Univ.-Prof. Dr. Radu Ioan Boţ

Vienna, November 29, 2019
To be discussed on December 6, 2019

Nonlinear Optimization

Exercise session 5

29. Let $A \in \mathbb{R}^{n\times n}$ be a symmetric and positive matrix, $b \in \mathbb{R}^n$ and the quadratic function $f : \mathbb{R}^n \to \mathbb{R}$, $f(x) = (1/2)x^T Ax - b^T x$. Further, let be $d^0, d^1, ..., d^{n-1} \in \mathbb{R}^n$ nonzero vectors with the property that

$$(d^i)^T Ad^j = 0 \text{ for all } i, j = 0, ..., n-1,\ i \neq j.$$

We consider the following algorithm for the minimization of the function f over $\mathbb{R}^n$:

1: Choose $x^0 \in \mathbb{R}^n$ and set $g^0 := Ax^0 - b$.

2: If $g^k = 0$: STOP. Set $m := k$. Then x^m is the global minimum of f.

3: Set

$$t_k := -\frac{(g^k)^T d^k}{(d^k)^T Ad^k}, x^{k+1} := x^k + t_k d^k, g^{k+1} := g^k + t_k Ad^k.$$

4: Set $k := k + 1$ and go to Step 2.

Show by means of complete induction over k: if $g^0, ..., g^k \neq 0$, then x^{k+1} is an optimal solution of the problem

$$(P_k) \qquad \text{minimize } f(x), x \in x^0 + \operatorname{span}\{d^0, ..., d^k\}.$$

Here, $\operatorname{span}\{d^0, ..., d^k\}$ denotes the linear subspace generated by the vectors $\{d^0, ..., d^k\}$. Since $x^0 + \operatorname{span}\{d^0, ..., d^{n-1}\} = \mathbb{R}^n$, the algorithm stops after $m = n$ steps at latest and x^m is the global minimizer of f.

Hint. Show that $x^{k+1} \in x^0 + \operatorname{span}\{d^0, ..., d^k\}$ and $(g^{k+1})^T d^i = 0, i = 0, ..., k$. (4 points)

30. Replace in the Fletcher-Reeves algorithm (Algorithm 7.5 in the lecture notes) β_k^{FR} by (*the Myers formula*)

$$\beta_k^M := -\frac{\|\nabla f(x^{k+1})\|^2}{\nabla f(x^k)^T d^k}.$$

Show that: if f is continuously differentiable and bounded from below, and for every $k \geq 0$ it holds $\nabla f(x^k) \neq 0$, then the algorithm is well-defined.

Hint. Prove that $\nabla f(x^k)^T d^k < 0$ for all $k \geq 0$ and use Theorem 5.5 in the lecture notes. (3 points)

31. The mapping $\|\cdot\| : \mathbb{R}^{n\times n} \to \mathbb{R}$ is called *matrix norm* if the following statements hold:

(i) $\|A\| = 0 \Leftrightarrow A = 0$;

(ii) $\|A\| \geq 0$ for all $A \in \mathbb{R}^{n\times n}$;

(iii) $\|\alpha A\| = |\alpha|\|A\|$ for all $\alpha \in \mathbb{R}$ and all $A \in \mathbb{R}^{n\times n}$;

(iv) $\|A + B\| \leq \|A\| + \|B\|$ for all $A, B \in \mathbb{R}^{n\times n}$.

(a) Let $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ be an arbitrary vector norm on $\mathbb{R}^n$. Show that

$$\|A\| := \max\{\|Ax\| : \|x\| = 1\}$$

defines a matrix norm. This is is the so-called matrix norm induced by a vector norm.

(b) Show that for a vector norm and the matrix norm induced by this vector norm it holds

$$\|Ax\| \leq \|A\|\|x\|, \; \forall A \in \mathbb{R}^{n\times n} \; \forall x \in \mathbb{R}^n$$

and

$$\|AB\| \leq \|A\|\|B\|, \; \forall A, B \in \mathbb{R}^{n\times n}.$$

(3 points)

32. (a) Let be $M \in \mathbb{R}^{n\times n}$ with $\|M\| < 1$. Show that $I - M$ is regular and

$$\|(I - M)^{-1}\| \leq \frac{1}{1 - \|M\|}.$$

(b) Let be $A, B \in \mathbb{R}^{n\times n}$ with $\|I - BA\| < 1$. Show that A and B are regular and that

$$\|B^{-1}\| \leq \frac{\|A\|}{1 - \|I - BA\|} \text{ and } \|A^{-1}\| \leq \frac{\|B\|}{1 - \|I - BA\|}.$$

(3 points)

33. Let be $f : \mathbb{R} \to \mathbb{R}$, $f(x) = (1/4)x^4$. Show that the local Newton algorithm (Algorithm 8.8 in the lecture notes) converges to the unique global minimum of f for every starting point $x^0 \in \mathbb{R}$. (2 points)

34. The local Newton algorithm (Algorithm 8.8 in the lecture notes) is invariant with respect to affine-linear transformation. In other words: let $A \in \mathbb{R}^{n\times n}$ be regular and $c \in \mathbb{R}^n$, $\{x^k\}_{k\geq 0}$ be the sequence by the local Newton algorithm for minimizing a function f with start vector x^0, and $\{y^k\}_{k\geq 0}$ be the sequence generated by the local Newton algorithm for minimizing the function $g(y) := f(Ay + c)$ with start vector y^0; then it holds

$$x^0 = Ay^0 + c \Rightarrow x^k = Ay^k + c, \; \forall k \geq 0$$

(3 points)

35. Implement the modified Polak-Ribiere algorithm in MATLAB. Create a file `mod_polak_ribiere.m` with

```
function [x, Iter]=mod_polak_ribiere(func, x0, kmax, sigma, beta, epsilon,
                                     delta1, delta2)
```

as first line. Here, `func` denotes a function handle, `x0` the starting vector, `kmax` the maximal number of allowed iterations, `epsilon` the parameter for the stopping criterion, and `sigma`, `beta`, `delta1` and `delta2` the parameters used for the determination of the step size. A matrix $X = [x0, x1, x2, ..]$ containing the iteration history and the number of performed iterations `Iter` should be returned.

