Math 352, Winter 2015, Project
Numerical Analysis II, Prof. Bownik

Instructions: 20% of your grade will come from a programming or research project due on the last day of classes. You can work individually or in teams of two. If working with another student, you should turn in a single project for both of you. You should decide on a topic within a week and then discuss with me either in person or by e-mail. I recommend that you turn in a rough draft before the last week of classes in order to receive a feedback.

Computer programming: Topics can be written in any language, such as Java, C, C++, or Mathematica. You must submit your source code, which should be thoroughly commented, and also a text file with a description of the program.

Research project: Projects should be 4–5 typed pages and should include a discussion of the theory, examples of computations, and a complete bibliography. You might consider writing your research project as a Mathematica notebook, since it has built-in formula formatting and you can use it for your calculations.

Topics: You can choose any of "Student Research Projects" listed in the textbook, an assortment of computer problems from the textbook, or another project of your own design in consultation with me. Most importantly, choose a topic that you find interesting. For example, a topic that ties Numerical Analysis to your other studies or interests. Here are a few suggested projects.

(1) **Gaussian elimination** Implement on a computer Gaussian elimination with scaled partial pivoting and investigate its numerical stability on some examples such as Hilbert matrix. Alternatively, write a research projection on algorithms for solving dense linear systems such as Gauss-Huard algorithm, see problem 7.2C:24.
(2) **Eigenvalues and eigenvectors.** Investigate some methods for computing eigenvalues and eigenvectors, see §8.3.
(3) **Singular Value Decomposition.** Investigate singular value decomposition, see §8.3.
(4) **History of splines.** Investigate history of splines, see problem 9.1:28.
(5) **Splines.** Implement an algorithm for interpolation using B splines, or for Bézier curves, see §9.3.
(6) **Ordinary Differential Equations.** Implement an adaptive Runge-Kutta algorithm described in §10.3.
(7) **Random number generators.** Investigate algorithms for generating random numbers, see problem 13.1C:22.
(8) **Minimization of functions.** Investigate algorithms for minimizing functions of several variables, see §16.2.
(9) ℓ^1 **minimization.** Investigate methods for solving inconsistent linear systems through ℓ^1 minimization, see §17.3.