Math 104A, Summer 2010

Introduction to Numerical Analysis
Monday, Thursday, Wednesday, & Tuesday, 12:30-1:35pm
GIRV 2108

Instructor: Francisco de la Hoz.

Office: South Hall, 6709.
Phone: (805) 893-5917.
E-mail: `delahoz@math.ucsb.edu`
URL: `http://www.math.ucsb.edu/~delahoz/Math104A`
Office Hours: Tuesday and Thursday, 11:00am-12:30pm

Teaching Assistant: There is no T.A. for this course.

Textbook: *Numerical Analysis*, by Richard L. Burden and J. Douglas Faires, 8th edition.

Course description: This is the first part of a three-quarters introductory course on Numerical Analysis. This quarter we will study numerical methods for the solution of nonlinear algebraic equations, interpolation, extrapolation, numerical differentiation and integration, and numerical solution of ordinary differential equations. Although the emphasis will be in applications, the course will have a strong theoretical component. By the end of the course, the following will be expected:

1. Convergence of a numerical algorithm: What it means, and how to determine whether the algorithm converges or not.

2. A solid knowledge of Approximation Theory:

   (a) Polynomial Interpolation.

   (b) Numerical Integration: Design and implementation of quadrature rules.

   (c) Numerical Differentiation: Design and implementation of differentiation formulas of arbitrary order of accuracy.

   (d) Estimation of the error in a given approximation.

3. Solution of Differential Equations:
   (a) Design and implementation of solvers for Ordinary Differential Equations (ODEs).
   (b) Being able to obtain the stability region of a multistep method for ODEs.
   (c) Estimation of the error and the stability properties of a given method.
4. Solid programming skills.

I will not hand out *practice exams* or anything like that. As part of the course, you should develop the necessary skills to study this subject. In order to help you study, I will periodically post *Self-Assessment Questions* questions. You do not need to hand them in. Those questions will help you know whether you are studying effectively or not, and whether you need to review some parts of the material covered in class.

Prerequisites: Math 5 A, B, and C or equivalent. Knowledge of a computer language suitable for numerical computing: FORTRAN, C, C++, or Matlab.

Assignments and grading: Homework will be assigned on Monday, and will be collected at the beginning of the class on the following Monday. **Late homework will not be accepted. If you have a predictable absence, you will need to hand it in earlier**. The homework will generally consist of some theoretical questions, and some computational assignments. You will be required to write a program to solve certain problems. The program must be given to me as part of the assignment, together with the output of the program, in the format indicated in the assignment, and an interpretation of the results whenever necessary. You can write the programs either in FORTRAN, C, C++, or Matlab. The book comes with a CD that contains the code for the problems. You may use this code as a guide, but **you must write your own original code for the assignments. No credit will be given for using the code in the CD**.

There will be a midterm and a final exam. Your final grade for the course will be given by the formula

$$\text{Final Grade} = 20\%\ \text{Homework} + 30\%\ \text{Midterm} + 50\%\ \text{Final}.$$

You must obtain 50% or more in the Final exam.

Syllabus: During this course we will try to follow the following schedule. However, much like everything said earlier, this is subject to change.

Week 1: Mathematical and Computational Preliminaries.

1. Review of Calculus.
2. Round-off Errors and Computer Arithmetic.
3. Algorithms and Convergence.
4. Numerical Software.
5. Introduction to programming with Matlab.

Week 2: Solutions of Equations in One Variable.

1. The Bisection Method.
2. Fixed-Point iteration.
3. Newton's method.
4. Error Analysis for Iterative Methods.
5. Accelerating Convergence*.
6. Zeros of Polynomials and Müller's Method.

Week 3: Interpolation and Polynomial Approximation.

1. Interpolation and the Lagrange Polynomial.
2. Divided Differences.
3. Hermite Interpolation.
4. Cubic Spline Interpolation*.
5. Parametric Curves*.

Weeks 4 & 5: Numerical Differentiation and Integration.

1. Numerical Differentiation.
2. Richardson's Extrapolation.
3. Elements of Numerical Integration.

4. Composite Numerical Integration.
5. Romberg Integration.
6. Adaptive Quadrature Methods.
7. Gaussian Quadrature*.
8. Multiple Integrals*.
9. Improper Integrals*.

Weeks 5 & 6: Initial-Value Problems for Ordinary Differential Equations.

1. Elementary Theory of Initial-Value Problems.
2. Euler's Method.
3. Higher-Order Taylor Methods.
4. Runge-Kutta Methods.
5. Error Control and the Runge-Kutta-Fehlberg Methods*.
6. Multistep Methods.
7. Variable Step-Size Multistep Methods*.
8. Extrapolation Methods*.
9. High-Order Equations and Systems of Differential Equations.
10. Stability*.
11. Stiff Differential Equations*.

*If time permits.