Math 446, Summer-B01 / B02, 2022
Course Summary

Suddhasattwa Das*

June 25, 2022

Instructor. Suddhasattwa Das (Shuddho),
Here is my email sdas27@gmu.edu and website.

Class location. The entire course will be taught asynchronously.

Semester span. This Summer B session lasts from June 07 to July 24. You can find the GMU semester schedule here .

Office hours. Monday, Tuesday, Wednesday, 9:00–10:00 pm

Text-book. The prescribed textbook for this course is Tim Sauer's "Numerical Analysis". However, all the course content will be presented and also uploaded. The main focus of this course will be on numerical projects.

Course website. Please find this course in your list of registered courses on Blackboard, https://mymasonportal.g
The course is listed as sections B01 and B02 respectively.

Books and calculator policy. You are allowed to use non-programmable / non-graphing calculators for your exams, quizzes etc, although the problems will be set so that there is not much computation involved.

Software requirements. This course is meant for you to acquire advanced computation skills. Computation and programming is an integral part of this course. The demonstrations in the class will only be in **Julia**. However, students are allowed to submit their computation based assignments in **any** of the following languages

(i) Julia : open-source, high precision, high-speed, most-modern software.
Read more here : https://julialang.org/
Installation links : https://julialang.org/downloads/

(ii) Matlab : a commerical computation software, available on campus in the computer labs (if you are on campus). Students can also purchase the software for their personal computers. There is also a **free student version** which GMU students can access by logging in using their GMU email and password. There are plenty of Matlab tutorials on the internet. There is a comprehensive tutorial at Mathworks, and another one in the textbook's appendix.

*Department of Mathematical Sciences, George Mason University, Virginia, USA

(iii) Python : open-source software, used in majority of industrial applications : https://www.python.org/downloads/
All the above languages are powerful computation softwares with good libraries and graphing capability.

(iv) C, C++ : users of these languages need to find the suitable linear algebra packages, and graphing / plotting.

Some of you may have had programming experience before in one or more of the above languages. You are welcome to choose a language of your liking. If you have had no prior experience with programming before, I would recommend Julia. It is syntactically simple and high level like Python or Matlab. However, it is the fastest of all of the above languages. It is also both compiled and interpreted.

Grade distribution. You will be evaluated based on your performance on 5 homework projects, 1 midterm and 1 final exam, and mini-quizzes pertaining to each lesson. Homework will be given regularly, **but not graded**.

Category	% of total score
Mini-quizzes	20
Projects(5)	70
Final Exam	10
Total	100

Homework policy. Let us abide by these policies throughout the course.

1. Students are supposed to submit only their original work, codes, and program-outputs for their homework. You are encouraged to discuss and help each other, I see it as a part of the learning process. However the text and code in your homework submission must be your own composition.

2. There will be regular problems given as homework which will not be graded. You are expected to try to solve them on your own. The homework that will be graded is the computation projects. They will be posted at least one week in advance. Late submissions will be penalized.

3. The computational projects are meant to provide you a more creative way of doing coursework, and explore and learn at the same time. Instead of treating them as routine exercise based homeworks, try to see them as your first steps in numerical investigation and research.

4. In your project assignments, if you are asked to implement an algorithm, you need to implement all the steps of the algorithm yourself. A common mistake is to call / execute ready made functions from existing libraries. This will not fetch any points.

Course content. The course will cover portions of Chapters 1-4 of the book mentioned above. Here is the breakdown :

1. Floating point arithmetic :

   1.1 Binary and decimal representations, conversion from one to another.

   1.2 Representation of real numbers

   1.3 Floating point arithmetic.

   1.4 Loss of significance during arithmetic operations.

2. The solution of nonlinear equations in one variable
    2.1 Bisection method :
    2.2 Fixed point iteration :
    2.3 Newton's method :
    2.4 Secant method :
    2.5 Method of false position
    2.6 Inverse quadratic interpolation, Brent's method
3. Error analysis
    3.1 Forward and backward error
    3.2 Multiple roots
    3.3 Cause for large and small backward and forward errors.
4. The solution of systems of linear equations
    4.1 Gaussian elimination : technique, complexity.
    4.2 LU factorization : connection with Gaussian elimination; back-substitution; complexity.
    4.3 LUP factoriations : permutation matrix, partial pivoting.
    4.4 Complexity of algorithms
    4.5 Condition number : sources of error; swamping.
    4.6 Symmetric, positive definite matrices.
5. Least squares problems
    5.1 Inconsistent systems of equations
    5.2 Column space and rank
    5.3 Least Squares and the Normal Equations
    5.4 Gram–Schmidt orthogonalization and least squares
    5.5 QR Factorization
    5.6 Generalized Minimum Residual (GMRES) Method
6. Nonlinear least squares problems
    6.1 Gauss–Newton Method
    6.2 Models with nonlinear parameters
    6.3 Levenberg–Marquardt Method
7. The solution of nonlinear systems
    7.1 Multivariate Newton's Method
    7.2 Broyden's method (time permitting)
8. Interpolation and polynomial approximation
    8.1 Data and Interpolating Functions

8.2 Lagrange interpolation

8.3 Application of Linear least squares to interpolation

8.4 Interpolation Error - various formulas

8.5 Splines - cubic splines, properties, endpoint criterion

8.6 Bézier Curves (time permitting)

Weekly schedule. Our curriculum is going to be divided on a weekly basis. Here is the intended weekly schedule.

Week	Date-span	Lectures	Section	Exams / Projects due
1	06/06 – 06/10	1–6	1, 2.1–2.3, 3	
2	06/13 – 06/17	7–12	2.4–2.6, 4.1	Project 1
3	06/20 – 06/24	13–17	4.2–4.3, 5.1	Project 2
4	06/27 – 07/01	18–22	4.4–4.5, 5.2–5.3	Project 3
5	07/4 – 07/8	23–26	4.6, 5.4–5.5	
6	07/11 – 07/15	27–30	8.1–8.5	Project 4
7	07/18 – 07/22	31–33	6, 7.1	Project 5
8	07/25 – 07/26	34	All	Final Exam

List of important dates. Here are the various projects and exams, along with their due dates.

Event	Relevant	Date-available	Date due
Project 1	Root finding	06/09	06/19
Project 2	Solutions to linear systems	06/17	06/26
Project 3	Methods for symmetric matrices	06/04	07/03
Project 4	Least Squares	07/03	07/15
Project 5	Nonlinear least squares	07/13	07/22
Final Exam	1–7	07/26	07/26