MA 3257C SPRING 2026
NUMERICAL METHODS
FOR LINEAR AND NONLINEAR SYSTEMS
COURSE SYLLABUS

DR. BINAN GU
DEPARTMENT OF MATHEMATICAL SCIENCES
WORCESTER POLYTECHNIC INSTITUTE

Contents

1. Contact Information

Name	Email	Office Hours	Delivery
Dr. Binan Gu	bgu@wpi.edu	See Canvas; additional meetings by appointment	Zoom `https://wpi.zoom.us/j/4371273067`

2. Course Description

This course provides an introduction to modern computational methods for the numerical solution of systems of linear and nonlinear equations and their applications. Topics covered include numerical precision and error, direct and iterative algorithms for systems of linear equations, least-squares problems, eigenvalue approximation, scalar and multidimensional root-finding, and introductory numerical optimization. Both theoretical understanding of methods and their practical performance will be stressed. Computation is an essential component of the course, and there will be frequent computing assignments.

This summer offering is a compressed five-week version of a standard seven-week course. Lecture delivery is primarily asynchronous through short videos and assigned lecture notes. The weekly Discussion is live and synchronous, and it will be used as a coached coding session tied directly to the homework.

Recommended background: MA 2071 and some prior programming experience. Familiarity with MATLAB will be helpful. Students should enter the course with basic working knowledge of matrix-vector multiplication, matrix-matrix multiplication, solving small systems of equations, convergence of sequences and series, and basic calculus. A short preparatory exercise sheet will be posted on Canvas to indicate the expected technical baseline.

3. Textbook and Lecture Notes

Main textbook: *Numerical Linear Algebra* by Folkmar Bornemann, freely available on Springer under WPI wireless network.

The text is intended to serve as a background reference on most but by no means all of the material covered in class. **The material covered in the lecture videos, lecture notes, homework, and discussion coding work constitutes the course content for which you will be responsible.**

Supplementary textbooks:

(1) *Linear Algebra and Learning from Data* (2019) by Gilbert Strang, available here.
(2) *Numerical Analysis* (2010) by Richard Burden and Douglas Faires, Edition 9.

4. Course Objectives

4.1. **Goal for Each Topic.** For each topic, your main job is to understand

- why this topic is introduced and studied;
- what its fundamental connections are to prior knowledge in linear algebra, calculus, and programming;
- the chronology of the topics introduced in this class and the logic from one topic to the next;
- how the numerical method behaves in implementation, not only how it is written on paper.

4.2. **Main Topics.** This course explores the use of algorithms to solve systems of linear and nonlinear equations. The technical contents include:

- numerical precision, floating-point arithmetic, round-off error, cancellation, conditioning, stability, and convergence;
- direct methods for linear systems: Gaussian elimination, partial pivoting, LU/PLU decomposition, Cholesky decomposition;
- least-squares problems and normal equations;
- iterative methods for linear systems: Jacobi, Gauss-Seidel, SOR, residuals, convergence criteria, and spectral radius;
- eigenvalue approximation through the power method;
- scalar and multidimensional root-finding: fixed-point iteration, Newton's method, secant method, bisection, and safeguarded Newton;
- introductory numerical optimization: gradient descent, line search, Armijo backtracking, and a brief application such as logistic regression or stochastic gradient descent if time permits.

Meanwhile, students will develop their scientific communication skills and computational habits, including:

- clear and concise use of mathematical language;

- responsible coding etiquette, including commenting, function documentation, reproducibility, and readable deliverables;
- clear presentation of figures, including axis labels, legends when needed, informative captions, and scientifically meaningful formatting;
- understanding of accuracy, speed, stability, conditioning, and convergence of algorithms.

4.3. **Goals.**

4.3.1. *Course Goals.* Two end goals are measured in this class:

(1) **Understanding:** the ability to recall definitions, provide motivating examples, interpret numerical behavior, and argue with mathematical logic.
(2) **Implementation:** the ability to carry out important algorithms by hand for small examples and implement them correctly in code for larger examples.

4.3.2. *Personal Goals.* There are a few personal goals I wish everyone achieves by the end of the class.

(1) **Learn how to learn.** This includes:
  (a) forming good learning habits:
    (i) consistent contact with the course material;
    (ii) starting homework and coding assignments early;
    (iii) reading the relevant lecture notes before watching videos when possible;
    (iv) reviewing lecture notes and examples after the videos;
    (v) using office hours and Discussion before small confusions become large failures.
  (b) eradicating bad learning habits:
    (i) giving up too quickly before using all available permitted resources;
    (ii) immediately looking for solutions online;
    (iii) using AI to bypass understanding;
    (iv) procrastinating until the due date;
    (v) cramming before an exam.
(2) Learn about yourself: what are your strengths and weaknesses, academically and mentally, before, during, and after this class?

