GG 250 Scientific Programming [3 cr]

Overview:
The computer is an indispensable tool used in all aspects of the sciences. It may be the single most important instrument that students should learn to operate. Mastery of basic computer software such as “office productivity tools” is essential for any profession. Students of the physical sciences, however, need further skills allowing them to leverage the computer in their work. Skills in Computer Programming can be taught using a variety of computer languages. Practicing scientists also need to quickly visualize results from programming and from data analysis. We introduce this material in a one-semester required course using the Matlab programming environment. Matlab is a compact procedural language with built-in vector/matrix notations. It comes with a vast library of numerical and graphical functions and is therefore ideal for solving a broad range of scientific problems over a short timeframe. It is also available on all major platforms (Windows, Macintosh, Linux/Unix), a low-cost student edition is available, and free near-clones such as Octave exist.

Prerequisites:
MATH 241 (or concurrent) or departmental approval.

Course Content:
The course will be divided into four main sections:

I. Introduction [1 week]: Purpose and Goals, Organization and Course Rules, Highlight of Matlab
II. Using Matlab [~4 weeks]: Evaluate expressions, Read data, Plotting, Layouts and Saving Figures, Arrays and Indexing
III. Programming Matlab [~7 weeks]: Design, flowcharts, Pseudo Code, and Documentation, Data types, Relational operators, Control flow, Loops, Procedural Programming, Debugging
IV. Solving Composite Problems [~4 weeks]: Modeling and Simulations

Following the introductory section I, we will spend a few weeks using Matlab as a tool to calculate, visualize, and analyze data (section II), and then the bulk of the class will be devoted to learning programming (section III). The final three weeks will see integrated uses of Matlab, building on the skills already learned to do some modeling and simulations (section IV).

Text:
Amos Gilat, Matlab – An Introduction with Applications, John Wiley. The main text will be supplemented by the instructor's notes on Laulima.

GG Student Learning Objectives (SLOs):
GG department has defined 5 learning objectives for the undergraduate degree program related to Relevance of Geology and Geophysics, Technical knowledge, Scientific method, Oral and written skills, and Evaluating Phenomena. This course incorporates content relevant to 3 of those:

- SLO2: Students can apply technical knowledge of relevant computer applications, laboratory methods, field methods, and the supporting disciplines (math, physics, chemistry, biology) to solve real-world problems in geology and geophysics.
• SLO3: Students use the scientific method to define, critically analyze, and solve a problem in earth science.
• SLO4: Students can reconstruct, clearly and ethically, geological knowledge in both oral presentations and written reports.

While SL02 enters particularly strongly via the development of programming and applying their math and physics skills, SLO3 enters in how we break down a problem into multiple steps and test how each part works. SL04 mostly enters via the written lab reports.

Course Goals:
Enable students to solve practical problems using the Matlab computer language. In particular students will
• Learn the concepts of programming, i.e., variables, control flow, input/output, functions, etc.
• Gain experience in developing solutions to multi-step problems.
• Build solutions using their math and physics background.
• Practice translating such solutions into working code.
• Experience how to test new code and find bugs.
• Appreciate the importance of documentation and clarity of code comments.

Students will reach the SLOs by doing weekly lab sessions and will be tested for factual knowledge at mid-semester and at the final exam. Tests are used to determine areas of student weaknesses, which will then be addressed via lectures and lab exercises.

Assessment and Grading:
The labs are the most important aspect of the course since programming is very much hands on and experience-based. Lab reports must be submitted via Laulima by the due date in order for you to get full credit, unless you have a valid excuse and have made arrangements with us to hand it in late. Late reports will receive 50% credit only. If you anticipate a conflict for exams, you must re-schedule the exam prior to the scheduled date. Final grade will be a weighted average of grades for labs (65%), mid term (15%), and final exam (20%).

Class Format:
Each week has two 50-minute lectures and one 3-hour lab. You are encouraged to actively ask questions in class, particularly if you do not understand the material being discussed. Most lectures involve mini examples on the computer and later on mini projects, interspersed with standard lecturing. Most of the important material will be introduced and discussed in the lectures, and each weekly lab starts with an overview of the current lab exercise and relevant background material in support of it. Students then work on the lab in small groups until time is up, asking questions of the instructor along the way. The instructor will clarify common misconceptions and interrupt the groups with impromptu presentations to lead them in the most productive direction. The lab submission is due one week later.

Lecture Notes:
These will be laboratory-supporting materials for students to download (usually the week of the lab) from Laulima. Data sets used to demonstrate particular projects will be made available as well. Typically, all typing the lecturer does during the lecture will be posted afterwards on Laulima.