Spring 2014

Math 190 – Introduction to Programming (2)

An introduction to numerical algorithms and structured programming, using Fortran, MATLAB, or other appropriate language.

Pre: One semester of calculus (or concurrent), or consent

1. Purpose of this course. The purpose of this course is to teach students the rudiments of mathematical programming in the computer languages most often used in numerical analysis. The students should learn:

1. the syntax of these languages,
2. how to translate mathematical ideas into algorithms, and
3. how to translate mathematical algorithms into programs.

These skills would be useful for students in engineering, the sciences, and mathematics.

The elementary, practical nature of the course (reflected in the course number) and one–unit format are designed to encourage students to use the course as a supplement to their current curriculum, and this seems to be what is happening. Any first–semester calculus course will provide the students with sufficient background to understand the problems used in the course. The calculus course could be taken concurrently.

2. Organization of course content. The following topics are covered (at a rate of one general topic per week).

1. An introduction to the mechanics of the computer lab and QBasic.
2. Types of variables and operations. Input and output of variables.
3. Arrays, loops and iteration.
4. Logical expressions and "If...then..." statements.
5. Branching. Formatting output. "While" loops.
6. Working with files.
7. Functions.
8. Subprograms.
9. Introduction to Fortran. Compiled versus interpreted programs.
10. The mechanics of using telnet and uhunix. Translating a Basic Program to Fortran.

11–16 Items 2–7 in Fortran.

The following represent typical programming assignments, which are varied from semester to semester.

1. Write a program which produces interest tables.
2. Write a program to compute binomial coefficients.
3. Write a program to sort numerical data.
4. Write a program to solve any quadratic equation.
5. Write a program to compute trigonometric functions using series.
6. Write a program to compute the mean and standard deviation of a set of data.
7. Write a program to compute the terms of a recursively defined sequence (e.g., Fibonacci numbers).
8. Write a rootfinding program (bisection and/or Newton's method).
9. Write a program to integrate using Simpson's rule.
10. Write a program to multiply matrices.
11. Write a program implementing Euler's method to solve a differential equation.

The course uses the Mathematics Department Computer Lab for both classes and individual work.