Math 6601 - Numerical Methods in Scientific Computing 1
Fall 2013-14

Instructor:	Ed Overman	Office Hours:	MWF2:50–4:00 and by appointment
Office:	440 Math Tower		(I'm around a lot of the time)
Telephone:	292-1046	e-mail:	overman@math.ohio-state.edu
		web page:	www.math.ohio-state.edu/~overman

Course Material –

You will download the "almost-book" *The Art of Scientific Computing* by Greg Baker and Ed Overman.

Course Content –

This new discipline of scientific computing lies somewhere between the disciplines of theoretical science and of experimental science. The goal in all three disciplines is to advance our knowledge and understanding of physical systems — it is just the methods that we use which are different. Experimenters use apparati to determine causal relations in a physical system. Theoreticians obtain mathematical models which simulate this physical system and solve these models analytically. Computational scientists also use mathematical models of this physical system. However, since these models are generally quite intractable, they carry out computer simulations of these mathematical models in order to understand the underlying physical system.

The basic principle in all three of these disciplines is that the results must be *reliable*. In both theoretical and experimental science there are a number of commonly accepted principles and general techniques which have evolved over many, many decades to enable researchers to carry out reliable studies. Computational science is a young discipline and a commonly accepted methodology for guaranteeing reliable computational results has not yet coalesced. In this book we present our view of a sound methodology, that is, the principles and procedures which are necessary to achieve this aim. In order to better present this methodology, we separate scientific computing into four steps:

- the formulation of one or more mathematical models to represent the physical system.
- the design of numerical methods to enable the mathematical model or models to be simulated in a computer.
- the production of bug-free codes to execute these numerical algorithms.
- the generation of reliable numerical results in order to study the original physical system.

The topics which will be covered during the year are reasonably standard: the evaluation of functions, the solution of linear systems, zeroes and minima of nonlinear systems, bifurcation studies, the approximation of functions, numerical differentiation and integration, the solution of ordinary differential equations, eigenvalue problems, and random number generators. In addition, we will spend some time on the numerical solution of partial differential equations.

Prerequisites –

It is assumed that students have some familarity with linear algebra and the analytical solution of ordinary differential equations.

The computer language in this course is **not** a computer language — it is MATLAB. It

is much easier to program in MATLAB than in any “real” computer language. It is very easy to try out “ideas”, and to modify them, in MATLAB to see how well they work — and, indeed, if they work at all. The amount of time needed to try something out in MATLAB is a small fraction of the time needed in a “real” computer language.
If needed, there will be a separate lab class of about two hours to discuss the necessary fundamentals of MATLAB. I also have a MATLAB tutorial on my web page which you can use to either learn MATLAB or refresh your knowledge of it.

Attendance –

Attendance is not required in class.

Homework –

Computational and analytical homework will be assigned every week. You are expected to do the homework yourself — although you are free to discuss the problems with others. You can code all of the computational homework in MATLAB — although you can use Fortran 77, Fortran 95, C, C++, or Java if you prefer (but you shouldn’t). Some of these problems will require the programs you will be using in the lab.

Homework

All parts of a problem must be *together* — not scattered throughout the homework. *If a problem is computational, the listing must be included and must be at the end of the problem. The homework is to be turned in in* ***one*** *piece. I don’t care how you do it, but I do not want it falling apart when I try to grade it.*

When you use a computer program in solving a probrem, include the listing at the **end of the problem**. There are a number of reasons for this:

(1) When you turn in homework you should turn in everything you do to solve the problem;

(2) if your results differ from another student’s, you can compare listings;

(3) if your results are incorrect, I can get some idea of what you have done (if I am interested); and,

(4) good programming style is very important. I want to see your code so we can have a heart-to-heart talk if you use ☹ BAD ☹ programming style.

In addition, there will be one computational project assigned each semester. **The project must be written up in the style of a laboratory report:**

(1) Describe the analytical problem you are studying in detail.

(2) Discuss how you are solving the problem numerically. This includes:

- what computer you are using (and if the program is in single or double precision);

- what methods of discretization you are using;
- what error criteria you are using;
- if you *do not* know the exact solution of the problem you are studying what methods you will use to verify that your code is working correctly, and then what the results of your verification are.
- what software packages, if any, you are using.

(3) Show and explain your results.

The homework is due when you come to class on the date due. **Late homework will not be accepted.** If you are going to be out of town, let me know **beforehand** and we can work something out.

Plagiarism –

A problem which has, unfortunately, occurred occasionally is that of plagiarism. You are expected to write your own computer programs, although you are certainly free to discuss them with other students, and you are free to try to get other students to help you debug your programs (free food and/or liquid refreshment is a good inducement). However, the programs must be your own. Program plagiarism will be suspected if two students' programs are identical up to simple transformations, or if a student cannot explain how his or her program works.

Note: The reasons I want you to include computer listings with your homework is **not** because I want to search through everyone's programs and check for plagiarism.

Exams –

None.

Grade –

The course grade is made up of the regular homework (70%) and the computational project (30%).

Disability Statement

Students with disabilities that have been certified by the Office for Disability Services will be appropriately accommodated, and should inform the instructor as soon as possible of their needs. The Office for Disability Services is located in 150 Pomerene Hall, 1760 Neil Avenue; telephone 292-3307 and VRS (614) 429-1334; webpage http://www.ods.ohio-state.edu.

Academic Misconduct Statement

It is the responsibility of the Committee on Academic Misconduct to investigate or establish procedures for the investigation of all reported cases of student academic misconduct. The term "academic misconduct" includes all forms of student academic misconduct wherever committed; illustrated by, but not limited to, cases of plagiarism and dishonest practices in connection with examinations. Instructors shall report all instances of alleged academic misconduct to the committee. For additional information, see the Code of Student Conduct:
http://studentaffairs.osu.edu/resource_csc.asp