The implemented algorithm should be tested for `epsilon`$=10^{-5}$, `kmax`$=500$, `sigma`$=10^{-4}$, `beta`$=0.5$, `delta1`$=0.1$ and `delta2`$=10$ on the following functions and starting values:

(a) $f(x_1, x_2) = (1-x_1)^2 + 100(x_2 - x_1^2)^2$ (Rosenbrock function) for `x0`$=(-1.2, 1)^T$.

(b) $f(x_1, x_2, x_3, x_4) = \sum_{i=1}^4 \left(4 - \sum_{j=1}^4 \cos x_j + i(1-\cos x_i) - \sin x_i\right)^2$ (trigonometric function) for `x0`$=\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)^T$.

(c) $f(x_1, x_2) = (x_1 - 10^6)^2 + (x_2 - 2\cdot 10^6)^2 + (x_1 x_2 - 2)^2$ (Brown function) for `x0`$=(1, 1)^T$.

(d) $f(x_1, x_2, x_3, x_4) = 100(x_2 - x_1^2)^2 + (1-x_1)^2 + 90(x_4 - x_3^2)^2 + (1-x_3)^2 + 10(x_2 + x_4 - 2)^2 + \frac{1}{10}(x_2 - x_4)^2$ (Wood function) for `x0`$=(-3, -1, -3, -1)^T$.

(4 points)

36. Implement the local Newton algorithm in MATLAB. To do this, create a file `local_armijo.m` with

```
function [X, Iter] =local_newton(func, x0, kmax, epsilon)
```

as first line. Here, `func` denotes a function handle, `x0` the starting vector, `kmax` the maximal number of allowed iterations, and `epsilon` the parameter for the stopping criterion. A matrix $X = [x0, x1, x2, ..]$ containing the iteration history and the number of performed iterations `Iter` should be returned.

The implemented algorithm should be tested for `epsilon`$=10^{-6}$ and `kmax`$=200$ on the following functions and starting values:

(a) $f(x_1, x_2) = (1-x_1)^2 + 100(x_2 - x_1^2)^2$ (Rosenbrock function) for `x0`$=(-1.2, 1)^T$.

(b) $f(x_1, x_2, x_3, x_4) = \sum_{i=1}^4 \left(4 - \sum_{j=1}^4 \cos x_j + i(1-\cos x_i) - \sin x_i\right)^2$ (trigonometric function) for `x0`$=\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)^T$.

(c) $f(x_1, x_2) = (x_1 - 10^6)^2 + (x_2 - 2\cdot 10^6)^2 + (x_1 x_2 - 2)^2$ (Brown function) for `x0`$=(1, 1)^T$.

(d) $f(x_1, x_2, x_3, x_4) = 100(x_2 - x_1^2)^2 + (1-x_1)^2 + 90(x_4 - x_3^2)^2 + (1-x_3)^2 + 10(x_2 + x_4 - 2)^2 + \frac{1}{10}(x_2 - x_4)^2$ (Wood function) for `x0`$=(-3, -1, -3, -1)^T$.

(e) $f(x) = \sqrt{1 + x^2}$ for `x0`$=2$, `x0`$=1$ and `x0`$=\frac{1}{2}$.

(4 points)

37. Implement the globalized Newton algorithm in MATLAB. To do this, create a file `global_newton.m` with

```
function [X, Iter]=global_newton(func, x0, kmax, sigma, beta, epsilon, rho,
                                  p)
```

as first line. Here, `func` denotes a function handle, `x0` the starting vector, `kmax` the maximal number of allowed iterations, `sigma` and `beta` the parameters for the determination of the Armijo step size, `epsilon` the parameter for the stopping criterion, and `rho` and `p` the parameters used in the algorithm. A matrix $X = [x0, x1, x2, ..]$ containing the iteration history and the number of performed iterations `Iter` should be returned.

The implemented algorithm should be tested for `epsilon`$=10^{-6}$, `kmax`$=200$, `rho`$=10^{-8}$, `p`$=2.1$, `sigma`$=10^{-4}$ and `beta`$=0.5$ on the following functions and starting values:

(a) $f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$ (Rosenbrock function) for `x0`$=(-1.2, 1)^T$.

(b) $f(x_1, x_2, x_3, x_4) = \sum_{i=1}^4 \left(4 - \sum_{j=1}^4 \cos x_j + i(1 - \cos x_i) - \sin x_i\right)^2$ (trigonometric function) for `x0`$=\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)^T$.

(c) $f(x_1, x_2) = (x_1 - 10^6)^2 + (x_2 - 2 \cdot 10^6)^2 + (x_1 x_2 - 2)^2$ (Brown function) for `x0`$=(1, 1)^T$.

(d) $f(x_1, x_2, x_3, x_4) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10(x_2 + x_4 - 2)^2 + \frac{1}{10}(x_2 - x_4)^2$ (Wood function) for `x0`$=(-3, -1, -3, -1)^T$.

(e) $f(x) = \sqrt{1 + x^2}$ for `x0`$=2$, `x0`$=1$ and `x0`$=\frac{1}{2}$.

(5 points)

The generated files should be sent by December 13, 2019 to `radu.bot@univie.ac.at`.