Projects are due in electronic form by 11:59pm, Friday December 12, 2014.

**Purpose of the project.** Students are to complete and submit a written report about an individual “mini-research” project involving a topic relevant to numerical optimization. Each project report should demonstrate your engagement with a topic not covered in class in detail, and should discuss what you learned during the project, explicitly pointing out your original contributions. All projects must involve computing (both numbers and associated explanations) based on your code and your numerical experiments.

*Do not submit a project of which major portions have been submitted in a previous class or current class.*

**Advance approval is required.** By 11:59pm on Friday, November 14, 2014, every student must have received individual approval of his/her project from me via an email message. Before approving a project topic, I will need to receive an email message from you in which you describe your project in general terms. If your project fits into one of the categories listed below, please specify that topic in your description; if not, please say that it does not fit into one of the broad topics listed below. *No credit will be given for projects on topics that I have not explicitly approved in an email message to you.*

**Form of the project.** A project should have the ingredients of a scientific paper (title, your name and affiliation, abstract, project body, and bibliography). The project body will typically consist of approximately 7–10 pages that describe the topic and summarize your investigations, including theoretical analysis and computational experiments. A supplement should be provided containing or pointing to the code (written by you or others, properly attributed) used in the project.

**The project’s role in your course grade.** There are three weights (40%, 35%, 25%), which will be associated with your three scores in (homework, midterm, project) so that the weighted sum is maximized. (This is an easy optimization problem.)

**Evaluation criteria.** Projects will be graded on (1) knowledge, understanding, and creativity related to your topic, (2) clarity and correctness of explanations and examples, (3) insightfulness of your discussion and (4) originality. *Your grade will be based on both scientific content and quality of exposition.*

**Citing the work of others.** Everyone is expected to seek references and information, from and beyond the initial pointers mentioned below. It is *essential* to include explicit citations to any and all reference material used in the project, including material from the Web. If you fail to cite work by someone else that you use in your project, this will be considered plagiarism and your score on the project will be zero.

The following list provides ideas for possible projects in various areas, but it is not even close to complete. 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.

Each numbered item on the list includes enough material for more than one project. If different students want to work on one specific topic, we will need to decide in advance, and confirm via email, how the content will be divided among you.

1. **Optimization in finance.**

   Information about possible topics within this huge area can be gleaned from Google, the SIAM *Journal on Financial Mathematics* (epubs.siam.org/journal/sjfmbj), and from the descriptions of the
numerous courses at the NYU Mathematics Department in financial mathematics. Matlab has a Financial Toolbox, which can be reached from your Courant account by using the “help” tab and then typing “financial toolbox”.

There are many websites about optimization in finance, but not all are reliable. An interesting project would identify and correct the terrible advice about numerical algorithms given in some of these websites, papers, and books.

2. Data science.
Many students in this course are interested in data science, and are much more familiar than I am with appropriate topics, so I encourage you to investigate and propose one.

3. Evaluating and comparing optimization methods.
How to decide which implementation of which method is “best” for your problem? Dolan and Moré, in “Benchmarking Optimization Software with Performance Profiles”, Mathematical Programming Series A, 91, 201–213 (2002), introduced the concept of performance profiles, which have since become ubiquitous. Moré and Wild, in “Benchmarking algorithms for derivative-free optimization”, SIAM Journal on Optimization 20, 172–191 (2009), introduced the further concept of data profiles. They have an interesting website about what they call a “shootout” (a comparison) of derivative-free optimization solvers: www.mcs.anl.gov/~more/dfo/shootout.html There could be several projects about to compare and evaluate optimization software.

Projects in this area would involve optimization methods commonly used in machine learning and describe their relationship to mainstream optimization techniques, ideally suggesting ways in which there could be increased connections. Two places to look are:

   (a) A 2012 talk by Nocedal at the International Symposium on Mathematical Programming (ISMP): users.eecs.northwestern.edu/~nocedal/PDFfiles/ismp.pdf

   (b) A tutorial by Srebro and Tewari, ICML, 2010: ttic.uchicago.edu/nati/Publications/ICML10tut.pdf

5. Matlab toolboxes related to optimization.
Matlab provides several optimization toolboxes, which are available when using Matlab from your Courant account. The Optimization Toolbox includes standard techniques for linear and quadratic programming, unconstrained optimization, etc. The Global Optimization Toolbox includes genetic, simulated annealing, and evolutionary algorithms.

In the same spirit as projects in item 3, how good are the codes in this toolbox? An interesting project could compare their performance on an interesting set of test problems with that of alternative methods for the same problems, such as the solvers available through NEOS (Network-Enabled Software for Optimization); see www.neos-server.org/neos/ and neos-guide.org/solver-software.

6. Integer programming; mixed-integer optimization
Many important real-world problems, and problems arising in computer science, involve variables that are constrained to have integer values (or to be chosen from a discrete set). This is a huge topic with many connections to continuous optimization—for example, linear programming is often the means for solving subproblems within integer programming algorithms. The numerical issues in integer programming tend to have a different flavor from those discussed in our class, but there are connections. Sven Leyffer’s lectures, available at wiki.mcs.anl.gov/leyffer/index.php/Sven_Leyffer’s_Lectures suggest several topics that could become projects.