ETH zürich

Short Course on Constrained Nonlinear Optimization

Part II

Christian Boehm

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad g(x) \leq 0, \quad h(x) = 0. \tag{P}$$

Two types of optimality conditions:

- Projection formula

$$\bar{x} = P_X\big(\bar{x} - \gamma \nabla f(\bar{x})\big).$$

- KKT conditions

$$\begin{aligned} \nabla f(\bar{x}) + \nabla g(\bar{x})\bar{\lambda} + \nabla h(\bar{x})\bar{\mu} &= 0, \\ h(\bar{x}) &= 0, \\ g(\bar{x}) \leq 0, \quad \bar{\lambda} \geq 0, \quad \bar{\lambda}^T g(\bar{x}) &= 0. \end{aligned}$$

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad g(x) \leq 0, \quad h(x) = 0. \tag{P}$$

Two types of optimality conditions:

- Projection formula

$$\bar{x} = P_X\big(\bar{x} - \gamma \nabla f(\bar{x})\big).$$

- KKT conditions

$$\begin{aligned} \nabla f(\bar{x}) + \nabla g(\bar{x})\bar{\lambda} + \nabla h(\bar{x})\bar{\mu} &= 0, \\ h(\bar{x}) &= 0, \\ g(\bar{x}) \leq 0, \quad \bar{\lambda} \geq 0, \quad \bar{\lambda}^T g(\bar{x}) &= 0. \end{aligned}$$

Today: Algorithms to find points satisfying these conditions.

- Projected Gradient Method
- Penalty Method

well suited if constraints are cheap to evaluate and projection is possible

- Projected Gradient Method
- Penalty Method

well suited if constraints are cheap to evaluate and projection is possible

- Sequential Quadratic Programming
- Interior Point Methods

well suited for general nonlinear problems with many active constraints

- Projected Gradient Method
- Penalty Method

well suited if constraints are cheap to evaluate and projection is possible

- Sequential Quadratic Programming
- Interior Point Methods

well suited for general nonlinear problems with many active constraints

Problem classification:

- Decision Tree of Optimization Software:
  http://plato.la.asu.edu/guide.html
- NEOS Guide:
  http://www.neos-guide.org/Optimization-Guide

Limitations

Eduard Imhof, Auf dem Säntis, Blick gegen Abend.
http://www.library.ethz.ch/

- Algorithms search only for necessary not sufficient conditions.

- Algorithms search only for necessary not sufficient conditions.
- For non-convex problem: convergence to local not global minima.

- Algorithms search only for necessary not sufficient conditions.
- For non-convex problem: convergence to local not global minima.
- $f(x^{k+1}) < f(x^k)$ does not imply $\|x^{k+1} - \bar{x}\| < \|x^k - \bar{x}\|$.

- Algorithms search only for necessary not sufficient conditions.
- For non-convex problem: convergence to local not global minima.
- $f(x^{k+1}) < f(x^k)$ does not imply $\|x^{k+1} - \bar{x}\| < \|x^k - \bar{x}\|$.

Remedies:

- Exploit "local convexity" around the global minimum with good initial value x^0.
- "Convexify the problem" with the choice of objective function and constraints.

Idea: Apply steepest descent method but project the path onto X.

Idea: Apply steepest descent method but project the path onto X.

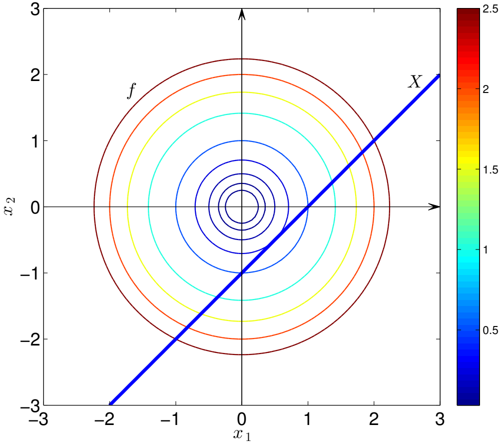
Algorithm

- Choose $x^0 \in X$.

  For $k = 1, 2, 3, \ldots$
- If $\|x^k - P_X(x^k - \nabla f(x^k))\| < \varepsilon$ STOP.
- Set $s^k = -\nabla f(x)$.
- Choose step-size σ_k by a projected Armijo-rule such that

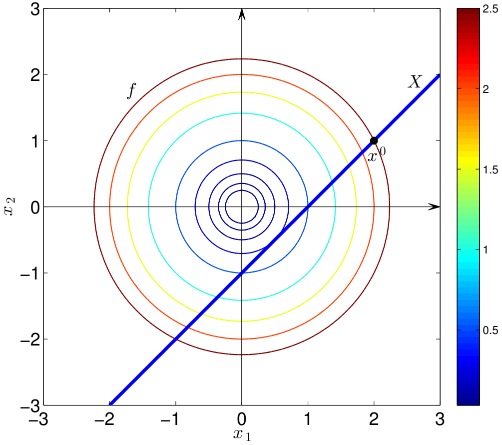
$$f(P_X(x^k + \sigma_k s^k)) < f(x^k).$$

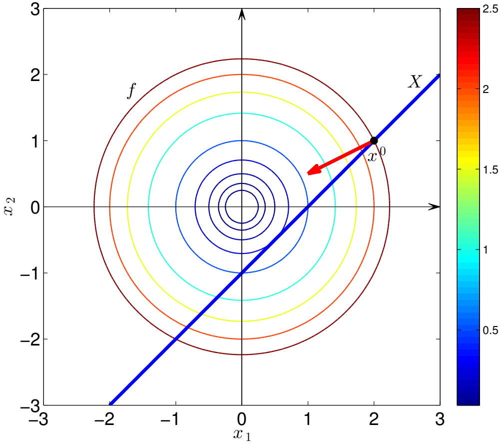
- Update $x^{k+1} = P_X(x^k + \sigma_k s^k)$.



