University of Mannheim
Scientific Computing, B6 26, C312, 68131 Mannheim
Dr. Andreas Sommer (ansommer@mail.uni-mannheim.de)

Nonlinear Optimization (FSS 2023)

Exercise Sheet #4

Due on 19.03.2023 (before 13:00).

---

1. **Efficient step sizes are admissible** [4 points]

   Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable, and let the sequences $(x^k), (d^k), (\sigma_k)$ be generated by algorithm (1). Futhermore, assume

   $$f(x^k + \sigma_k d^k) \leq f(x^k) \qquad \forall\ k\ \geq 0\ .$$

   Prove the following statement: If $(\sigma_k)_K$ is a subsequence of efficient step sizes, then the subsequence is also admissible.

2. **Exact line search is efficient** [4 points]

   Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable, and let $x^0 \in \mathbb{R}^n$ be given. Consider the minimization rule for the step size σ_k of the general descent method (algorithm (1)), with exact line search $\sigma_k = \sigma_{k,E}$, i.e. with

   $$f(x^k + \sigma_{k,E} d^k) = \min_{\sigma \geq 0} f(x^k + \sigma d^k)$$

   for $k = 0, 1, 2, \ldots$

   Show that the following statement holds true: If the level set $N_f(x^0)$ is compact and ∇f is Lipschitz continuous on $N_f(x^0)$, then the exact line search rule is well-defined and efficient, i.e. there exists a constant $\theta > 0$, independent of x^k, d^k, and k, and such that

   $$f(x^k + \sigma_k d^k) \leq f(x^k) - \theta \left(\frac{\nabla f(x^k)^\top d^k}{\|d^k\|} \right)^2\ .$$

3. **Armijo (sufficent descent) does not guarantee admissible step sizes** [2 points]

   Consider $f(x) = \frac{x^2}{8}$, with starting point $x^0 = 1$, and search directions $d^k = -2^{-k} \nabla f(x^k)$. Determine the (unique) global minimum $\tilde{x}$ of f and show that the descent algorithm with Armijo step size rule generates a sequence of iterates $(x^k)_k$ that does not converge to $\tilde{x}$.

4. Programming assignment: Steepest descent with Powell-Wolfe rule [8+2 points]

Put all files into a single zip-archive, named by the lexicographically ordered family names of all group members separated by a hyphen, and send the zip-file to ansommer@uni-mannheim.de. Please add printouts from code and output to your submissions.

Implement the steepest descent method with Powell-Wolfe step size strategy: Write a function that implements algorithm 4 with header

```
function sigma = powellwolfe(f, gradf, x, d, gamma, eta)
```

Here, `f` and `gradf` are functions which return for given x the functional value $f(x)$ and the gradient $\nabla f(x)$, respectively. Furthermore, `x` denotes the current iterate, `d` the descent direction and `eta` and `gamma` are the constants introduced in the lecture. The return value `sigma` is the step length.

You may modify the implementation of the steepest descent algorithm (from exercise sheet 2). Use the function header

```
function X = steepestdesc_pw(f, gradf, x0, e, maxit)
```

and call the `powellwolfe` function to determine the step size in each iteration.

The functions `f` and `gradf` are defined as described above, `x0` denotes the starting point. The iteration should be terminated if $\|\nabla f(x^k)\| \leq \epsilon$ for a given tolerance `e` $= \epsilon > 0$ or a maximum number of iterations `maxit` has been reached. The return value `X` shall contain all iterates x^k as a matrix (up to 2 bonus points for collecting the iterates in a more efficient method than using growing arrays).

Test your implementation with the Rosenbrock function

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \; .$$

Use as a starting point $x^0 = (1, -0, 5)^\top$ and $x^0 = (-1.2, 1)^\top$ and as parameters $\epsilon = 10^{-3}$, $maxit = 10000$, $\gamma = 10^{-4}$ and $\eta = 0.9$.

Plot the trajectories of the iterates with the Rosenbrock function for each starting value (use, e.g. the `surfc` command for the Rosenbrock banana and `plot3` for displaying the linearly interpolated iterates). Use a different color for each trajectory.

Compare your results to the results of the steepest descent algorithm with Armijo rule.