DS288 3:0 (UMC202 3:1) Numerical Methods
Ratikanta Behera August 2025
Indian Institute of Science, Bangalore, India.

Instructor

◇ **Dr. Ratikanta Behera**

Office and Contact

◇ Room Number 317 (CDS)

◇ Office Hours 10:00 AM to 11:00 AM (Tuesday and Thursday) or by appointment only

◇ Contact: E-mail: ratikanta@iisc.ac.in, please use the subject line `[DS288 or UMC202] ...`

◇ Phone: +91-80-2293-2317

Teaching Assistants

◇ Himanshu Pandey (PhD Student, NATL lab CDS-226),
Meeting time: 10:30 AM to 11:30 AM (Saturday) or by appointment only.
E-mail: phimanshu@iisc.ac.in

◇ Aneesh Panchal (PhD Student, NATL lab - CDS-226),
Meeting time: 10:30 AM to 11:30 AM (Saturday) or by appointment only.
E-mail: aneeshp@iisc.ac.in

◇ Ghodake Shubham Sambhaji (PhD Student, NATL lab CDS-226),
Meeting time: 10:30 AM to 11:30 AM (Saturday) or by appointment only.
E-mail: sgshubham@iisc.ac.in

◇ Meet Jiten Thakkar (PhD Student, NATL lab CDS-226),
Meeting time: 10:30 AM to 11:30 AM (Saturday) or by appointment only.
E-mail: meetjitent@iisc.ac.in

◇ Subham Patel (PhD Student, NATL lab CDS-226),
Meeting time: 10:30 AM to 11:30 AM (Saturday) or by appointment only.
E-mail: subhampatel@iisc.ac.in

Class Meetings

◇ 08:30 AM - 10:00 AM (Tuesday and Thursday)

◇ **Location**: CDS 102

◇ **MS Teams Link**: `https://teams.microsoft.com/l/team/19%3As5hotRJ6mhujYc5glfjgIzVb1W6MYBhHSh`
`EVqrE1%40thread.tacv2/conversations?groupId=e184e21c-83ba-45b4-acd6-893404553424&`
`tenantId=6f15cd97-f6a7-41e3-b2c5-ad4193976476`

Text Book

◇ Richard L. Burden and J. Douglas Faires, **Numerical Analysis** (9th edition).

Course Outline

◇ Numerical methods are routinely used in all disciplines of science and engineering, which serve as the approximate solutions to real-time problems. This course will cover the formulation/mathematics behind the contemporary numerical methods, which later will be applied to solving the example real world problems. The emphasis of the course will be more on the deep understanding of these numerical methods, with homework geared towards application and analysis of this. Midterm and final exam will be geared towards testing your understanding of advantages/limitations of numerical methods.

Grading

◇ Project/Homework: 50%; Midterm Exam: 20%; and Final Exam: 30%.

◇ Midterm: September 22-27, 2025. Final: November 21 to December 02, 2025. More details

Project/Homework	Weight 50%	Due dates/time
Homework - 1	10 %	**21-08-2025** (11 : 59 *PM*)
Homework - 2	10 %	**16-09-2025** (11 : 59 *PM*)
Homework - 3	10 %	**16-10-2025** (11 : 59 *PM*)
Final Project - 1	20 %	**16-11-2025** (11 : 59 *PM*)

Project or Homework: ◇ There will be **four** homework/project problem sets with a specific due date and time. All homework/project problems require computer programming (preferably in MATLAB/Python). Each student will submit the solutions via MS Teams by the due date and time, meet the instructor to discuss the solutions for the homework problems after the due date. Solutions will be discussed in class after the evaluation. All homework problems require computer programming. **Late posting/submission (beyond the due date and time) of homework solutions/project report will result in no credit.**

Midterm or Final Exam: ◇ Both midterm and final exam will test primarily your understanding of advantages/limitations of numerical methods covered in the class.

Honor Principle: ◇ You are welcome to exchange ideas in solving homework problems with your colleagues, but all the work submitted for grading (homework/project, and midterm/final exam) must be your own work (i.e., you must have worked out all details by yourself). Copying computer code or files (including the material on the web) without proper citation is considered as plagiarism. Any deviations from this principle will result in failing of the class.

Homework Tips

Homeworks are designed, in part, to increase your confidence in programming and therefore, generally increase in programming complexity. It is, therefore, a good idea to begin practicing good programming style on the first homework. These suggestions may also save your time.

- Start Early (key point): Start looking into homework as soon as it has been posted. It is never too early to start looking at the assignment. Begin by reading through the homework and figuring out what aspects of the lecture we are implementing and testing.
- Write Pseudo Code: It is really is a bad idea to sit down in front of a computer terminal and pull out the assignment for the week (especially on Monday or Tuesday night!) You will save many hours of frustration if you first sit down with pen and paper and write down your ideas first. Begin by writing the general tasks that must be performed by your code and then fill in the details of each task.
- Modularize Your Code: Break down the problem in modules and write separate pieces of code (functions) to perform specific tasks. Not only will your final code be much easier to debug, you can also reuse a lot of the functions you create, thereby minimizing the amount of coding necessary to finish the assignment. While it is always a good idea to program in this manner, the idea of modularizing your code will become increasingly important on the remaining assignments.

Suggestions for Homework Write-ups (submissions for grading)

The following are some suggestions and tips to keep in mind when writing up your homeworks. Although the course does not stress formal write-ups for the homework, remember that you should be thinking about how you can present your work in a way that is clear and makes you explain your work easily. So aside from the obvious task of getting all the answers correct here are some things that you can do to increase the likelihood that you make the instructor as well as your life easy while grading your homework:

- Make everything as a single PDF (answers to all parts of homework should be in a single PDF). With many documents to grade, you will have less time to explain, so it is important to have an only one document. (see minimizing pages below).
- Code (.m or .py):
  - Use an editor to format your code for printing (vi, emacs, AbiWord, MSWord, etc.)
  - Use size eight font (we can read it).

  - Format into two columns (optional, but desirable).

- Results: Do not just print your Matlab output! The direct output from Matlab is not suitable for printing. Copy the output into an editor and tidy it up, i.e., tabulate data, etc.

- Plotting: Print multiple plots per page. Use the subplot(m,n,k) command in Matlab, where m is the number of plots per row, n is the number of plots per column, and k specifies the k th plotting window.

- Box your answers or clearly mark or highlight them. Make it easy for everybody to find your answers.

- Group your answers. Put your analysis for a particular problem up front, followed by your clearly marked answers (see above), then include any supporting code (see above for code for- matting). Then move on to the next question.

Just a reminder that all work must be your own.