MATH478: Computational Mathematics

Syllabus for Fall 2019

Instructor:	Longfei Li
Email:	longfei.li@louisiana.edu (best way to contact me)
Lecture:	MWF 09:00 AM - 09:50 AM, Maxim Doucet Hall 212
Office Hours:	MWF 10:00 AM - 10:50 AM or by appointment, Maxim Doucet Hall 433

Academic Dishonesty: If I believe a student is performing academic dishonesty, I will collect evidence as I see fit and it will be handled according to the University's code of conduct found at http://studentrights.louisiana.edu/student-conduct/code-conduct.

Course Description: basics tools for computational math (i.e., unix-like shell scripting, version control system (git and github), etc.); machine arithmetic, computer architecture (i.e., floating point arithmetic etc.) and memory hierarchies; introduction of MATLAB, python, R and their applictions in computational mathematics; solving system of linear equations (LAPACK/BLAS, petsc, etc.), data analysis and machine learning (tensorflow) and parallel computation using MPI (if time permits).

Learning Outcomes: Successful completion of the course should enable you to have a basic understanding of computational mathematics. In addition, you should be able to have a basic working knowledge of computational tools (such as MATLAB, python and R) and have the ability to use these tools to solve simple computational problems.

References (there is no required text):

1. *Learning MATLAB*, Tobin A. Driscoll.
2. *Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib (2nd Edition)*, Robert Johansson.
3. *Learning R: A Step-by-Step Function Guide to Data Analysis*, Richard Cotton.
4. *An Introduction to Statistical Learning: with Applications in R*, Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani.
5. *High Performance Scientific Computing (AMATH 483/583 course materials, Spring Quarter, 2014, University of Washington)*, Randall J. LeVeque.
6. *The Python Tutorial*, Guido van Rossum and the Python development team.
7. Fortran 90 Tutorial, Michael Metcalf.
8. Numerous resources are avaible online for free, Google.

Online Resources: Both Moodle and github are utilized to post and distribute class materials. The Moodle page for the course is used to post class announcement, homework assignment and grades. A git repository is created on github.com to host the demo codes and other class notes. The class repo can be accessed using "git clone https://github.com/longfeili86/math487.git"

Software and Computer: We use a Unix-like operating system (e.g., Linux, Mac OS X); learning Unix commands is part of this class. We also need many open-source libraries and software (see installation list for detail). You may choose to install all the needed software by yourself if you have a Unix-like computer. Alternatively, a linux virtual machine (VM) preloading all the necessary software packages is created; the VM is the easiest way to get started with the class.

Homework: Programming homework will be assigned biweekly. Homework consists 30% of the final grade.

Project: There are two projects (midterm and final); each consists 30% of the final grade. A typed report with necessary codes, figures and explanations should be turned in for each project. No report will be accepted after due date unless a written documented excuse is provided.

Attendance and Participation: Attendance and participation are expected for the class. I will take attendance at the beginning of randomly selected lectures. Absense will be recorded unless a written documented excuse is provided. A 2% penalty will be applied to your final grades if you have more than 3 absences.

Grading Mistakes: Please check your graded work when returned to you. If you believe there is a grading mistake, you can ask me for a review within **24 hours**. Scores may be adjusted at my discretion. No points will be adjusted under the following circumstances: (i) the graded paper is returned to you more than 24 hours and (ii) the graded paper is modified by you.

Final Grading Weights:

Attendance & Participation	Homework	Midterm Project	Final Projects	Total
10%	30%	30%	30%	100%

Grading Scale:

Grade	A	B	C	D	F
Percentage	90% – 100%	80% – 89 %	70% – 79 %	60% – 69 %	<60%

Tentative Lecture Plan (subject to change):

Lecture	Topic
1 – 5	Basics tools commonly used by the computational math community (shell scripting, git & github, working habits for documentation and reproducibility of the results). Knowledge about machine arithmetic, computer architecture (i.e., floating point arithmetic etc.), and memory hierarchies.
6 – 7	Language issues, compiled vs. interpreted. Demo using MATLAB, Python and fortran90.
8 – 24	Focusing on python. Introduction of numpy, scipy and matplotlib. Using python to solve problems in computational mathematics (numerical PDE, linear algebra, etc). MATLAB is also discussed for comparison purpose.
25 – 35	Statistical learning and introduction of R (linear, logistic regression, classification, etc.).
36 – 38	Introduction of machine learning (ternsorflow in python and R).
39 – 41	Introduction of parallel computation (MPI in python and C++).