5. Course Organization

This course is delivered primarily asynchronously. Each week, a Canvas module will contain lecture videos, lecture notes, discussion information, homework, and other supporting material. The live Discussion is synchronous and will be used mainly as a coached coding lab. Students are responsible for all material assigned in videos, lecture notes, homework, and Discussion.

5.1. **Lecture Videos.** Instead of long lecture recordings, the course will use short videos. A typical week will contain approximately 10 videos, each around 10–15 minutes. These videos are meant to introduce and organize the material, demonstrate examples, and prepare students for homework and Discussion. The videos will not cover every detail in the lecture notes. Students are responsible for reading the assigned lecture notes carefully.

The compressed summer schedule means that the videos are only the beginning of the work. Mastery comes from coding, written problem solving, debugging, revision, and explaining your work.

5.2. **Weekly Topic Schedule.** The following schedule is tentative and may be adjusted as needed.

Week	Main Content
1	Numerical precision and direct methods: floating-point arithmetic, error, cancellation, conditioning/stability intuition, Gaussian elimination, triangular solves, partial pivoting, LU/PLU, and Cholesky.
2	Least squares and first iterative methods: normal equations, norms, matrix condition number, Jacobi iteration, Gauss-Seidel iteration, residuals, and stopping criteria.
3	Advanced iterative methods and eigenvalue approximation: SOR, residual correction viewpoint, iteration matrices, spectral radius criterion, convergence rate intuition, and the power method. Oral Exam I occurs near the end of Week 3.
4	Root-finding methods: fixed-point iteration, Newton's method, secant method, bisection, safeguarded Newton, and multidimensional Newton.
5	Optimization and synthesis: gradient descent, exact and inexact line search, Armijo backtracking, and a brief capstone application such as logistic regression or stochastic gradient descent if time permits. Oral Exam II occurs near the end of Week 5.

5.3. **Practice Problems.** The majority of your association with this course lies in problem solving outside the videos. Students are expected to take the intuition gained from lecture videos and put it into careful examination by doing assigned and suggested practice problems. Practice problems form the main technical preparation for homework, Discussion, and oral exams. Solutions to suggested practice problems are not necessarily collected, but they indicate the level of fluency expected in the course.

5.4. **Discussion.** The weekly Discussion is a live, synchronous coding session. Students will work through preliminary coding tasks related to the current homework, such as implementing a core loop, testing a simple case, plotting convergence, or debugging a numerical method. The Discussion is not merely optional extra help; it is part of the design of the summer course.

In each Discussion session, students are expected to attend, participate, write code, ask questions when stuck, and submit the required Discussion deliverable. The Discussion deliverable is typically due on the day of Discussion and is intended to document reasonable engagement during the live session.

5.5. **Expected Weekly Workload.** This is a five-week compressed course. Students should plan for approximately 15–20 hours of total work per week. A realistic weekly allocation is:

- 2–3 hours watching lecture videos and taking notes;
- 2–3 hours reading lecture notes, reviewing examples, and completing preparatory practice;
- 1 hour attending live Discussion;
- 8–12 hours completing homework and coding assignments;
- 1–2 additional hours in weeks with oral exam preparation.

Students who begin the weekly homework late should expect the course to become substantially harder. Numerical coding assignments almost always require debugging time, and debugging time cannot be reliably compressed into the last evening before a deadline.

5.6. **Canvas.** Course materials can be found on the Canvas page. Each weekly module will contain the lecture videos, lecture notes, homework assignment, Discussion deliverable, and other supporting files. Zoom links for live Discussion, office hours, and oral exams will be posted on Canvas.

5.7. **Communications.** The primary interface for communication with the instructor and course staff will be email, the Canvas course website, office hours, and Discussions. All information about the course will be maintained on the course Canvas page. Check it often.

Check your WPI email daily. Students can expect a response to email within 24 hours on weekdays and within 48 hours on weekends.

6. Assignments

6.1. **Written and Coding Homework.** Written and coding homework weigh **considerably** in this course (see Grades). Take them seriously. These assignments involve mathematical solutions, algorithmic explanation, and computational implementation. Written solutions should be **second drafts** and thoroughly demonstrate your thought process, including justifications of steps. Code submissions should be organized, commented, reproducible, and accompanied by readable outputs when requested.