f
X
x_1
x_2
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
2.5
2
1.5
1
0.5

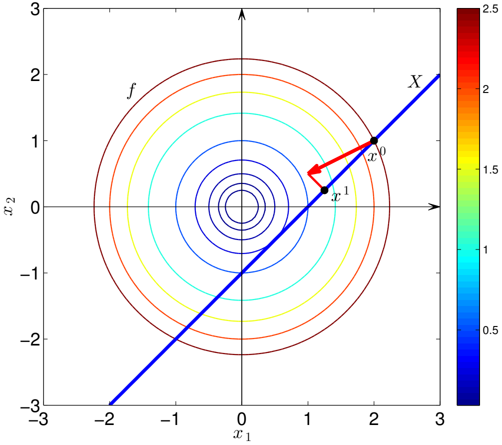
ETH zürich

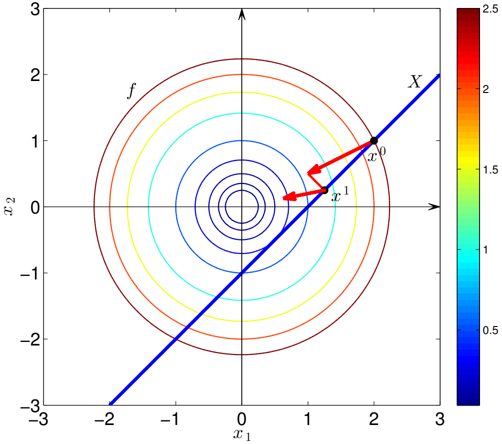




f
X
x0
x1
x2
3
2
1
0
−1
−2
−3
2.5
2
1.5
1
0.5

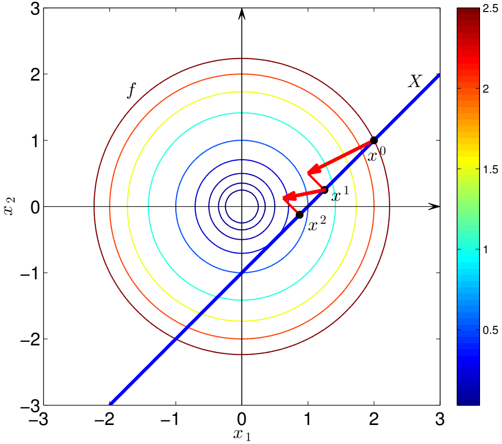
Projected Gradient Method - Example

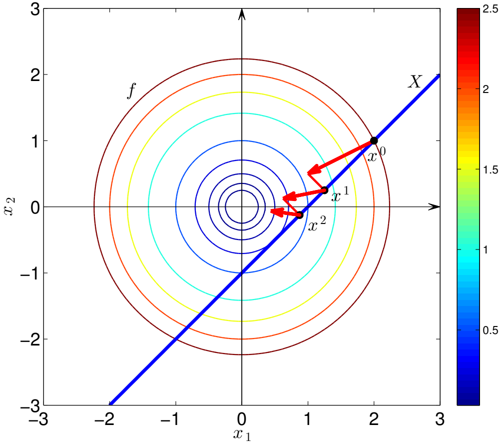




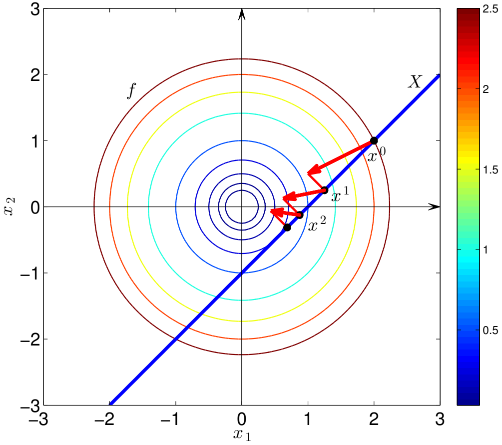
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
x_1
x_2
f
X
x^0
x^1
2.5
2
1.5
1
0.5

Projected Gradient Method - Example

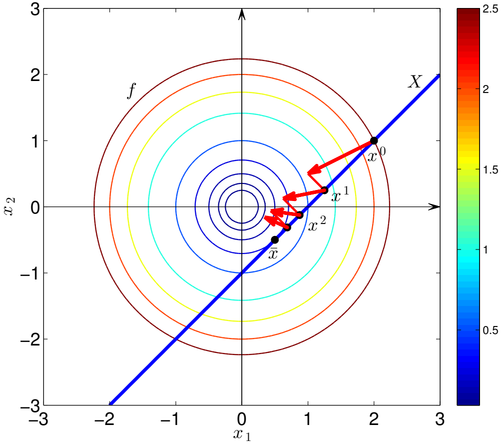




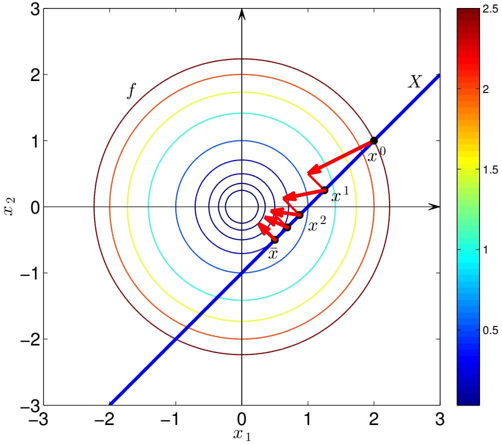
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
x_1
x_2
f
X
x^0
x^1
x^2
2.5
2
1.5
1
0.5


3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
x_1
x_2
f
X
x^0
x^1
x^2
2.5
2
1.5
1
0.5

Projected Gradient Method - Example



Projected Gradient Method - Example



How to compute $y_p = P_X(y)$?

