OCS Hints for Questions

Julian Wolf

January 29, 2017

Derivations

Die Antworten sind teilweise unvollständig, einerseits, weil er die Antworten als "Eh Klar" abgestempelt hat, anderer seits weil er so schnell durchging, dass ein Mitschreiben nicht mehr möglich war.

1. Draw level lines and arrows
   - objective function is the function we want to minimize
   - constraint set is a set of functions
   - optimal solution: find $f(x^*) \leq f(x), \forall x \in X$
   - level set: compareable to level lines of terrain, convex function => convex level set (but there are non convex fct with convex level sets),
2. - **Linear**: Objective Function and Constraints may only be linear $min\ c^T x, s.t.\ Ax \leq b, x \geq 0$
     Polynomial solvable
   - **Non Linear**: Objective Function and Constriants may be non linear $min\ \frac{1}{2}x^T Qx + c^T x, s.t.\ Ax \leq b, Ex = d$
     Q symmetrical and pos. definite, polynomial solvable
   - **Quadratic**: objective function is quadratic, constraints are linear $min_{x \in \mathbb{R}}\ f_0(x)$ `(objective)`,
     $s.t.\ f_i(x) \leq i = 0..m$ `(contraints)`
     polynomial time
   - **convex set**: $\alpha x + (1 - \alpha)y\ inX, \forall x, y \in X, \alpha \in [0, 1]$
   - **convex fct**: $f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y), \forall x, y \in X, \alpha \in [0, 1]$
3. - When hessian is strictly positive, it is a strict global maximum
   - **unconst Local minimum:** $f(x^*) \leq f(x), \forall x$ `with` $||x - x^*|| \leq \varepsilon$
   - **unconst global minimum:** $f(x^*) \leq f(x), \forall x \in \mathbb{R}$

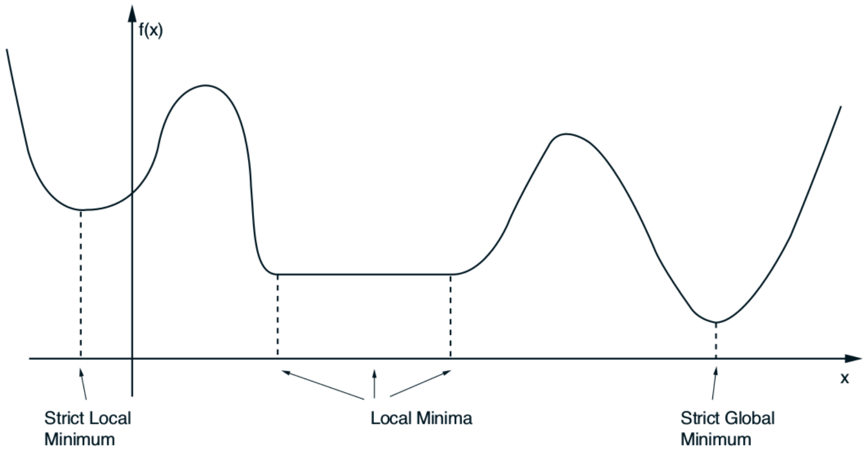

Figure 1: Local/Global minimas

4.
   - If positive and negative Eigenvalues, we can not define convexity
   - First order necessessary optimality condition: $\nabla f(x^*) = 0$
   - Second order necessessary optimality condition: $\nabla^2 f(x^*)$ is positive semi definite

