of phonology & morphology VS syntax
8. Logic
    - Topics: propositional logic and first-order logic

    - Application: semantics, model-theoretic syntax, subregular linguistics
9. Probability theory
    - Topics: calculating probabilities with addition and multiplication
    - Application: weighted context-free grammars, corpus-based techniques
10. Information theory
    - Topics: entropy, cross-entropy
    - Application: probabilistic machine learning

## Teaching Goals

- master essential concepts and techniques in mathematics and theoretical computer science
- apply mathematical techniques to the study of language
- formalize linguistic ideas in mathematical terms
- develop learning autonomy and the ability to expand your mathematical knowledge through self-study

## Grading

This course can only be taken for 0 or 3 credits. Student grades are determined by 4 components:

1. **Class participation (30%)**
   While physical attendance is mandatory only for exercise sessions, students are expected to participate actively in the class, both offline and online. This includes:
    - providing feedback on lecture materials
    - helping other students during exercise sessions
    - asking questions on github, and helping other students with their questions
2. **Presentation of homework solutions in exercise sessions (30%)**
   Students must present their solutions on the board during exercise sessions. Performance is evaluated not on correctness but the student's ability to explain their thought process. An incorrect answer that clearly outlines how the student proceeded, where they got stuck, and why, is worth more than a correct one where the student can't explain their path from the question to the answer.
3. **Final oral exam (30%)**
   At the end of the semester students take an oral exam in groups of 2 or 3. Each student gets a different exercise and must present their solution on the board while the other student asks clarification questions. The format thus resembles that of exercise sessions. Performance is evaluated based on the correctness of the solution, the clarity of presentation, and the questions asked during the other student's presentation.

4. **Taking the initial assessment (10%)**
   At the beginning of the semester, students are asked to take a survey to assess their mathematical aptitude. Participation is worth 10 percentage points.

## Policies

### Contacting me

- Emails should be sent to [coursenumber]@thomasgraf.net. Disregarding this policy means late replies and is a sure-fire way to get on my bad side.
- Reply time $<$ 24h in simple cases, possibly more if meddling with bureaucracy is involved.
- If you want to come to my office hours and anticipate a longer meeting, please email me so that we can set apart enough time and avoid collisions with other students.

### Disability Support Services

If you have a physical, psychological, medical or learning disability that may impact your course work, please contact Disability Support Services, ECC (Educational Communications Center) Building, Room 128, (631) 632-6748. They will determine with you what accommodations, if any, are necessary and appropriate. All information and documentation is confidential.

Students who require assistance during emergency evacuation are encouraged to discuss their needs with their professors and Disability Support Services. For procedures and information go to the following website: http://www.stonybrook.edu/ehs/fire/disabilities

### Academic Integrity

Each student must pursue his or her academic goals honestly and be personally accountable for all submitted work. Representing another person's work as your own is always wrong. Faculty are required to report any suspected instances of academic dishonesty to the Academic Judiciary. Faculty in the Health Sciences Center (School of Health Technology & Management, Nursing, Social Welfare, Dental Medicine) and School of Medicine are required to follow their school-specific procedures. For more comprehensive information on academic integrity, including categories of academic dishonesty, please refer to the academic judiciary website at http://www.stonybrook.edu/uaa/academicjudiciary/

### Critical Incident Management

Stony Brook University expects students to respect the rights, privileges, and property of other people. Faculty are required to report to the Office of Judicial Affairs any disruptive behavior that interrupts their ability to teach, compromises the safety of the learning environment, or

inhibits students' ability to learn. Faculty in the HSC Schools and the School of Medicine are required to follow their school-specific procedures.

## Link List

### Using git

- Github app for Windows; supports only Windows 7 or later
- Github app for Mac; supports only OS X 10.9 or later
- List of alternative GUI clients for git
- Tutorials for using git via the command line
- Official documentation for git

### Markdown

- Interactive tutorial to markdown basics
- Complete markdown syntax
- Overview of Github's markdown dialect