MS&E 315 and CME 304
NUMERICAL OPTIMIZATION
Class Project

The purpose of the project is to gain some experience in the difficulties of defining a problem, implementing an algorithm and interpreting output. Also to appreciate that good optimization algorithms are necessary.

You will be required to write a routine that minimizes a function $F(x)$ subject to the bounds $l \leq x \leq u$. You will eventually use a quasi-Newton method. However, start with steepest descent method. You will eventually use your routine to determine a control of a solar sail to move a cargo vessel from an Earth orbit to an orbit of Mars. Details of how to formulate that problem will be given later. You will need a linesearch routine. You may if you wish write a sophisticated routine but I would not do this immediately and it may not be necessary to solve the problem. A simple routine would be to take a unit step and stop if that reduces $f(x)$ and $p^T y > 0$, where y is the difference of the new gradient minus the original. If $f(x)$ is not reduced backtrack using bisection. If it is reduced but $p^T y < 0$ take a larger step (say double it). You need to work out the details but this approach is simpler than a sophisticated algorithm using approximations.

Dealing with bounds on the variables is quite straightforward for the steepest descent algorithm. Basically in any given iteration a subset of the variables are kept fixed and the steepest descent method is applied to a function of the remaining variables. In the linesearch a maximum step is first computed such that it is the largest step possible along the search direction for the step to be feasible (lie within the bounds). If the result of the search is that the next iterate is on a new bound then in the following iteration the relevant variable is kept fixed on this bound (by making the corresponding element of p zero). If other variables are on their bounds whether or not they are fixed depends on the sign of the gradient with respect to the variable. If a variable is on its upper bound then the variable remains on the upper bound, say u_i, if $g_i < 0$ (the steepest descent step would increase that variable) when computing the next iterate. Likewise, a variable is kept on its lower bound if $g_i > 0$. In the coming weeks we shall deal with the rules for the quasi-Newton method, where it will be shown they are similar to those for steepest descent.

You will not be required to submit code and may use any language you wish. In your final report you will need to justify why you are confident your code runs correctly. This is an individual project but you may consult with others on aspects of programming and the use of latex. You may also submit a request to me to do a project of your own.