Final Exam

Due 5pm on Thursday 12/15/2016, in my office box (Chapman 101).

135 points total. As stated on the syllabus, this exam is 20% of your course grade.

Rules. This take-home exam is *your own work*. You may not talk or communicate about it with any person other than me, Ed Bueler. You *are* encouraged to ask me questions about the exam during lecture time, and also during my office hours. You may use any reference book or article, print or electronic, as long as it is clearly cited. (References to the textbook Nocedal & Wright are optional.)

Recommendations. You *may* use codes posted at the class webpage,

bueler.github.io/M661F16/

Concretely, it is *recommended* that you use the following codes as needed in solving problems on this exam:

`bfgsbt.m`	`bt.m`	`newtonbt.m`
`newtonsolve.m`	`rsimpII.m`	`sdbt.m`

You may modify these codes as desired, but that is not particularly recommended. Unmodified versions of my codes do *not* need to be shown in your submitted work, but if you make modifications, or if you re-implement in another language, such codes should be included. Just as for Assignments during the semester, you *may not* use black box optimization codes from outside of this class, whether from Octave/Matlab/Scipy/etc. or elsewhere.

When you use a code, *do* show me the inputs and commands you used. *Do* show me code which you wrote. *Do* show me a few numbers which report and justify the answer. Do *not* spew unnecessary numbers at me. In general, report norms of vectors to show smallness or closeness. Report vectors themselves only if they are answers.

F1. Define the smooth function

$$f(x) = \frac{1}{5}\arctan(x_1) + x_1^2 + 3x_2 + (x_2 - x_3)^2 + \frac{1}{4}x_3^4$$

for $x \in \mathbb{R}^3$. Consider the unconstrained optimization problem $\min_{x\in\mathbb{R}^3} f(x)$.

(a) (*5 pts*) Compute the gradient $\nabla f(x)$ and the Hessian $\nabla^2 f(x)$.

(b) (*15 pts*) Show that the Hessian is everywhere symmetric and positive semi-definite. (*Hint.* The Hessian is not constant. You would be wise to think about, and calculate, its eigenvalues.) It follows that $f(x)$ is convex.[1] Next, find all points where the Hessian is *not* positive definite. Show these are *not* locations of local minima. (Identify a theorem in the book that shows this.) Conclude that there is a unique global minimum x^* where both $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is SPD. (Identify theorem(s) in the book that allow you to conclude this.)

(c) (*15 pts*) Write a code that uses (i) steepest descent and (ii) Newton method to find x^*. Both approaches (i) and (ii) should give the same result to 5 decimal digits. Also report the number of steps taken for (i) and (ii).

[1] There is nothing to prove here. Nonetheless the following is a consequence of Taylor's Theorem: A smooth function f is convex on $\mathbb{R}^n$ if and only if its Hessian matrix is positive semi-definite at all points.

F2. For $x \in \mathbb{R}^2$, define the smooth function

$$f(x) = x_1^4 + 2x_2^4 - (x_1 + x_2)^2 - x_1.$$

(a) (*10 pts*) Suppose $x_0 \in \mathbb{R}^2$. What method would you choose, of those covered during the semester, to find a local minima x^* of f? Justify your choice of method/algorithm; write four to eight sentences.

(b) (*20 pts*) In fact this function $f(x)$ has exactly two local minima, and they are in the square $(x_1, x_2) \in [-2, 2] \times [-2, 2]$. Given this knowledge, set up a grid of 25 to 100 (total) points x_0 in this square. Write a code which uses your chosen method to converge to a local minimum x^* from each initial x_0 in the grid. By examining the results, identify the locations of the two local minima, accurate to 5 digits, and their corresponding f values. Thus determine which one is the global minimum. Generate a figure which shows both the grid of initial points and the two local minima.

(c) (*5 pts*) Confirm your solution in part **(b)** by generating a contour plot of $f(x)$ on $[-2, 2] \times [-2, 2]$. Take some care in choosing which contour lines to show.

F3. For $x \in \mathbb{R}^3$, define the smooth function $f(x) = x_1^2 + x_2^4 + 3x_3^2$. Consider the equality-constrained problem

$$\min_{x \in \mathbb{R}^3} f(x) \qquad \text{subject to} \quad x_1 - x_2 - x_3 = 4.$$

(a) (*5 pts*) Explain geometrically, though informally, why this problem has exactly one solution. State the KKT conditions (12.34) which apply to this problem.

(b) (*5 pts*) The KKT conditions are a system of nonlinear equations in the four variables x_1, x_2, x_3, λ. Write this system of equations in the form used in Chapter 11 of the textbook, namely in the form $r(z) = 0$ where $z = \begin{bmatrix} x_1 & x_2 & x_3 & \lambda \end{bmatrix}^\top$. Compute $r(z)$ and the Jacobian $J(z)$. Note that you can set things up so that the Jacobian is a symmetric matrix; do so.

(c) (*15 pts*) Write a code that solves $r(z) = 0$ by using Newton's method with a line search. Run it and report the solution x^* and λ^* accurate to 5 digits, and the number of steps.

(d) (*5 pts*) How is line search done by the algorithm in part **(c)**? Explain in a couple of sentences.

(e) (*5 pts*) Confirm numerically that the "gradient balance" condition (12.35) holds, namely $\nabla f(x^*) = \sum_{i \in \mathcal{A}(x^*)} \lambda_i^* \nabla c_i(x^*)$. (Do this by computing a norm that it is small.)

F4. Consider—read carefully!—the linear program

$$\max_{x \in \mathbb{R}^3} 4x_1 + 2x_2 + x_3 \qquad \text{subject to} \quad \begin{aligned} x_1 \qquad\qquad &\le 2 \\ 3x_1 + x_2 \qquad &\le 8 \\ 5x_1 + 3x_2 + x_3 &\le 20 \\ x_1 \ge 0, x_2 \ge 0, x_3 &\ge 0 \end{aligned}$$

(a) (*10 pts*) Carefully sketch the feasible region in $\mathbb{R}^3$. Identify the coordinates of each vertex, and label the vertices with these coordinates. Make the sketch both nice and big.

(b) (*5 pts*) By adding slack variables x_4, x_5, x_6, and, by also correcting other aspects, put this problem in standard form (13.1).

(c) (*5 pts*) Starting with $\mathcal{B} = \{4, 5, 6\}$, do one step by hand using the simplex method template. The initial iterate will be $x^{(0)} = [0\ 0\ 0\ x_4\ x_5\ x_6]$; do the work to compute $x^{(1)}$. You may use a blank template `bueler.github.io/M661F16/linprogtemplate.pdf` and attach it at the end.

(d) (*10 pts*) Run `rsimpII.m` with the `spew = true` option. This will give the solution and the number of steps k. Using a colored pen, circle and label the vertices for iterates $x^{(0)}, x^{(1)}, \ldots, x^{(k)}$.