Convex Optimization and Modeling	Jun.-Prof. Matthias Hein
Solution of Exercise Sheet 8	09.06.2010

Exercise 18 - Implementation of Gradient Descent and Newton method

a. **(6 Points)**

- Implement gradient descent with the Armijo rule in Matlab,
- Implement the Newton method (stepsize selection with Armijo rule) in Matlab,
- use separate functions for
  1. `getStepSize`: stepsize selection with Armijo rule. One chooses $\beta \in (0,1)$ and $\sigma \in (0,1)$ and $s > 0$. Then the stepsize α^k is defined as $\alpha^k = \beta^m s$, where m is the first non-negative integer such that
     $$f(x^{k+1}) - f(x^k) = f(x^k + \beta^m s d^k) - f(x^k) \leq \sigma \beta^m s \left\langle \nabla f(x^k), d^k \right\rangle .$$
     We fix $s = 1$ and use only the parameters σ and β.
     input: current point, current gradient, descent direction, β, σ.
     output: stepsize.
  2. `f`: returns the function value evaluated at a point
     input: a point x,
     output: the objective evaluated at x.
  3. `gradf`: returns the gradient of f evaluated at a point
     input: a point x,
     output: the gradient of f evaluated at x.
  4. `Hessf`: returns the Hessian of f evaluated at a point
     input: a point x,
     output: the Hessian of f evaluated at x.

  Sample matlab files `NewtonExercise.m` and `DescentExercise.m` can be downloaded from the course webpage.

As a function f use the example from the book (Equation 9.20),

$$f(x_1, x_2) = e^{x_1 + 3x_2 - 0.1} + e^{x_1 - 3x_2 - 0.1} + e^{-x_1 + 0.1}.$$

As initial point take a random sample from a Gaussian (`x=randn(2,1)`).

$$\text{Stopping criterion:} \qquad \|\nabla f\| \leq 10^{-4}.$$

Test your code ! If it does not run $\Longrightarrow$ 0 points.

b. **(2 Points)** Run the gradient descent code for $\sigma = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and $\beta = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ for 10 different starting values for each set of parameters σ, β in the stepsize selection. Plot the average number of required steps in dependency of σ, β. Explain the plot.

c. **(2 Points)** Run the Newton method for $\sigma = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and $\beta = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ for 10 different starting values for each set of parameters σ, β in the stepsize selection. Plot the average number of required steps in dependency of σ, β. Explain the plot.

d. **(2 Points)** Verify the experiment done in BV (page 481). Use gradient descent with the norms P_1 and P_2 (see Equation 9.25, page 476) and run it once for the gradient descent for P_1 and P_2 and directly Newton's method. All runs with the **same starting point**. Plot $f(x^k) - p^*$ as on page 482 for all three cases.

Solution:

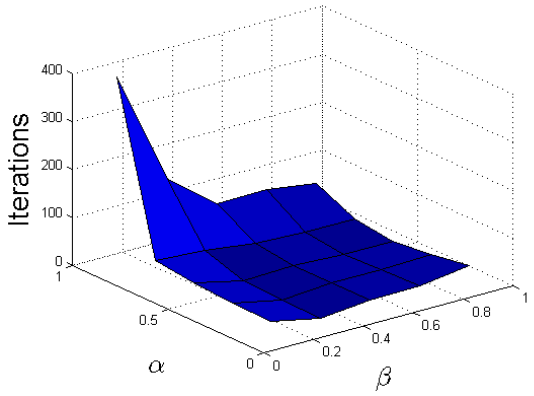

Figure 1: For this **particular problem** we see that a large σ is generally not favorable. The minimal number of steps can be achieved using a small value of σ and a moderately small value of β. Note that the number of steps is minimal for $\sigma = \frac{1}{2}$ which is what the bound predicts.

As a general result we see that values of $\sigma > \frac{1}{2}$ should be avoided.

d. In the last exercise we have fixed the initial vector with `randn('state',2)`.

a.

$\sigma \backslash \beta$	0.1	0.3	0.5	0.7	0.9
0.1	35	14	22	25	38
0.3	38	15	22	21	25
0.5	42	17	22	17	16
0.7	51	45	32	27	26
0.9	397	157	78	75	64

Table 1: The average number of iterations (rounded to the next integer) over 100 runs for different values of σ and β for the gradiend descent method.

b.

$\sigma \backslash \beta$	0.1	0.3	0.5	0.7	0.9
0.1	6	6	6	5	6
0.3	6	6	6	6	6
0.5	9	7	7	6	6
0.7	68	22	13	12	10
0.9	67	76	55	42	37

Table 2: The average number of iterations (rounded to the next integer) over 100 runs for different values of σ and β for the Newton method.

c.

