Purpose of the project. The project in this course is intended to represent, to a large extent, your own individual original work, including consideration of related previous work, significant numerical computation devised by you, and the associated insights that you gained from the project.

Form of the project. A project should have the form of a scientific paper, i.e., typed pages, beginning with the project title, your name (as author), an abstract, the project body, and a bibliography. The project body is likely to consist of at least 5-7 pages that describe the topic, summarize your investigations, and state what you learned, based on a combination of mathematical analysis and computational experiments. (However, there are no rigid rules about length.) A supplement containing your code (or code from others, properly attributed) that was used for the computations should be provided.

Advance approval. Each student must arrange an individual meeting with me to discuss his/her project before starting serious work on it. As part of the discussion (possibly afterwards), I will ask for a “prospectus” in which you describe your project in general terms. Submitted projects whose topics I have not explicitly approved in advance will receive no points.

Your project for Numerical Computing should not be a recycled version of a project previously done for another class.

Grading. As discussed in class, your project will count as at least 30%, but no more than 40%, of your grade in the course (depending on which result is more favorable to you). Your project grade will be based on (1) knowledge and understanding of numerical computing, (2) creativity in applying this knowledge and understanding to a new problem, (3) clarity and correctness of explanations and examples, (4) insightfulness of the computational testing, and (5) originality. The exposition of the paper as well as the quality of its content will count in the grade.

Citing the work of others. I expect everyone to seek references and information on the Web and in the library beyond the initial pointers mentioned below. It is essential for your project to cite any and all reference material used in the project, including material from the Web. A failure to cite work by someone else that you use in your project will be considered plagiarism.
The following list provides several suggestions. If you wish to do your project on some other topic, that is fine as long as it meets the criteria described above and I have approved the project in advance following our individual meeting.

1. **Condition estimators.** Various procedures have been suggested for obtaining an estimate of the condition of a matrix without forming an explicit inverse. For background, see Chapter 15 of Nick Higham’s book *Accuracy and Stability of Numerical Algorithms* (second edition, 2002), published by the Society for Industrial and Applied Mathematics (SIAM; [www.siam.org](http://www.siam.org)). This project could (for example) describe the more popular condition estimators, including their strong and weak points, trying them out on your own examples using Matlab, and commenting on the overall usefulness of such techniques.

2. **Linear algebra in Web search, data mining, and clustering.** This is a very large topic, suitable for more than one project.

   The 2011 Research Experiences for Undergraduates (REU) project supervised by Carl Meyer, Professor of Mathematics at North Carolina State University, focused on mathematical and computational techniques used in data mining, search, and information retrieval systems. The final report ([http://meyer.math.ncsu.edu/Meyer/REU/REU2011/REU2011.html](http://meyer.math.ncsu.edu/Meyer/REU/REU2011/REU2011.html)) describes several approaches that were taken, including discussion of the linear algebra issues.

   Amy Langville, a former PhD student of Carl Meyer, is now on the faculty at the College of Charleston. Her web site contains many pointers and references about her research on clustering and ranking; see [http://langvillea.people.cofc.edu/](http://langvillea.people.cofc.edu/).

3. **Smoothed analysis applied to Gaussian elimination.** This project is likely to appeal to people who are interested in theoretical computer science. Dan Spielman (Yale) and Shang-Hua Teng (University of Southern California) developed a remarkable approach, *smoothed analysis*, that tries to explain the observed behavior of algorithms in practice, particularly algorithms with poor worst-case complexity that tend to work well in practice. See the discussion of Gaussian elimination and condition number in [www.cs.yale.edu/homes/spielman/Research/nopivotdas.pdf](http://www.cs.yale.edu/homes/spielman/Research/nopivotdas.pdf) and [cacm.acm.org/magazines/2009/10/42479-smoothed-analysis/fulltext](http://cacm.acm.org/magazines/2009/10/42479-smoothed-analysis/fulltext) and also more recent papers on Spielman’s website: [www.cs.yale.edu/homes/spielman/SmoothedAnalysis](http://www.cs.yale.edu/homes/spielman/SmoothedAnalysis).

4. **Numerical computing in econometrics.** The Mathworks™ Econometrics Toolbox provides several functions for modeling economic data. Depending on your interests, several possible projects could be defined in which you would choose a particular problem and try different numerical methods for different scenarios/data, explaining strategies that work well and situations when difficulties can arise.