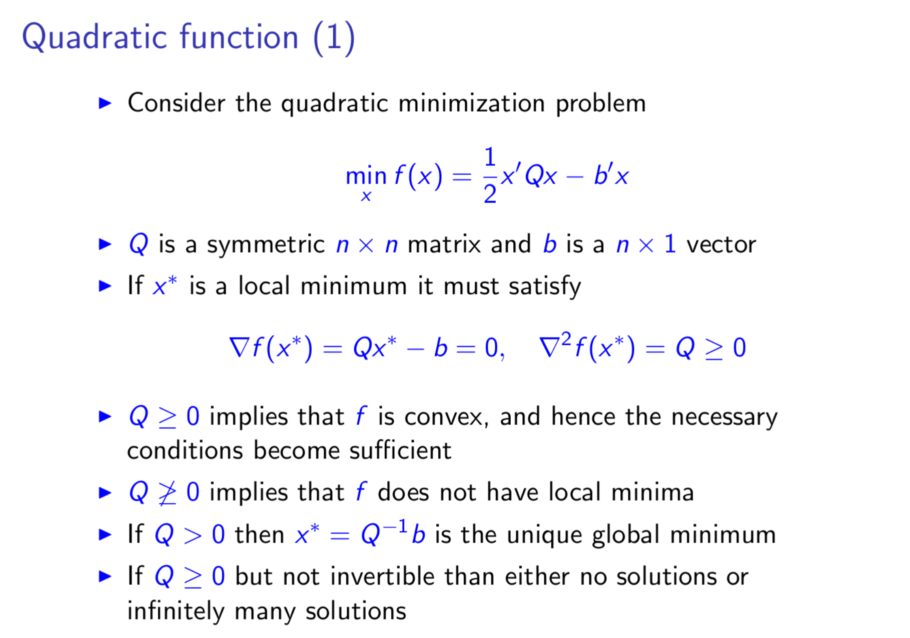

Figure 2: Different scenarios for Q

5.
- Descent direction: angle of step and derivation direction $< 90°$
- General form of gradient method:
  1. Choose an initial vector $x^0 \in \mathbb{R}^n$
  2. Choose a descent direction d^k that satisfies $\nabla f(x^k)'d^k < 0$
  3. Choose a positive step size α^k
  4. Compute the new vector as
  $x^{k+1} = x^k + \alpha^k d^k$
  5. Set $k = k + 1$ and goto 2

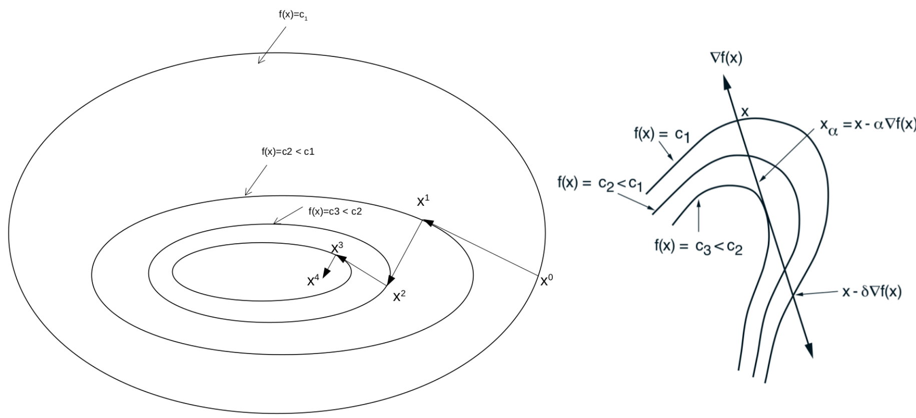

Figure 3: Simple descent direction

6. $d^k = -D^k \nabla f(x^k)$

   - Identity: $D^k = I$, = Gradient descent, zig zagging problem, very bad on Rosenbrock Fct
   - Hessian: $D^k = \nabla^2 f(x^k)$, = Newtons method, very fast convergence, very good on rosenbrock, unstable in despite of initial values (may diverge or find local maxima instead of minima), con: calculation of inverse of hessian - very expensive in large networks
   - Diagonal Hessian (approximation of Newton): $d_i^k \approx \left(\frac{\partial^2 f(x^k)}{(\partial x_i)^2}\right)^{-1}$, very bad performance on Rosenbrock,
   - Gauss Newton method: Too complicated to remember, replace D^k with non linear least square problem, even better performance on rosenbrock then newton, con: again calculation of inverse, but not of hessian
   - **Step size** α:
     - **Minimization rule:** choose α such that $f(x+\alpha d)$ is minimized along d. Hard if f is complicated
     - **Limited minimization rule:** iterative: start small and increase size of α until $f(x)$ is bigger then before, then choose the previous. Easy to implement
     - **Armijo rule:** it is not sufficient that $f(x^{k+1}) < f(x^k)$, thus, the step sizes $\beta^m s$ for $m = 0, 1, ...$ are chosen such that the energy decrease is sufficiently large (dependent on derivation of $f(x)$, formula too complicated), or graphical:

