Assignment #7

Due Wednesday, 9 November 2022, at the start of class

Please read sections 11.2, 11.3, 11.4, 11.5, 12.1, and 12.2 from the textbook.[1] Please also read the online slides *Steepest descent needs help* at

bueler.github.io/opt/assets/slides/sd.pdf

DO THE FOLLOWING EXERCISES from section 11.2, pages 361–364:

- Exercise 2.7
- Exercise 2.10

DO THE FOLLOWING EXERCISES from section 11.3, pages 369–371:

- Exercise 3.2
- Exercise 3.3
- Exercise 3.4
- Exercise 3.6

Problem P13. Let $c \in \mathbb{R}^n$ and suppose $Q \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix. Consider this quadratic function on $x \in \mathbb{R}^n$:

$$f(x) = \frac{1}{2}x^\top Q x - c^\top x. \tag{1}$$

a) Show that $\nabla f(x) = Qx - c$.

b) Show that f is strictly convex. (*Hint. Use facts from section 2.3.*)

c) Suppose p is a descent direction at x, so that $p^\top \nabla f(x) < 0$. Prove that the exact solution of the line search problem $\min_{\alpha>0} f(x+\alpha p)$ is

$$\alpha = \frac{-p^\top \nabla f(x)}{p^\top Q p}.$$

(*Hint. Define $g(\alpha) = f(x+\alpha p)$, expand it, and compute $g'(\alpha)$. Explain why is it important that p is a descent direction.*)

Problem P14. In the slides I show a MATLAB implementation of steepest descent using back-tracking line search, something we will cover carefully in section 11.5. If we restrict the objective function $f(x)$ to being quadratic then we can use the result in **P13 c)** to choose the step size.

[1] Griva, Nash, and Sofer, *Linear and Nonlinear Optimization*, 2nd ed., SIAM Press 2009.

a) Implement steepest descent with optimal step size for quadratic functions (1):

```
function z = sdquad(x0,Q,c,tol)
```

As before, stop when $\|\nabla f(x_k)\| < \texttt{tol}$. (*Hint. Only small modifications of code from the slides is needed. Replace evaluations of f and ∇f by formulas for the quadratic case. Replace back-tracking by the result from* **P13 c)**.)

b) Use `sdquad()` to reproduce the result of Example 12.1 on pages 404–405 of the textbook. Specifically, you should get $k = 216$ iterations using $\texttt{tol} = 10^{-8}$.

c) Now change Q to

$$Q = \begin{pmatrix} 2.3 & 0.19 & -0.89 \\ 0.19 & 1.84 & 0.32 \\ -0.89 & 0.32 & 1.86 \end{pmatrix}$$

but keep the same c, x_0, and `tol` as in part **b)**. What is x^*? How many iterations does `sdquad()` need? Why is this problem easier than part (b)? (*Hint. What does* `eig(Q)` *tell you?*)

Problem P15. (*This problem extends Exercise 3.8 in section 11.3.*)

Though section 2.5 fails to make this very clear, the actual definition of *superlinear convergence* is that

$$\lim_{k\to\infty} \frac{\|e_{k+1}\|}{\|e_k\|} = 0$$

for sequences $\{x_k\}$ which converge to x_*, and of course using $e_k = x_k - x_*$.

a) Let $\{x_k\}$ be a sequence which converges super-linearly to x_*. Show that

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x_k\|}{\|x_k - x_*\|} = 1.$$

b) Explain why a super-linearly converging iterative algorithm for approximating some unknown number x_*, which stops according to the criterion $\|x_{k+1} - x_k\| < \texttt{tol}$, for some user-supplied tolerance $\texttt{tol} > 0$, is probably going to work acceptably. Does the same apply to iterations which converge only linearly?