There will typically be one homework assignment per week. Because this is a compressed five-week course, each homework assignment contains a substantial amount of material.

6.1.1. *Collaborations and References.* Collaboration among classmates is welcome. Every homework submission must declare a list of collaborators at the beginning, though each student is expected to compose his/her own solutions and write his/her own code. **Extensive similarities between assignments without collaboration declarations will be considered plagiarism and will be penalized severely.**

If a solution depends on online sources, textbooks, AI tools, or other references, the student is responsible for citing them with a reference style of the student's choosing. Uncited solutions, if confirmed, will be considered plagiarism and will not receive credit. Please review AI Policy at the end of the syllabus.

6.1.2. *Return-and-Correct Policy.* Each student has 1 day to correct his/her solutions after homework is returned. Resubmissions may be graded without penalty if the initial submission has sufficient effort in solving **every** problem. Poor attempts at the initial submission will receive little credit and no resubmission opportunities.

Every resubmission shall contain the original copy or state clearly which problems you are correcting. You must address every query for the correction to be considered valid. The corrections should be written using a different colored pen/pencil, either in a side column clearly marked for short corrections, or on new sheets of paper for long corrections.

6.1.3. *File Type.* The initial submission and later corrections are to be submitted as a **single PDF file** via the Canvas assignment module unless otherwise specified. When code files are required, they should be submitted in the requested format on Canvas. If the assignment asks for figures, tables, or written interpretation, these must be included in the PDF submission.

6.1.4. *Coding Language and Software Requirement.* The primary coding language is MATLAB. Students should have a working and up-to-date MATLAB installation ready by the first week of the course. You should be able to create scripts and functions, run loops, use arrays and matrices, make plots, save figures, and submit code files through Canvas.

Given the various backgrounds of students in this class, I am open to other programming languages, including C/C++, Python, Julia, and Fortran. If you decide to use a computing language other than MATLAB, then it is **your responsibility** to make your deliverables readable to the instructor and TA by:

(1) commenting more diligently on syntax or usage not available in MATLAB;
(2) submitting deliverables, such as plots and tables, in a scientifically readable fashion;
(3) ensuring that your code can be run and checked without special undocumented setup.

Failure to follow basic scientific coding etiquette may result in rejection of the code submission and deductions in homework grades.

6.1.5. *Coding Etiquette and Deliverable Standards.* For all coding assignments, students are expected to follow these standards:

- code should run without manual editing by the grader unless otherwise stated;
- scripts and functions should have clear names;
- functions should include a short comment describing inputs, outputs, and purpose;
- important lines or blocks should be commented, especially loops implementing numerical algorithms;
- figures should have clear axis labels, titles or captions, legends when appropriate, and readable tick labels;
- convergence plots should use appropriate scales when requested, such as semilog plots for residual decay;
- tables should have clear column labels;
- random experiments should set a seed when reproducibility matters;
- written interpretation should explain what the computation demonstrates, not merely show code output.

6.1.6. *Grading Rubric.* For each homework, selected problems will be graded according to the following criteria.

Type	Submission Qualities
Presentation	Quality is neat and easily readable. Ample spacing between each problem. Figures and tables are readable and labeled.
Completion	Every problem has a committed attempt. Required code and outputs are included.
Syntax	Correct use of basic mathematical symbols and clear use of computational notation.
Correctness	Correct, clear, and thorough solution, including appropriate mathematical justification, algorithmic explanation, and computational interpretation.

6.1.7. *Poor Work Policy.* The instructor and TA reserve the right to reject homework with poor presentation or careless completion without further review of its content. The first rejection acts as a warning, and its resubmission incurs a penalty of 15% of total points; any homework rejection after the first offense automatically receives a score of zero.

6.1.8. *Late Work Policy.* Any late homework follows a "15% of total points off per day late submission" policy unless a valid excuse is supported with proper documentation.

6.2. **Oral Code-Defense Exams.** There will be two oral code-defense exams, one near the end of Week 3 and one near the end of Week 5. Each oral exam is worth 25% of the final grade.

The oral exams are designed to check individual understanding in an asynchronous, coding-heavy course. Students may be asked to explain mathematical ideas, walk through an algorithm, interpret convergence behavior, identify an error in code, modify a short piece of code, or defend part of a submitted homework assignment. These exams are not intended to reward memorized speeches. They are intended to verify that students understand the methods they have implemented.

Each oral exam will be conducted through Zoom. Students must be prepared to share their screen, open their submitted code and written work when requested, and answer questions without outside help. Use of AI tools, messaging platforms, or unauthorized assistance during an oral exam is prohibited.