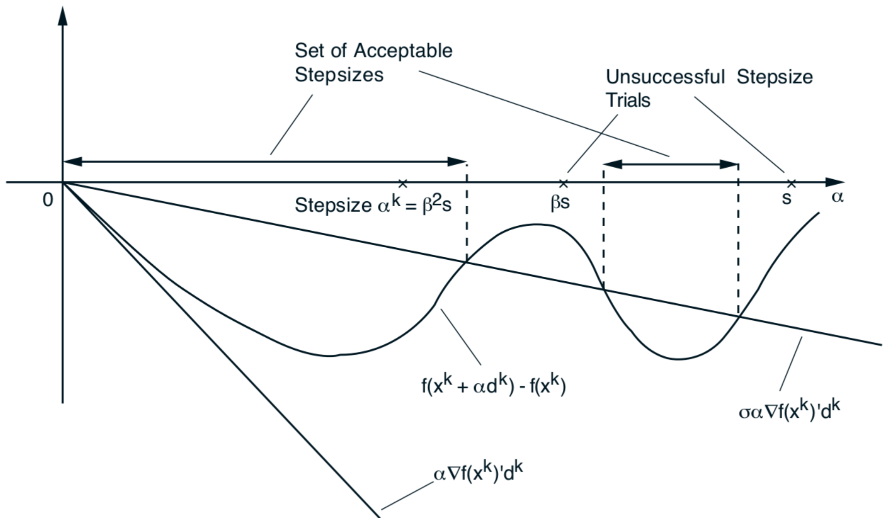

Figure 4: Graphical representation of the idea of Armijo

7. • there exists a definite real number such that, for every pair of points on the graph of this function, the absolute value of the slope of the line connecting them is not greater than this real number
   • (There is something missing here)

8. • **Linear:** $\limsup_{k\to\infty} \frac{e(x^{k+1})}{e(x^k)} \leq \beta$ (blue line)
   • **superlinear:** $\limsup_{k\to\infty} \frac{e(x^{k+1})}{e(x^k)^p} < \infty$ (red line)
   • **sublinear:** $\limsup_{k\to\infty} \frac{e(x^{k+1})}{e(x^k)} = 1$ (black line)

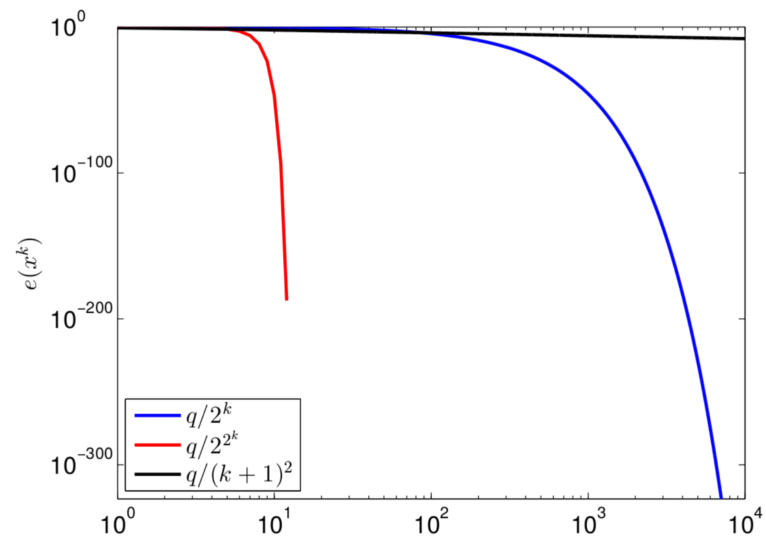

Figure 5: Graphical representation of linear, superinear and sublinear convergence

9. - $\tilde{g}(x, x^k) = g(x^k) + \nabla g(x^k)'(x - x^k)$
   - is affine: because the derivation of the function is independent to a move of constants?!?!?