How to compute $y_p = P_X(y)$?

In fact,

$$y_p = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \, \|x - y\|^2 \quad \text{s.t.} \quad x \in X. \qquad (\star)$$

How to compute $y_p = P_X(y)$?

In fact,

$$y_p = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; \|x - y\|^2 \quad \text{s.t.} \quad x \in X. \qquad (\star)$$

Idea: Derive KKT-conditions for $(\star)$ and solve auxiliary problem.

How to compute $y_p = P_X(y)$?

In fact,

$$y_p = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \|x - y\|^2 \quad \text{s.t.} \quad x \in X. \qquad (\star)$$

Idea: Derive KKT-conditions for $(\star)$ and solve auxiliary problem.

Example:

$$X = \{x \in \mathbb{R}^n : Ax = b\} \text{ with } A \in \mathbb{R}^{p \times n}, b \in \mathbb{R}^p.$$

How to compute $y_p = P_X(y)$?

In fact,

$$y_p = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \|x - y\|^2 \quad \text{s.t.} \quad x \in X. \qquad (\star)$$

Idea: Derive KKT-conditions for $(\star)$ and solve auxiliary problem.

Example:

$$X = \{x \in \mathbb{R}^n : Ax = b\} \text{ with } A \in \mathbb{R}^{p \times n}, b \in \mathbb{R}^p.$$

Here, the KKT conditions yield

$$y_p = \big(I - A^T \underbrace{(AA^T)^{-1}}_{\in \mathbb{R}^{p \times p}} A\big)y + A^T(AA^T)^{-1}b.$$

How to compute $y_p = P_X(y)$?

In fact,

$$y_p = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \; \|x - y\|^2 \quad \text{s.t.} \quad x \in X. \qquad (\star)$$

Idea: Derive KKT-conditions for $(\star)$ and solve auxiliary problem.

Example:

$$X = \{x \in \mathbb{R}^n : Ax = b\} \text{ with } A \in \mathbb{R}^{p \times n}, b \in \mathbb{R}^p.$$

Here, the KKT conditions yield

$$y_p = \left(I - A^T \underbrace{(AA^T)^{-1}}_{\in \mathbb{R}^{p \times p}} A\right) y + A^T (AA^T)^{-1} b.$$

This requires only matrix-vector operations involving n!

Seismic Tomography with additional constraints:

$$\min_{m} \underbrace{\chi(u(m))}_{\text{misfit}} \quad \text{s.t.} \quad g(m) \leq 0, \quad h(m) = 0,$$

where $u(m)$ solves the elastic wave equation $\mathcal{L}(u, m) = F$.

Seismic Tomography with additional constraints:

$$\min_m \underbrace{\chi(u(m))}_{\text{misfit}} \quad \text{s.t.} \quad g(m) \leq 0, \quad h(m) = 0,$$

where $u(m)$ solves the elastic wave equation $\mathcal{L}(u, m) = F$.

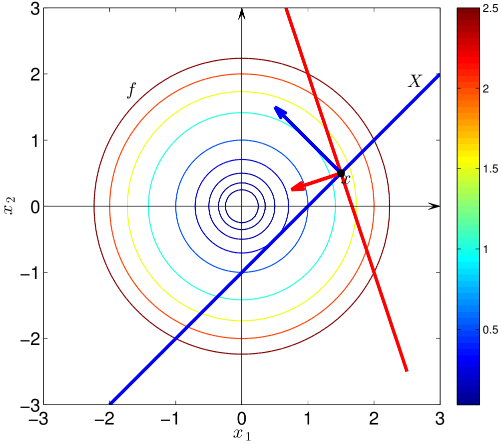
Auxiliary projection problem:

$$\min_m \|m - \hat{m}\|^2 \quad \text{s.t.} \quad g(m) \leq 0, \quad h(m) = 0.$$

Constraints only act on m and not an u.

Hence, no simulation is required to solve the auxiliary problem.

Can we replace the gradient by a different descent method?



Idea: Add a penalty term to the objective that penalizes infeasibility.

Idea: Add a penalty term to the objective that penalizes infeasibility.

Penalized Problem

For a fixed $\gamma \in (0, \infty)$ define

$$\min_{x \in \mathbb{R}^n} \quad f_\gamma(x) := f(x) + \gamma\phi(x), \qquad (P_\gamma)$$

with

$$\phi(x) := \frac{1}{2}\sum_{i=1}^{m}(\max\{g_i(x), 0\})^2 + \frac{1}{2}\sum_{i=1}^{p}(h_i(x))^2.$$

Idea: Add a penalty term to the objective that penalizes infeasibility.

Penalized Problem

For a fixed $\gamma \in (0, \infty)$ define

$$\min_{x \in \mathbb{R}^n} \; f_\gamma(x) := f(x) + \gamma\phi(x), \qquad (P_\gamma)$$

with

$$\phi(x) := \frac{1}{2}\sum_{i=1}^{m}(\max\{g_i(x), 0\})^2 + \frac{1}{2}\sum_{i=1}^{p}(h_i(x))^2.$$

Thus, we obtain an unconstrained optimization problem (P_γ).