6.2.1. *Oral Exam Rubric.* The oral exams will be graded according to:

- conceptual understanding of the numerical method;
- ability to explain the algorithm step by step;
- ability to interpret code and numerical output;
- ability to connect observed behavior to error, conditioning, stability, convergence, or residuals;
- ability to make small modifications or corrections when prompted;
- clarity and honesty in communication.

6.2.2. *Makeup Policy.* Unless there are extenuating circumstances, no makeup oral exams are allowed. Students requesting oral exams at times other than the proposed scheduling window must provide valid reasons with proper documentation, such as medical, familial, or religious reasons, or give notice at least one week in advance. A temporary grade of Incomplete may be assigned until all makeup requirements are finished.

6.3. **Discussion Coding Work.** In each Discussion, students will sit down and code preliminary testing examples that help explain the underpinnings of numerical methods. Deliverables of each Discussion include written code and outputs, such as a convergence plot or short numerical report. The main goal of collecting these works is to verify that students attended the Discussion and engaged in reasonable work during the allotted time.

Attendance plus reasonable work during each week's Discussion earns 2% of the final grade. There are five Discussion sessions, for a total of 10%.

7. Grades

Written and Coding Homework: 40% = 4 × 10%.

Discussion Coding Work: 10% = 5 × 2%.

Oral Code-Defense Exam I: 25%.

Oral Code-Defense Exam II: 25%.

Tentative Final Grades: A (> 90), B (80–89), C (70–79), NR (< 70).

Remark. There is no need to ask me to round up your grade, as I will consider this option for everyone. Rounding up to a higher letter grade is only possible when the following conditions are simultaneously met: 1) your current numerical grade is at most 1.00 point away from a letter grade change, up to two digits past the dot, e.g. 89.00–89.99 but not 88.99; 2) your performance does not show a major decline near the end of the course; 3) you complete all homework and Discussion requirements. Since these conditions tie directly to your well-being in the course, I will not consider any other criteria to round up your grade.

8. Special Arrangements

If you need course adaptations or accommodations because of a disability, or if you have medical information to share with me, please make an appointment with me as soon as possible. If you have not already done so, students who believe that they may need accommodations in this class are encouraged to contact the Office of Accessibility Services (OAS) as soon as possible to ensure that these accommodations are implemented in a timely fashion. The OAS is in Unity Hall, (508) 831-4908. Students who need accommodations for oral exams should contact OAS and the instructor early so that appropriate arrangements can be made.

9. Additional Policies

9.1. **Electronic Policy.** Instructor-produced lecture videos and course materials are for enrolled students in this course. Students may not distribute course videos, Zoom links, homework solutions, or Discussion materials outside the course. **No recording** of audio or video by students is allowed during live Discussion, office hours, or oral exams unless explicit permission is granted.

During live Discussion, laptops are expected and should be used for course-related coding work. During oral exams, students must follow the exam instructions and may not use unauthorized electronic resources.

9.2. **Exam Policy.** Oral exams are individual assessments. Students must complete the oral exams without unauthorized assistance. Use of AI tools, communication with other people, hidden notes not permitted by the exam instructions, or any other attempt to misrepresent understanding is an academic integrity violation. Students involved in plagiarism or cheating will be reported according to WPI academic integrity policies.

9.3. **AI Policy.** Artificial intelligence (AI) language models and online homework-help platforms are widely available. Examples include ChatGPT (OpenAI), Claude (Anthropic), Gemini (Google), Microsoft Copilot, Perplexity, as well as platforms such as Chegg®, Course Hero, and Brainly. Use of these or similar tools to generate assignment solutions is not permitted unless you follow the requirements below. This policy applies to all current and future AI-powered systems, whether or not they are explicitly listed here. In this course, you must not submit AI-generated or platform-generated solutions verbatim as your own work.

If you use AI tools as part of your learning process, you **must:**

(1) **cite the tool and provide a direct link** to the generated response at the end of your submission, so the grader is one click away; written links do not count;
(2) **explain in your own words** how you used the output to inform your solution, e.g. what you agreed with, what you adapted, and what you corrected;
(3) **show your full reasoning and intermediate steps**, not just a final answer;
(4) understand and be able to explain any code or mathematical solution you submit.

Submissions that do not follow these rules may receive zero credit. If the instructor or TA has reason to believe that a submission does not represent the student's own understanding, they may withhold the assignment grade and require the student to complete an oral check to demonstrate mastery. Failure to demonstrate sufficient understanding will result in a failing grade on the assignment and may be reported as academic dishonesty under the WPI policy "What is Academic Dishonesty".