10. **Linear**:

    - pointcloud, best fitting by polynomial equations
    - Solve least square problem to find optimal parameters
    - $\frac{1}{2}||Ax - z||^2$

    Non-Linear:

    - Measure distance to beacons to get our own unkown location
    - non linear least squares problem
    - $min_x \frac{1}{2} \sum_{i=1^m} (d_i - \sqrt{(x_i - p_1^i)^2 + (x_2 - p_2^i)^2})^2$

    Gauss-Newton:

    - Uses $(\nabla g(x^k) \nabla g(x^k)')^{-1}$ instead of hessian
    - Approximates Current point with parabola and minimizes this subproblem (as plain Newton)

    (check this shit out)

11. - incremental of gauss newton - incremental growing least squares estimate

- example watertank: many measurements with noise - a very good and fast convergence to correct level is reached
- extended Kalman Filter: input data behaves according to a function

12.
- Q positive definite: $\left(\frac{\sqrt{L/I}-1}{\sqrt{L/I}+1}\right)^n ||x^0 - x*||$
- Q positive semidefinite: not motivated to write formula
- Best method: conjugate gradient method (by Polyak) see next question

13.
- Motivation: converge faster the GD but avoid Newton overhead
- Q-conjugate if: $d^i Q d^j = 0, \forall i, j$ `with` $i \neq j$
- Algorithm: Use Gram-Schmidt to find conjugate direction, calculate new search direction (easy as all but one coefficient are zero), choose α^k by minimization method, algorithm terminates after at most n steps.

(I think this is the algorithm where every single dimension is differenciated and optimized - hence the n termination)

14.
- Heavy-ball: Idea is like in physics: a ball uses its momentum it gained beforehand to overcome small increases or flat areas of its way
- Problem: function needs to be strongly $\mu > 0$ convex and twice continously differentiable
- Nesterov overcomes the twice diff. and the $\mu > 0$ problem by a dynamic choice of overrelaxation param $\beta^k = \frac{t_k - 1}{t_k + 1} \to 1$, gradient is evaluated at extrapolated point
- Both algorithms are optimal
- HB is optimal like the Q positive semidefinite optimality, Nesterov also yields linear convergence rate on strongly convex sets

15.
- the gradient $\nabla f(x^*)$ makes an angle ≤ 90 in all feasable points
- this condition is in general not reachable

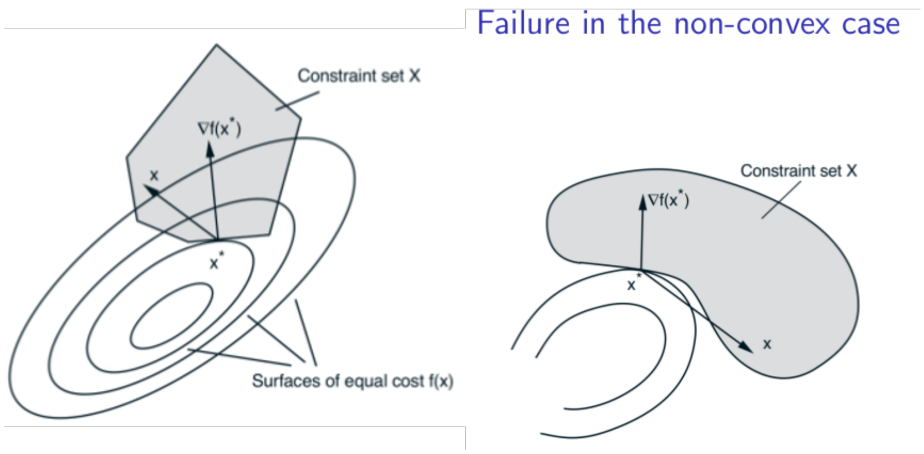

Figure 6: Graphical representation of convex and non-convex set

16.
    - z is a fixed vector, find vector x^* in a closed convex set X
    - $min_x f(x) = ||x - z||^2$

17. middle/end of pages slide 10 - start in interior and just take small steps $->$ we can ignore constraint under these conditions
    - Given a feasible vector x, a feasible direction at x is a vector d such that the vector $x + \alpha d$ is feasible for all sufficiently small $\alpha > 0$.
    - a feasable method generates starts at x^0 and generates multiple such points x^{k+1}

18.
    - Conditional gradient solves subproblem with linear cost, gradient projection method solves quadratic cost fct
    - The conditional gradient method generates the point $\overline{x}^k$ by finding a feasible point which is furthest way from x^k along the negative gradient direction $-\nabla f(x^k)$.