Idea: Add a penalty term to the objective that penalizes infeasibility.

Penalized Problem

For a fixed $\gamma \in (0, \infty)$ define

$$\min_{x \in \mathbb{R}^n} \; f_\gamma(x) := f(x) + \gamma \phi(x), \qquad (P_\gamma)$$

with

$$\phi(x) := \frac{1}{2} \sum_{i=1}^{m} (\max\{g_i(x), 0\})^2 + \frac{1}{2} \sum_{i=1}^{p} (h_i(x))^2.$$

Thus, we obtain an unconstrained optimization problem (P_γ).

Furthermore,

$$f_\gamma(x) = f(x) \qquad \text{if} \quad x \in X,$$

$$\nabla f_\gamma(x) = \nabla f(x) \qquad \text{if} \quad x \in X.$$

Algorithm

- Choose $\gamma_0 > 0$.

  For $k = 1, 2, 3, \ldots$
- Solve (P_{γ_k}) approximately and obtain x^k.
- If $x^k \in X$ STOP.
- Choose $\gamma_{k+1} > \gamma_k$.

Algorithm

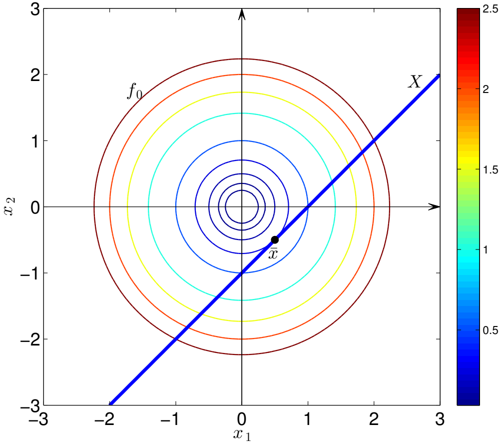
- Choose $\gamma_0 > 0$.

  For $k = 1, 2, 3, \ldots$
- Solve (P_{γ_k}) approximately and obtain x^k.
- If $x^k \in X$ STOP.
- Choose $\gamma_{k+1} > \gamma_k$.

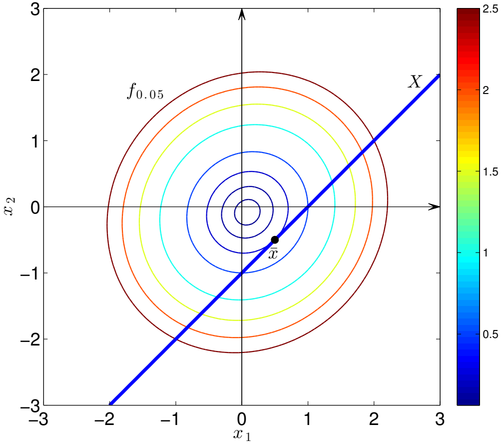
Remarks:

- x^k can be used as initial point for $(P_{\gamma_{k+1}})$.
- $\lambda_i^k := \gamma_k \max\{g_i(x^k), 0\}, \quad \mu_i^k := \gamma_k h_i(x^k)$

  converge to optimal Lagrangian multipliers.

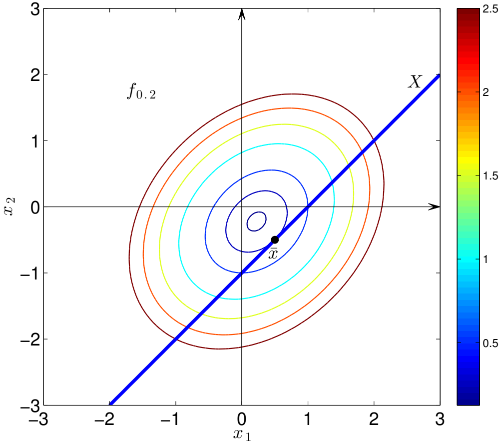


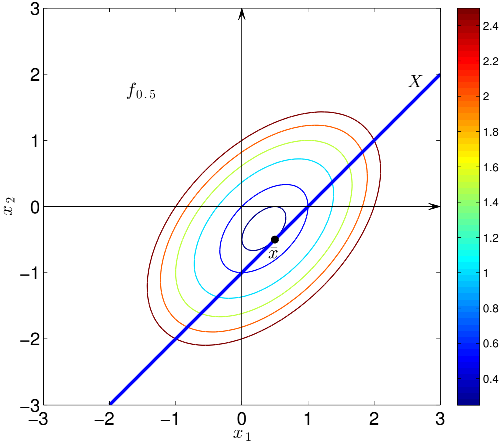
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
x_1
x_2
f_0
X
$\bar{x}$
2.5
2
1.5
1
0.5