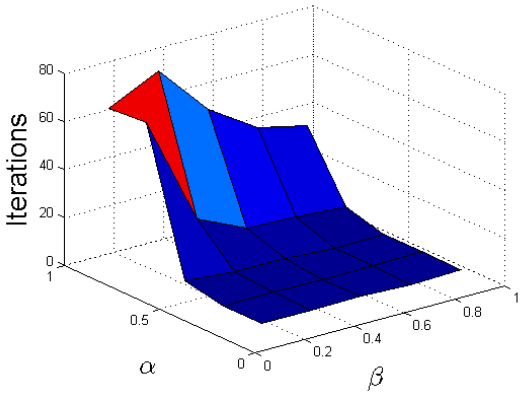

Figure 2: For this **particular problem** we see that a large σ is generally not favorable. The required number of iterations is very stable for $\sigma < 0.7$ for all values of β. This is in constrast to the descent method where the number of iterations varies much more.

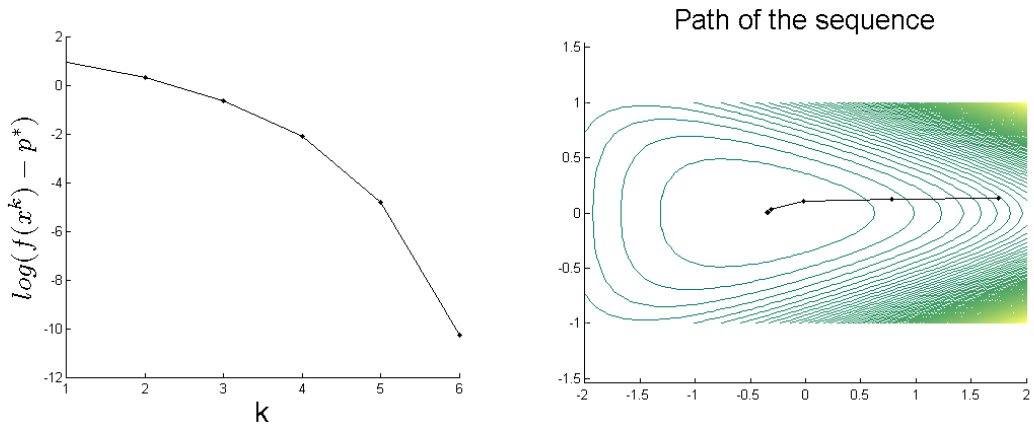

Figure 3: The Newton method converges in 5 steps.

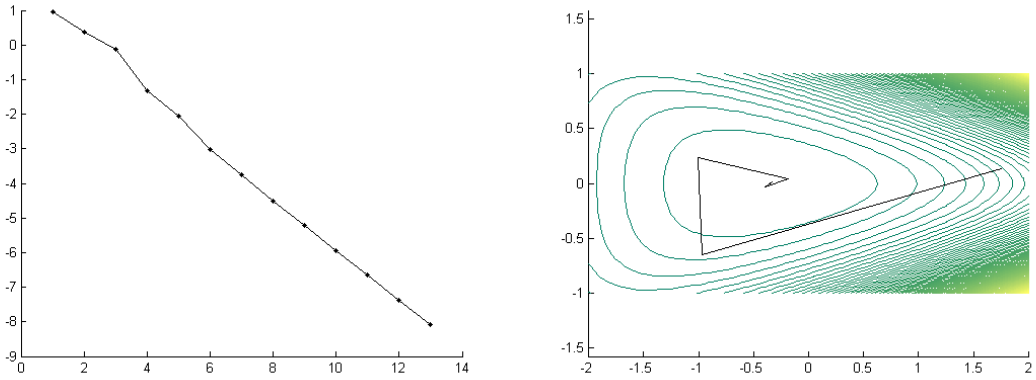

Figure 4: The steepest descent method with descent direction $d^k = -P_1^{-1}\nabla f$ converges in 13 steps.

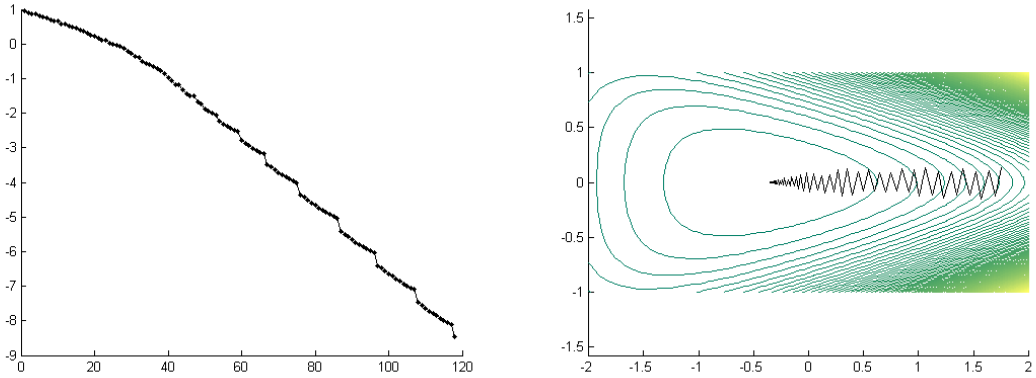

Figure 5: The steepest descent method with descent direction $d^k = -P_2^{-1}\nabla f$ converges in 118 steps. Note, the huge difference in the number of iterations. The reason for that is that the transformation with P_1 improves the condition number (P_1 is a good approximation of the Hessian) whereas P_2 worsens the condition number (P_2 is a bad approximation of the Hessian.