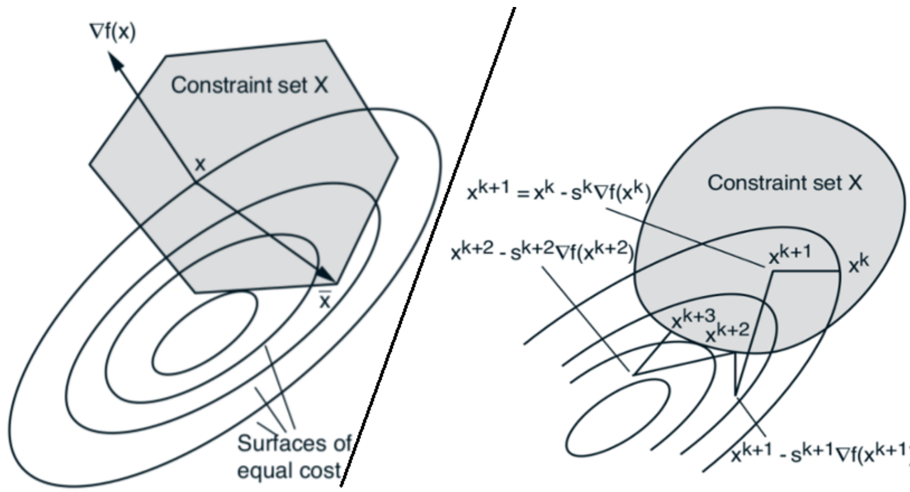

Figure 7: Graphical representation of conditional (left) and projective (right) method

19.
    - iterative method $x^{k+1} = x^k + \alpha^k (H^k)^{-1}$(`big AHA formula`)
    - affine scaling: choose $H^k = (X^k)^{-2}$, $X^k = diag(x_1, ..., x_n)$ leads to $y^{k+1} = y^k + \alpha^k (AX^kA')^{-1}b$, α^k ensures $x^{k+1} > 0$

20.
    - Interpretation 1: The gradient of the cost function $\nabla f(x^*)$ belongs to the subspace spanned by the gradients of the constraint functions $\nabla h_i(x^*)$
    - Interpretation 2: The cost gradient $\nabla f(x^*)$ is orthogonal to the subspace of first order feasible directions
    - for failure see figure 8. The Eigenvectors are lineary dependent, we loose one dimension and thus we can not optimize the problem (at least I think so)

21. solve $min_x \frac{1}{2}||x - y||^2$

22. solve $min_x \frac{1}{2}||x - y||^2$ `s.t.` $a^T x = b$

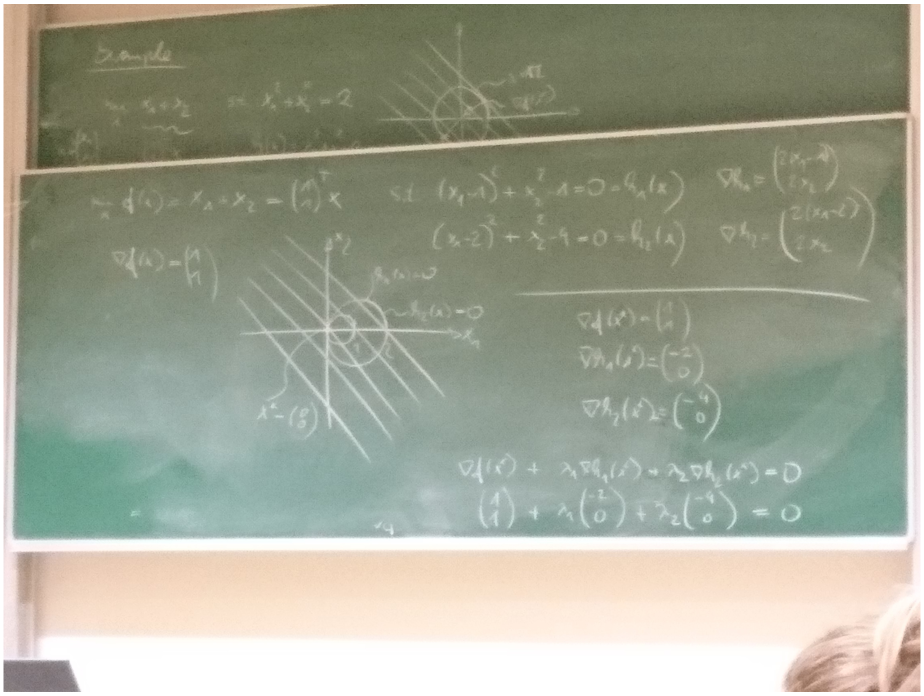

Figure 8: Example1, 24.01.2017

Figure 9: Example 2, 24.01.2017