$f_{0.05}$
X
$\bar{x}$
x_1
x_2
3
2
1
0
−1
−2
−3
2.5
2
1.5
1
0.5

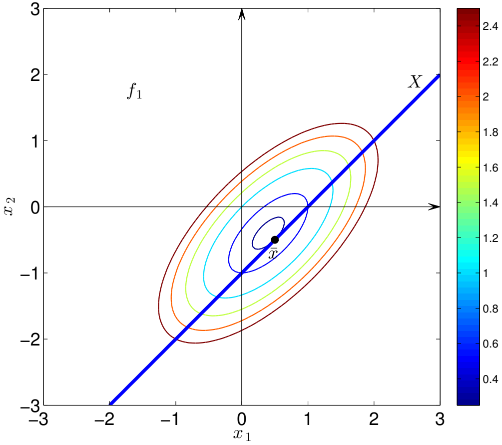
Penalty Method - Example

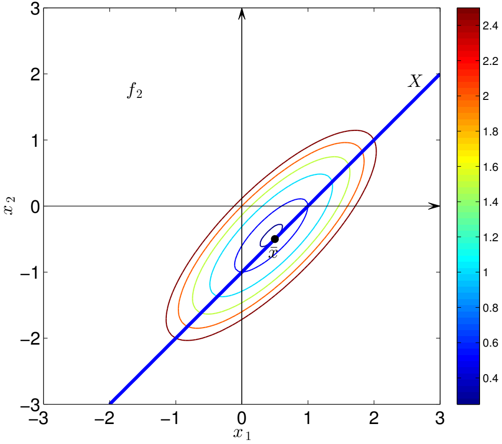




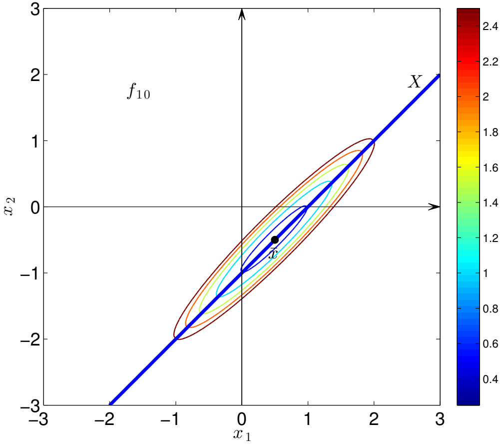
$f_{0.5}$
X
$\bar{x}$
x_1
x_2
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4

ETH zürich



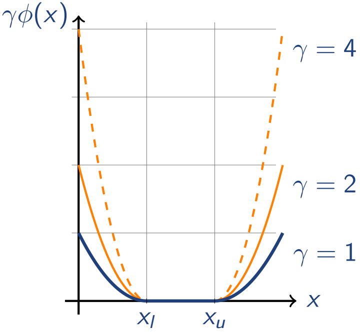


f_2
X
$\bar{x}$
x_1
x_2
3
2
1
0
−1
−2
−3
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4



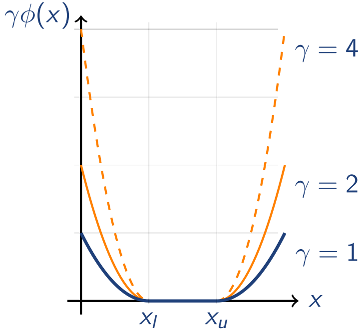
f_{10}
X
$\bar{x}$
x_1
x_2
3
2
1
0
−1
−2
−3
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4

Consider $X = \{x \in \mathbb{R} :\ x_l \leq x \leq x_u\}$



ill-conditioning for increasing γ

Consider $X = \{x \in \mathbb{R} : \; x_l \leq x \leq x_u\}$



ill-conditioning for increasing γ

Remedies:

- Continuation strategy for γ
- Augmented Lagrangian Method

$$F(x) = 0 \quad \text{with} \quad F : \mathbb{R}^n \to \mathbb{R}^n.$$

$$F(x) = 0 \quad \text{with} \quad F : \mathbb{R}^n \to \mathbb{R}^n.$$

Newton's Method for Equations

- Choose starting point x^0.

  For $k = 1, 2, 3, \ldots$

- Local approximation model $q(x)$

$$q(x) = F(x^k) + F'(x^k)(x - x^k).$$

- Find solution $\tilde{x} = x^k + s^k$ to $q(x) = 0$, i.e.

$$s^k = -F'(x^k)^{-1} F(x^k).$$

- Set $x^{k+1} = x^k + s^k$.

$$\nabla f(x) = 0 \quad \text{with} \quad f : \mathbb{R}^n \to \mathbb{R}.$$

Newton's Method for Optimization

- Choose starting point x^0.

  For $k = 1, 2, 3, \dots$

- Local approximation model $q(x)$

$$q(x) = \nabla f(x^k) + \nabla^2 f(x^k)(x - x^k)$$

- Find solution $\tilde{x} = x^k + s^k$ to $q(x) = 0$, i.e.

$$s^k = -\nabla^2 f(x^k)^{-1} \nabla f(x^k).$$

- Set $x^{k+1} = x^k + s^k$.

$$s^k = -\nabla^2 f(x^k)^{-1} \nabla f(x^k).$$

Problems:

- $\nabla^2 f(x^k)$ can be too expensive to compute,
- $\nabla^2 f(x^k)^{-1}$ is even more expensive!

$$s^k = -\nabla^2 f(x^k)^{-1} \nabla f(x^k).$$

Problems:

- $\nabla^2 f(x^k)$ can be too expensive to compute,
- $\nabla^2 f(x^k)^{-1}$ is even more expensive!

Idea:

Replace $\nabla^2 f(x^k)^{-1}$ by some approximation $B_k \approx \nabla^2 f(x^k)^{-1}$.

$$s^k = -\nabla^2 f(x^k)^{-1} \nabla f(x^k).$$

Problems:

- $\nabla^2 f(x^k)$ can be too expensive to compute,
- $\nabla^2 f(x^k)^{-1}$ is even more expensive!

Idea:

Replace $\nabla^2 f(x^k)^{-1}$ by some approximation $B_k \approx \nabla^2 f(x^k)^{-1}$.

Variants:

- $B_k = I$: steepest descent method
- BFGS, L-BFGS, L-BFGS-B

$\rightarrow$ require only matrix-vector operations.

