1. Time has come to think of a simple project to work on for this class. “Simple” is crucial: you won’t have much time to work on your project, so it is important that you do not bite more than you can chew. A small project well executed is much better than an ambitious project left incomplete.

(a) Pick a project. This is of course a critical decision, and here are some examples. You can pick one of the examples as is, modify it, or come up with your own idea altogether. The main reason for listing examples here is to convey the scope and size of an acceptable project. Examples are in no particular order.

- **A textbook project:** Pick any of the projects in the textbook. For instance, if you pick project 2.1 (Stochastic Bobcats) on pages 93-94, you could implement the environmental stochasticity part (as defined in Chapter 2), and collect statistics over several runs of the simulation.

- **Extend the theory:** The class covered only some basics of modeling, and many new directions could be explored in a project. For instance, you could read a book chapter on Euler methods for integrating ordinary differential equations, and use Matlab Euler solvers to explore trade-offs for a couple of the different methods. The Matlab Help pages on ODE solvers are a good starting point, too. If you read the material under *Initial Value Problems for ODEs and DAEs*, you can work through some of the examples, and experiment with using different solvers for different types of problems.

- **Implement part of a paper:** For instance, the paper “The Spot Model for random-packing dynamics” by M. Z. Bazant, *Mechanics of Materials* 38 (2006) pp. 717-731, proposes a new model for how particles move relative to each other in a granular material. The key concept of this model is the notion of a spot, that is, a distribution of free volume between granules. As granules move and settle, the spot redistributes in a partially random way. For the project, you could implement at least the motion of one spot over a short amount of space. [Caution: make sure that at least part of the paper you pick is understandable with moderate effort.]

- **Transportation problem:** The web page [http://mat.gsia.cmu.edu/QUANT/NOTES/chap11/node8.html](http://mat.gsia.cmu.edu/QUANT/NOTES/chap11/node8.html) describes a couple of transportation problems (of the type we considered when we covered linear programming in class). Solve one of them with a linear program solver. Discuss how the solution changes as some of the problem conditions are varied.

- **Fix the gas simulator:** The Resources subpage of the class web site gives you access to the gas simulator we discussed in class. If you change parameter N in file simulate.m to, say, 100, not only does the simulation slow down unacceptably, but you will notice some particles leaving the box. This is a bug. For your project, you could find and fix the bug. [Caution: requires familiarity with the Matlab debugging tools.] A second, optional part could modify the code to simulate something else (for instance, people moving in a crowd). If you are into coding for speed, you could try to rewrite the simulator in a compiled language instead, and see if you can make it faster.

- **Make a GUI to understand something:** The same Resources web page also provides the Matlab code of a crude GUI that displays the solution to a second-order ordinary differential equation. The solution is computed through its explicit, closed-form formula. Change that to use one of the numerical ODE solvers provided by Matlab, and verify that the results are compatible. Alternatively, you may want to write a similar GUI (lookup guide in the Matlab help to learn how to use the Matlab GUI tools) for something else: a nonlinear ODE, a discrete dynamic system, a stochastic system, the solution to an LP problem.

(b) Write at least one page to describe your project. Your description should include background information (that is, a description of what you are modeling), sources (where you found the problem and possible tools to address it), a description of the actual work you plan to do (implement x, explore issue y, and so forth), and a plan for how long you expect each step to take. What are you planning to deliver at the end (a piece of code, a report, a web page, a powerpoint presentation, . . .)? Where do you expect the greatest difficulties to lie? Remember that the semester ends on April 25, so there is not much time to work on your project. Be realistic in your plan.