Time and Place: MWF 9:00-9:50, SMLC 356
Instructor: Prof. Monika Nitsche
Contact: nitsche@math.unm.edu
Office Hours: MW 1:00-2:00 (SMLC 334)
F 11-11:50, Calculus Table (DSH, by elevator on 3rd floor)
or by appointment - please email

Prerequisite: (Math 316 or Math 314 or Math 321) and
(CS 151 or CS 152 or Phys 290 or ECE 131)

Text: *Numerical Analysis*, by Timothy Sauer, Pearson/Addison Wesley.

Description: This is an introductory numerical analysis course. We study numerical methods to solve linear and nonlinear equations, to interpolate and approximate data, and methods for numerical integration and differentiation. We will implement all algorithms in MATLAB, and begin the course with a MATLAB tutorial.

1. *MATLAB Tutorial*
   Vector operations, matrices; graphing functions, tables; if, while, for; scripts and functions
2. *Solving nonlinear equations* $f(x) = 0$
   Bisection method, Fixed-point iteration, Newton's method, Secant method, Newton's method for nonlinear systems
3. *Finite precision effects, Conditioning and Stability*
   integer and floating point representation, relative and absolute error, roundoff vs discretization error, conditioning of linear systems
4. *Linear Systems*
   Direct methods: tridiagonal and triangular systems, Gauss Elimination, LU, PLU, QR factorizations, operation counts
   Norms and Conditioning
   Iterative methods: Jacobi, Gauss-Seidel, SOR
5. *Least Squares*
   Overdetermined linear systems; Least squares, Normal Equations; QR factorization
6. *Interpolation*
   Polynomial interpolation (Vandermonde, Lagrange, Newton)
   Approximating functions (interpolation error, Runge phenomena, Chebyshev pts)
   Splines
   Trigonometric interpolation
7. *Numerical Differentiation and Integration*
   Numerical differentiation
   Trapezoid rule (error, Richardson extrapolation and Simpson's rule, corrected trapezoid)
   Newton-Cotes rules
   Gaussian quadrature
8. *Initial Value Problem* $y'(t) = f(y(t), t)$ *(Chapter 6)*
   Forward Euler, convergence and stability
   Backward Euler
   Runge Kutta methods

Grading: Your grade for this course is based on weekly homework and computing projects, in-class work/attendance, and a final project, in the following proportion:

Homework/computing projects	50 %
Two mid-term exams	30 %
Final Exam	20 %
	=====
Total	100 %

Homework: The homework will include computing problems. For these, you are to write your own code, in MATLAB, without copying from other sources, unless indicated. The writeup of each of your homework problems must follow the following format:

- neatly and legibly written, clear and concise;
- all problems should be answered in order, stapled together, with your name, homework number and date written on the front;
- for each problem:
    - clearly answer the question, using English sentences were appropriate;
    - all figures and tables should be presented *and discussed* in order;
    - all figures and tables should be clearly legible, and well labelled using sufficiently large fonts;
    - conclude each problem with a sentence or paragraph summarizing your results, where appropriate;
    - at the end of each problem, after your summary, include printouts of all the MATLAB code used (all functions and all calling scripts);
    - your codes should be commented.

There will be no credit for homework that is not legibly and neatly presented.

In class work and exams: In class and in exams you may be asked to write MATLAB code independently, from scratch. It is thus important that you get comfortable doing so in the homework.

You are encouraged to work with each other on the homework projects, but the computer codes and your writeups have to be your own.