Idea: Apply Newton's method to KKT conditions

$$\begin{aligned} \nabla f(\bar{x}) + \nabla h(\bar{x})\bar{\mu} &= 0, \\ h(\bar{x}) &= 0. \end{aligned}$$

Idea: Apply Newton's method to KKT conditions

$$\begin{aligned}\nabla f(\bar{x}) + \nabla h(\bar{x})\bar{\mu} &= 0,\\ h(\bar{x}) &= 0.\end{aligned}$$

- This is a nonlinear system of equations in (x, μ).

Idea: Apply Newton's method to KKT conditions

$$\begin{aligned} \nabla f(\bar{x}) + \nabla h(\bar{x})\bar{\mu} &= 0, \\ h(\bar{x}) &= 0. \end{aligned}$$

- This is a nonlinear system of equations in (x, μ).
- Newton step:

$$\begin{pmatrix} \nabla^2_{xx} L(x^k, \mu^k) & \nabla h(x^k) \\ \nabla h(x^k)^T & 0 \end{pmatrix} \begin{pmatrix} s_x^k \\ s_\mu^k \end{pmatrix} = -\begin{pmatrix} \nabla_x L(x^k, \mu^k) \\ h(x^k) \end{pmatrix}.$$

Idea: Apply Newton's method to KKT conditions

$$\begin{aligned} \nabla f(\bar{x}) + \nabla h(\bar{x})\bar{\mu} &= 0, \\ h(\bar{x}) &= 0. \end{aligned}$$

- This is a nonlinear system of equations in (x, μ).
- Newton step:

$$\begin{pmatrix} \nabla^2_{xx} L(x^k, \mu^k) & \nabla h(x^k) \\ \nabla h(x^k)^T & 0 \end{pmatrix} \begin{pmatrix} s_x^k \\ s_\mu^k \end{pmatrix} = -\begin{pmatrix} \nabla_x L(x^k, \mu^k) \\ h(x^k) \end{pmatrix}.$$

- "Infeasible" algorithm since $h(x^k) = 0$ can be violated.
- Fast local convergence if 2nd order sufficient conditions hold.

$$\begin{pmatrix} \nabla^2_{xx} L(x^k, \mu^k) & \nabla h(x^k) \\ \nabla h(x^k)^T & 0 \end{pmatrix} \begin{pmatrix} s^k_x \\ s^k_\mu \end{pmatrix} = -\begin{pmatrix} \nabla_x L(x^k, \mu^k) \\ h(x^k) \end{pmatrix}.$$

These are the KKT conditions for quadratic program (QP)

$$\begin{aligned} \min_{s \in \mathbb{R}^n} \quad & f(x^k) + \nabla f(x^k)^T s + \frac{1}{2} s^T \nabla^2_{xx} L(x^k, \mu^k) s \\ \text{s.t.} \quad & h(x^k) + \nabla h(x^k)^T s = 0. \end{aligned}$$

$$\begin{pmatrix} \nabla^2_{xx} L(x^k, \mu^k) & \nabla h(x^k) \\ \nabla h(x^k)^T & 0 \end{pmatrix} \begin{pmatrix} s_x^k \\ s_\mu^k \end{pmatrix} = -\begin{pmatrix} \nabla_x L(x^k, \mu^k) \\ h(x^k) \end{pmatrix}.$$

These are the KKT conditions for quadratic program (QP)

$$\begin{aligned} \min_{s \in \mathbb{R}^n} \quad & f(x^k) + \nabla f(x^k)^T s + \frac{1}{2} s^T \nabla^2_{xx} L(x^k, \mu^k) s \\ \text{s.t.} \quad & h(x^k) + \nabla h(x^k)^T s = 0. \end{aligned}$$

- Alternative: Solve (QP) to obtain update $(\bar{s}, \mu^k + \bar{\mu})$.
- QP can be interpreted as “local model”.
- $\nabla^2_{xx} L(x^k, \mu^k)$ can be replaced by an approximation H_k.

$$\begin{aligned}\nabla f(\bar{x}) + \nabla g(\bar{x})\bar{\lambda} + \nabla h(\bar{x})\bar{\mu} &= 0,\\ h(\bar{x}) &= 0,\\ g(\bar{x}) \leq 0, \quad \bar{\lambda} \geq 0, \quad \bar{\lambda}^T g(\bar{x}) &= 0.\end{aligned}$$

- Cannot directly apply Newton's method because of the inequalities.

$$\begin{aligned} \nabla f(\bar{x}) + \nabla g(\bar{x})\bar{\lambda} + \nabla h(\bar{x})\bar{\mu} &= 0, \\ h(\bar{x}) &= 0, \\ g(\bar{x}) \leq 0, \quad \bar{\lambda} \geq 0, \quad \bar{\lambda}^T g(\bar{x}) &= 0. \end{aligned}$$

- Cannot directly apply Newton's method because of the inequalities.
- Repeat QP idea:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) & & \min_{s \in \mathbb{R}^n} \quad f(x^k) + \nabla f(x^k)^T s + \frac{1}{2} s^T H_k s \\ \text{s.t.} \quad & h(x) = 0, & \rightarrow \quad & \text{s.t.} \quad h(x^k) + \nabla h(x^k)^T s = 0, \\ & g(x) \leq 0. & & \qquad g(x^k) + \nabla g(x^k)^T s \leq 0. \end{aligned}$$

$\rightarrow$ quadratic objective function, linear constraints

- Iteratively solve quadratic program to approach KKT-point
- QP "identifies the correct active constraints" for large k, i.e.,

$$g_i(\bar{x}) = 0 \quad \Leftrightarrow \quad g_i(x^k) = 0.$$

if $\bar{x}$ satisfies sufficient 2nd order conditions and $x^k \to \bar{x}$.

