University of Mannheim
Scientific Computing, B6 26, C306, 68131 Mannheim
Dr. Andreas Sommer (ansommer@mail.uni-mannheim.de)

Nonlinear Optimization (FSS 2023)

Exercise Sheet #5

Due on 26.03.2023 (before 13:00).

---

1. Invariance of Newton's method [4 points]

Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable, $A \in \mathbb{R}^{n\times n}$ invertible and $c \in \mathbb{R}^n$ be given. Furthermore, assume that the sequence $\{x^k\}$ are the iterates of Newton's method for the minimization of the function f with starting point $x^0 \in \mathbb{R}^n$.

Show that Newton's method for the minimization of the function $g(y) := f(Ay + c)$ with starting point $y^0 = A^{-1}(x^0 - c)$ gives the sequence of iterates $\{y^k\},\ y^k = A^{-1}(x^k - c)$.

2. Local convergence of full-step Newton method [4 points]

Consider $f : \mathbb{R} \to \mathbb{R}$ with

$$f(x) = \sqrt{x^2 + 1}\,.$$

(a) Show that second order sufficient optimality conditions are satisfied at $\tilde{x} = 0$.

(b) Compute the kth iterate x^k (with respect to the starting point x^0) generated by Newton's method (Algorithm (5)).

(c) Show that Newton's method converges only for starting point x^0 with $|x^0| < 1$.

3. Alternative motivation of Newton's method [4 points]

Consider the minimization of a twice continuously differential function $f : \mathbb{R}^n \to \mathbb{R}$.

Show that (local/full-step) Newton's method corresponds to iteratively solving quadratic approximations of the objective (Remark after Algorithm (5)).

(a) Determine a quadratic approximation $q_k(d) \approx f(x^k + d)$ of the objective function f using Taylor's theorem.

(b) Assume that $\nabla^2 f(x^k)$ is positive definite at the current iterate x^k, so that q_k has a unique minimum d_k^*. Show that choosing

$$x^{k+1} = x^k + d_k^*$$

delivers the local Newton method (Algorithm (5)).

Note: You may use that, if (SOC2) holds at a local minimum $\tilde{x}$ of f, then there exists a whole neighborhood $B_\varepsilon(\tilde{x})$ on which the Hessian $\nabla^2 f$ is positive definite.

4. Programming assignment: Local Newton Method [5 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@mail.uni-mannheim.de. Please add printouts from code and output to your submissions.
Comment your code intensely. Use a complete header that describes input and output arguments and also comment the implementation where appropriate (see the examples at the course web site). Avoid obvious inefficiencies like repeated evaluation of identical/unchanged expressions.

Implement the local Newton method (Algorithm (5)). Use a function header

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

Here, `f`, `gradf` and `hessf` are functions which return for given x the functional value $f(x)$, the gradient $\nabla f(x)$ and the Hessian $\nabla^2 f(x)$, and further `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 if a maximum number of iterations `maxit` has been reached. The return value contains all iterates x^k as a matrix.

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 starting points $x^0 = (1, -0, 5)^\top$ and $x^0 = (-1.2, 1)^\top$ and as stationarity threshold $\epsilon = 10^{-9}$ (nine!), set `maxit` $= 10000$.

Plot the trajectories of the iterates with the Rosenbrock function for each starting value. How many iterations are needed until stationarity $\nabla f(x^k) = 0$ (with equality) is reached?

Compare your results to the results of the steepest descent algorithm.

[NOTE: Do not calculate matrix inverses!]