5. **Principal component analysis; factor analysis.** Principal component analysis (PCA) is a technique used in statistics to identify the “most important” elements in a data set and to reduce dimensionality to the features that matter the most. Numerous tutorials about PCA can easily be found on the Web, and the better ones discuss the SVD; see, for example, [www.snl.salk.edu/~shlens/PCA.pdf](http://www.snl.salk.edu/~shlens/PCA.pdf).

   Factor analysis is related to PCA but is not exactly the same. As with PCA, many tutorials about factor analysis can be found on the Web, for example [www.chem.duke.edu/~clochmul/tutor1/factucmp.html](http://www.chem.duke.edu/~clochmul/tutor1/factucmp.html).
A variety of projects related to PCA or factor analysis would be appropriate as long as they include a discussion of numerical methods, your own interesting computational results, and your analysis of the results.

6. The global warming “hockey stick” controversy. Principal component analysis plays a key role in a scientific and political controversy that started in 1999 about global warming. For (too much) information, just Google ‘hockey stick controversy principal’ and start with the Wikipedia article.

7. Deblurring images. The book Deblurring Images: Matrices, Spectra, and Filtering (2006), by Per Christian Hansen, James Nagy, and Dianne O’Leary, published by SIAM and available in the Courant library, shows how numerical computing techniques are applied in deblurring images. Related material, including images for testing and experiments, is also available at www2.imm.dtu.dk/~pch/HNO.

8. Strong words by Velvel Kahan on roundoff error. As discussed in class, Professor William (“Velvel”) Kahan, University of California, Berkeley, was the principal mover in creation of the IEEE floating-point arithmetic standard. His pungent commentary on some of the difficulties with performing assessments of roundoff errors can be found at www.eecs.berkeley.edu/~wkahan/Mindless.pdf. Sections 2 and 8 of this document (especially the examples from Excel) suggest several projects.


10. Applications of the SVD. An increasing number of researchers in medicine, especially those who rely on images, use the singular value decomposition. See, for example www-sop.inria.fr/asclepios/Publications/Reyes/mreyes-3DFull05.pdf (respiratory motion correction in lung tomography), and public.lanl.gov/meewall/kluwer2002.html (gene expression analysis).

11. Safeguarded line searches in optimization. In class, we briefly discussed the concept of safeguarded methods for one-dimensional zero-finding. An analogous problem in optimization is performing a “line search”: given a nonlinear function \( f(x) \) of \( n \) variables, an \( n \)-vector \( x_0 \), and a direction \( p \), find a scalar step \( \alpha \) such that \( f(x_0 + \alpha p) \) is “sufficiently less” than \( f(x_0) \); see “Line search algorithms with guaranteed sufficient decrease”, by Jorge Moré and David Thuente, ACM Transactions on Mathematical Software 20, 286–307 (1994) (obtainable online via the ACM Digital Library). A project on this topic could (for example) explain the difficulties in designing a fully reliable line search, also provide illustrative numerical experimentation.

12. Calculation of special functions. The Digital Library of Mathematical Functions, hosted at the National Institute of Standards and Technology (NIST) (dlmf.nist.gov), explains in great detail how special functions like sine and exponential are calculated. A project on special functions could investigate the techniques used in calculating selected common special functions, discussing the tradeoffs in accuracy, computation time, and possible dependence on specific hardware features.
13. Safeguarded zero-finding methods. Our in-class discussion of these methods touched only on general principles. A project would involve looking at the nitty-gritty details of one safeguarded method, such as Matlab’s `fzero`. Essential background reading is Kahan’s paper “Personal calculator has key to solve any equation $f(x) = 0$”; this paper, called “SOLVE key on the HP 34-C”, is available at www.cs.berkeley.edu/~wkahan/Math128

14. Rank-estimation strategies and low-rank approximations. In class, we barely skimmed the surface of this important topic. A project could look at possible strategies for deciding about numerical rank, and investigate the reliability and effort associated with various strategies. For example, how does Matlab estimate rank? What are the consequences of conservative/liberal strategies, illustrated by numerical examples?

15. How to calculate the singular value decomposition and the eigendecomposition. Using Matlab, we take the commands for SVD (`svd`) and eigendecomposition (`eig`) for granted. This project would involve finding out how the SVD and/or the eigendecomposition are actually computed, including backward/forward error analysis properties, ensuring high relative accuracy even when the singular values/eigenvalues are small in norm, etc.