Crucial question: Can we solve (QP)?

- Iteratively solve quadratic program to approach KKT-point
- QP "identifies the correct active constraints" for large k, i.e.,

$$g_i(\bar{x}) = 0 \quad \Leftrightarrow \quad g_i(x^k) = 0.$$

if $\bar{x}$ satisfies sufficient 2nd order conditions and $x^k \to \bar{x}$.

Crucial question: Can we solve (QP)? Yes, but ...

- requires advanced techniques to handle non-convexity, globalization, second-order correction steps, ...
- Some codes: DONLP2, FilterSQP, Gurobi, SNOPT, ...

Reformulation of the problem:

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad g(x) \leq 0, \quad h(x) = 0.$$

Eliminate inequality constraints by slack variables:

$$g_i(x) \leq 0 \quad \Leftrightarrow \quad g_i(x) + s_i = 0 \quad \wedge \quad s_i \geq 0.$$

W.l.o.g. we consider

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad h(x) = 0, \quad x \geq 0.$$

Idea:

Add a barrier to the objective that turns to ∞ towards the boundaries of X.

Idea:

Add a barrier to the objective that turns to ∞ towards the boundaries of X.

Barrier Problem

For a fixed $\gamma \in (0, \infty)$ define

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f_\gamma(x) := f(x) - \gamma \sum_{i=1}^{m} \ln(x_i) \\ \text{s.t.} \quad & h(x) = 0. \end{aligned} \qquad (P_\gamma^B)$$

Idea:

Add a barrier to the objective that turns to ∞ towards the boundaries of X.

Barrier Problem

For a fixed $\gamma \in (0, \infty)$ define

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f_\gamma(x) := f(x) - \gamma \sum_{i=1}^{m} \ln(x_i) \\ \text{s.t.} \quad & h(x) = 0. \end{aligned} \qquad (P_\gamma^B)$$

Thus, we obtain an equality constrained optimization problem (P_γ^B).

ETH zürich

Algorithm

- Choose $\gamma_0 > 0$.

  For $k = 1, 2, 3, \dots$
- Solve $(P^B_{\gamma_k})$ approximately and obtain x^k.
- Choose $\gamma_{k+1} \in (0, \gamma_k)$.

Barrier Method

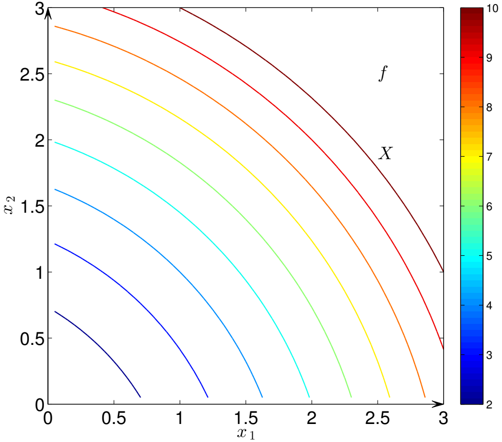
Algorithm

- Choose $\gamma_0 > 0$.

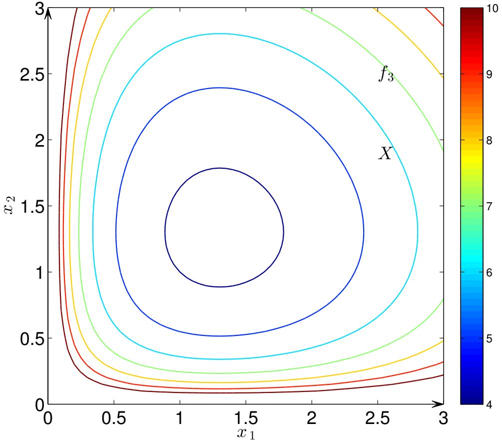
  For $k = 1, 2, 3, \ldots$
- Solve $(P^B_{\gamma_k})$ approximately and obtain x^k.
- Choose $\gamma_{k+1} \in (0, \gamma_k)$.

Remarks:

- x^k can be used as initial point for $(P^B_{\gamma_{k+1}})$.
- Methods for equality constrained problems can be used.
- Fast local convergence for fixed γ.
- Stopping criterion?

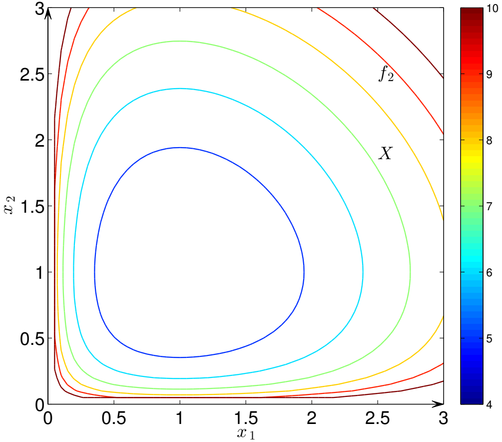


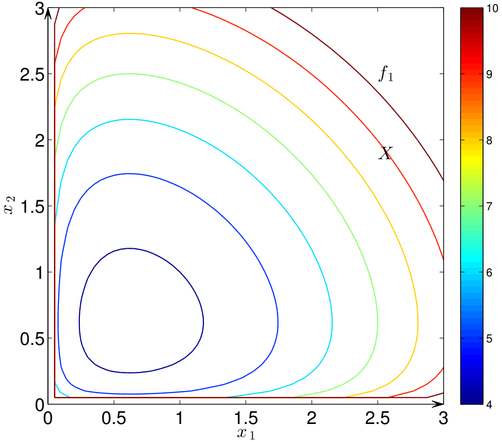
f
X
x_1
x_2
0
0.5
1
1.5
2
2.5
3
2
3
4
5
6
7
8
9
10

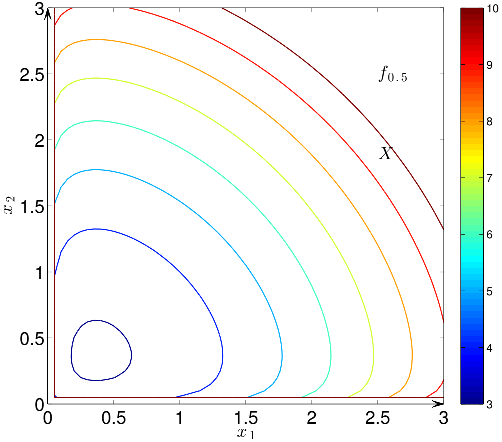


3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
x_1
x_2
f_3
X
10
9
8
7
6
5
4

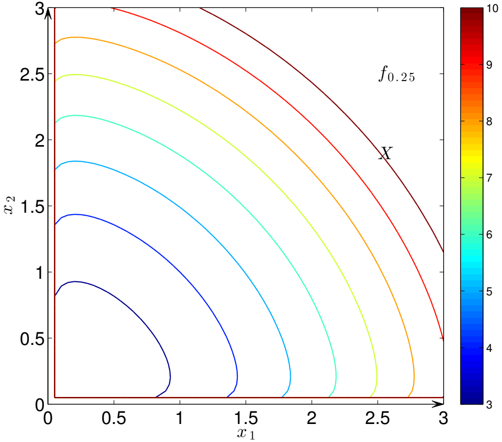
Barrier Method - Example




f_1
X
x_1
x_2
0
0.5
1
1.5
2
2.5
3
4
5
6
7
8
9
10

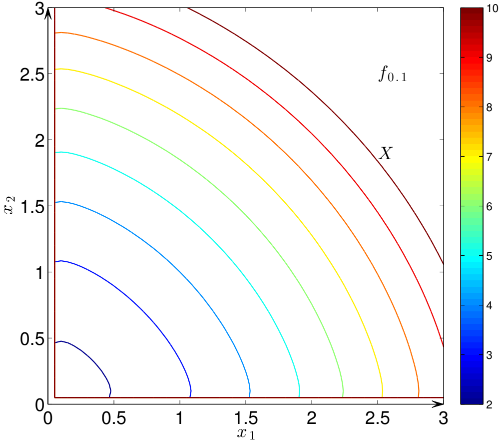


$f_{0.5}$
X
x_1
x_2
0
0.5
1
1.5
2
2.5
3
10
9
8
7
6
5
4



3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
x_1
x_2
$f_{0.25}$
X
10
9
8
7
6
5
4
3

Barrier Method - Example



Optimality conditions of (P_γ^B) can be rewritten as perturbed KKT system

$$\begin{aligned} \nabla f(\bar{x}) + \nabla h(\bar{x})\bar{\mu} + \lambda &= 0, \\ h(\bar{x}) &= 0, \\ x_i \lambda_i &= \gamma, \quad i = 1, \dots, n. \end{aligned}$$

- Rich theory on computational complexity, updating rules for γ and required accuracy to solve (P_γ^B).
- Interior Point Methods are among the most efficient methods for linear and nonlinear optimization methods.

Software: IPOPT

- Interior Point Optimizer
- Open source software for non-convex NLP
- Available from `https://projects.coin-or.org/Ipopt`
- Various interfaces (C++, Fortran, Python, Matlab)

- Interior Point Optimizer
- Open source software for non-convex NLP
- Available from https://projects.coin-or.org/Ipopt
- Various interfaces (C++, Fortran, Python, Matlab)

Features:

- Primal-dual interior point method based on barrier subproblems
- Filter globalization
- Infeasibility restoration

Minimal Requirements:

- Specify problem dimensions and initial point.
- `function f_value = eval_f(x)`
- `function grad_f = eval_grad_f(x)`
- `function constraints_value = eval_c(x)`
- `function grad_constraints = eval_grad_c(x)`

Minimal Requirements:

- Specify problem dimensions and initial point.
- `function f_value = eval_f(x)`
- `function grad_f = eval_grad_f(x)`
- `function constraints_value = eval_c(x)`
- `function grad_constraints = eval_grad_c(x)`

Optional:

- Second derivatives
- Many parameters to tune performance

We discussed:

- First-order necessary optimality conditions for constrained problems
- Iterative algorithms that converge to KKT-points.

Choice of optimization method depends on problem characteristics:

- What is more expensive: objective function or constraints?
- Are many constraints (expected to be) active at the minimum?
- Can f be computed for infeasible x?
- Can P_X be computed efficiently?

- Books on nonlinear optimization:
  - J. Nocedal and S. J. Wright: Numerical Optimization (2nd edition), Springer 2006
  - C. T. Kelley: Iterative Methods for Optimization. SIAM 1999
  - M. Hinze, R. Pinnau, S. Ulbrich, and M. Ulbrich: Optimization with PDE Constraints. Springer 2009

- Websites with optimization codes
  - Decision Tree of Optimization Software: http://plato.la.asu.edu/guide.html
  - NEOS Guide: http://www.neos-guide.org/Optimization-Guide