

Mathematical modeling and simulation: Practical matters

1.1 General

Here are the practical matters concerning the course. Please take the issues regarding e-mail and office hours seriously.

- Textbook: "An Introduction to Computer Simulation Methods" by Harvey Gould, Jan Tobochnik, and Wolfgang Christian. Required.
- Classes: Thursdays 14:30-17:30, room VAC100
- In-class exams: 1: Thu Jan. 29, 2: Thu. Feb. 26, 3: Thu. Mar. 19
- Classes finish: Apr. 9, 2009
- Conference week: Feb. 16-20, no classes
- Office hours: Mondays and Wednesdays 9:30-10:30
- E-mails concerning the course: You absolutely *must* have **AM3911G-2009** on the subject line. Otherwise your mail will NOT be read. Please take it seriously, I get 100+ emails per day and if your email does not get sorted to the right folder, you lose.
- Course web sites: <http://www.apmaths.uwo.ca/~mkarttu/modeling-2009.shtml>
- If there are scheduling conflicts, you should let me know as soon as possible.
- Prerequisites: Calculus 2303/2503 and AM2813B or Stats 2864 A/B. Knowledge of a programming language (e.g., C, C++, Fortran, Pascal, Python), access to a computer.
- Report of your final project must be submitted by the end of the term (Apr. 9) (see the instructions below). The deadline is **ABSOLUTELY STRICT**.

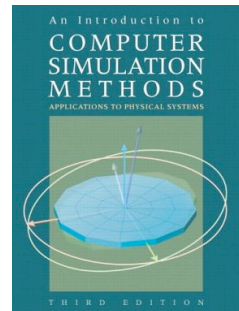


Fig. 1.1. Textbook: "An Introduction to Computer Simulation Methods" by Gould, Tobochnik, and Christian.

1.2 Brief summary of the course

We will cover an overview of some of the most commonly used and applied modeling and simulations techniques, including molecular dynamics and Monte Carlo. It is essential that the students have a background in C, C++, Java, Fortran or comparable. The course contains a project work which is of computational nature. Rough outline of the topics we may cover: Random numbers and random number generation, random walks, particle simulations Monte Carlo integration and simulations, Ising model, normal modes, oscillatory systems, chaos, fractals, bifurcations, percolation, neural networks

1.3 Evaluation

There are three in-class exams, weekly homework problems, and a project work. The following grading scheme will be used:

- Homework problems: 40%
- Project work: 30%

- In-class exams: 3 exams, each worth 10%. Total: 30%.
- Make-up exam: Only with instructors permit and has to be arranged at least one week before the scheduled exam. For students with extraordinary personal or family circumstances, it is the student's responsibility to contact the instructor WITHIN ONE WEEK of the missed exam. Documentation may (and practically always will) be required to prove the reason for absence. Please notice that the faculty has an illness/compassionate absence form that you MUST use in such a case & medical doctors also provide a proof if requested and you must have one if you are absent due to an illness. The above means that you have to get the forms from your faculty, not me.

1.3.1 Assignments

There will be up to 6 assignments. Each problem is of equal value (but the number of problems per assignment may vary). Every single assignment must have your name and student ID typed clearly and you must show *all* steps in your calculations. Providing just the answer without any calculations will give you exactly zero points. All the sheets in must be stapled together. Use a stapler, it is not overly difficult. Code listing(s) must be provided when a problem involves programming. The assignments are to be returned to me or the TA by the end of the corresponding Thursday's class (17:30). Late assignments: Late up to one day: points will be reduced by 10

1.3.2 Exams

There will typically be one verbal question (typically consisting of several smaller items) in all types of exams. As for the exams, here is some information:

Most of the problems (typically at least 80%) will come directly from assignments, textbook, and lectures (examples, etc). No tricks, just study (and do it well!) the assignments, textbook & lecture notes and you will do very well. Please keep in mind: anything that is in the assignments, textbooks or lectures (including possible additional notes) may be asked, even the long questions (you have solved them once or at least have had the chance to look at the sample solutions and hence you should be able to solve those problems quickly).

1.3.3 Project work

The course involves an individual project work. See separate list for possible projects and choose one as soon as possible.

1.3.4 Grading of the project & how to submit it

First of all, you should use a proper programming language such as C, C++, Java, Fortran, Python, Pascal, or comparable. If you, for some reason, choose to do the final project using Matlab, there will be a reduction of 10% (of the total). Furthermore, you should write your own routines for simple things such as Gaussian quadratures, LU-decomposition, etc. Using library calls for those will also reduce the mark. If in doubt when it is all right to use library calls, please ask.

At the end of the final project you must:

1. Send me your source code and *detailed* instructions for compiling it. If the instructions are not sufficient, that affects your mark (see below).
2. Prepare a report of roughly 8-12 pages. See Appendix A for detailed instructions.

Grading is based on your code, how well it functions and how general problems it can handle, and the clarity and presentation of your documentation, If I cannot get your code to compile or run, it will lead to an automatic deduction of points by 10% (of the maximum)

And finally: The deadline for submission of your final project is absolutely STRICT.

1.4 Teaching assistant

- TBA
- Office hours: TBA

1.5 How things work:

- You SHOULD take notes during the lectures. I will NOT provide additional transcripts. If you miss a class, please talk first to your fellow students. It is your responsibility to take notes and make sure you have all the material. Some exam problems may come from the examples I present in the classes – so make sure you go through them before when you are reading for the exams.
- The one and only way is to practice, practice and practice until you get a good feel of the methods covered here. Do your homework and ask the TA or me for help if/when needed. Understanding is gained through practice.

1.6 IMPORTANT!

And at the end, please keep in mind that

you, the student, are responsible of your own success!



Fig. 1.2. Take notes.