

## Stat 310, 2020, Homework 3. Due at the beginning of the class, Wednesday, January 29.

### Problem 1: (20 points)

Prove Zoutendijk's theorem (Theorem 3.2 in the textbook) for the case where the steplength  $\alpha_k$  is computed by backtracking.

### Problem 2: (20 points)

Extend the proof I did in class for Lemma 3.1 in order to prove that there always exists an  $\alpha_k$  that satisfies the Goldstein conditions (3.11), under the same assumptions as the ones in Lemma 3.1.

### Problem 3: (35 points, out of which 5 for correct code submission)

- I. Implement Newton's method with Hessian modification and backtracking line search (Algorithm 3.2; with the matrix  $E_k$  computed by Algorithm 3.3). *Make sure that your code supports sparse formats.* Allow the number of iterations to be a parameter.
- II. Apply the method to the following function (Fenton's function);

$$f(x) = \left\{ 12 + x_1^2 + \frac{1 + x_2^2}{x_1^2} + \frac{x_1^2 x_2^2 + 100}{(x_1 x_2)^4} \right\} \left( \frac{1}{10} \right)$$

If you code the derivatives by hand, then implement the function (in Matlab) `[f,g,H]=fentonfgh(x)`, where  $x$  is the point of function, gradient, and Hessian evaluation.

- III. Initialize the method at  $x = [3, 2]$ . Compare with the algorithm from homework 2.
- IV. Initialize the method at  $x = [3, 4]$ . Compare with the algorithm from homework 2.
- V. (15 points) Apply the method to the order  $n$  Rosenbrock function

$$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

starting from the point with all entries equal to 3 and then all entries equal to 10, for increasing  $n$ . If you code the derivatives by hand, then implement the function (in Matlab) `[f,g,H]=rosenbrocknfgH(x)`, where  $x$  is the evaluation point of function, gradient, and Hessian evaluation. The Hessian should be computed and returned in sparse format. For the entry of all 10s, plot the compute times on your computer as a function of  $n$ . Use the gradient norm less than  $1e-3 \dots 1e-6$  as a stopping criterion.

- VI. Submit your code to the Matlab submission folder as follows. *Respect the name convention exactly, please including capitalization.*

- a. In your submission folder create a folder HWK3.
- b. Your implementation of the Rosenbrock function needs to be in that folder too.
- c. Implement a function whose name is hwk3p3.m in Matlab, that should take in the initial point, the number of iterations, and the gradient precision, and return the final point when applied to the Rosenbrock-n problem. In Matlab, the calling sequence should be:  
`[x]=hwk3p3(x0,niter,eps)` (Yes, you need an OR double test for exiting the loop, one for the gradient and one for the number of iterations; to use it for part V just take the niter as large as it needs to be).

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs