Università
della
Svizzera
italiana

Faculty
of Informatics

Institute of
Computational
Science
ICS

# A4: Quadratic Programming

solving fundamental non-linear optimization problem

CECAM workshop, Mainz, 2019

**Illia Horenko, <u>Lukas Pospisil</u>**

# Outline

1.) Motivation
2.) Fundamental concepts
3.) QP without constraints
4.) QP with constraints


- *Nocedal J, Wright S.J:* Numerical Optimization. *(Springer, New York), 2nd edition, 2006*
- *Boyd S, Vandenberghe L.:* Convex Optimization. *Cambridge University Press, New York, 1st edition, 2004.*

- *Pospíšil L.:* Development of Algorithms for Solving Minimizing Problems with Convex Quadratic Function on Special Convex Sets and Applications, *PhD thesis, VSB-TU Ostrava, supervised by Z. Dostál, 2015*
- *Dostál Z.:* Optimal Quadratic Programming Algorithms, with Applications to Variational Inequalities, *SOIA, first ed., vol. 23, Springer, US, New York, 2009.*

# 1.) Motivation

# What is QP and basic properties

QP:

$$x^* := \arg\min_{x \in \Omega} f(x), \quad f(x) := \frac{1}{2} x^T A x - b^T x, \quad f : \mathbb{R}^n \to \mathbb{R}, A = A^T, A > 0$$

Matlab quadprog: $f(x) := 0.5 * x' * H * x + f' * x$

- Solution of Ax=b with SPD A, solution of Ax=b in LS sense

- Polynomial regression, modelling with linear parametric models
- Non-stationary time-series modelling with H1 regularization (FEM-H1, SPA-H1)

- (Euclidean) projection onto closed convex set is QP

- Linear elasticity, linear elasticity contact problems
- Rigid body motion, granular dynamics

- Subproblems in general optimization (Taylor expansion of second degree = QP)

# What is QP and basic properties

QP:

$$x^* := \arg\min_{x\in\Omega} f(x), \quad f(x) := \frac{1}{2}x^T A x - b^T x, \quad f:\mathbb{R}^n \to \mathbb{R}, A = A^T, A > 0$$

Lemma:

$$\forall x, d \in \mathbb{R}^n \ \forall \alpha \in \mathbb{R}: \quad f(x+\alpha d) = f(x) + \alpha\langle Ax - b, d\rangle + \frac{1}{2}\alpha^2\langle Ad, d\rangle$$

Corollary:

$$\alpha := 1, y := x + d: \quad f(y) = f(x) + (Ax - b)^T(y - x) + \frac{1}{2}(y - x)^T A(y - x)$$

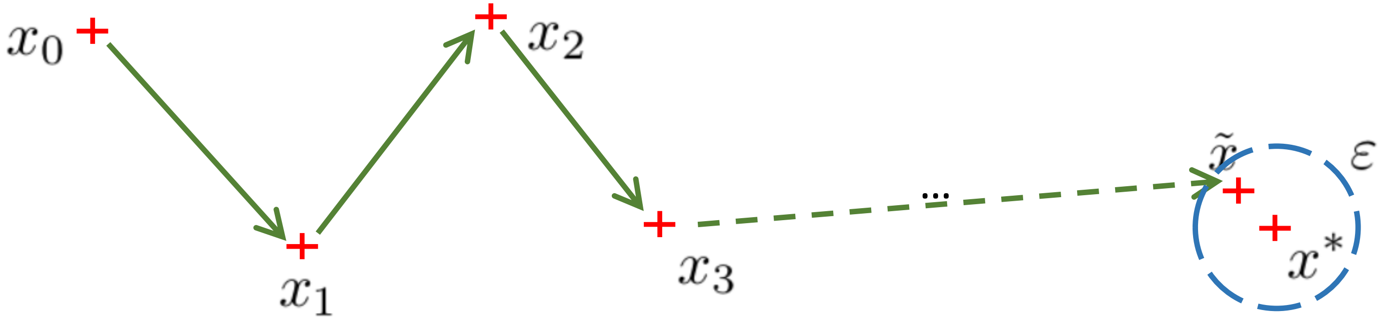
Taylor: $f(x) = f(a) + (\nabla f(a))^T(x - a) + \frac{1}{2}(x - a)^T \nabla^2 f(a)(x - a) + \cancel{o(\|x - a\|^2)}$

$$\Rightarrow \quad \nabla f(x) = Ax - b, \ \nabla^2 f(x) = A$$

# 3.) Iterative solvers for unconstrained QP

# Iterative solvers

$$x_0 \in \mathbb{R}^n, \quad x_{k+1} = x_k + \alpha_k d_k$$



# Iterative solvers

$$x_0 \in \mathbb{R}^n, \quad x_{k+1} = x_k + \alpha_k d_k$$

Gradient descent methods $\quad d_k := -\nabla f(x_k)$

# Iterative solvers

$$x_0 \in \mathbb{R}^n, \quad x_{k+1} = x_k + \alpha_k d_k$$

Gradient descent methods $\quad d_k := -\nabla f(x_k)$

$$\boxed{\alpha_k := \alpha \text{ const.}}$$

Constant step-size

$$x_{k+1} = \underbrace{x_k - \alpha \nabla f(x_k)}_{=: F(x_k)}$$

Using Banach theorem:

$$\forall x, y \in \mathbb{R}^n : \ \|F(x) - F(y)\| \leq \underbrace{\|I - \alpha A\|}_{<1} \cdot \|x - y\|$$

$$< 1 \quad \Leftrightarrow \quad \max_i \{|1 - \alpha\lambda_i|\} < 1$$

$$\Leftrightarrow \forall i : \ \lambda_i \alpha \in (0, 2)$$

$$\Leftrightarrow \alpha < \frac{2}{\lambda_{\max}}$$

# Iterative solvers

$$x_0 \in \mathbb{R}^n, \quad x_{k+1} = x_k + \alpha_k d_k$$

Gradient descent methods $\quad d_k := -\nabla f(x_k)$

Cauchy step-size

$$\boxed{\alpha_k := \arg\min_{\alpha} \underbrace{f(x_k + \alpha d_k)}_{=:\, \varphi(\alpha)}}$$

$$\varphi(\alpha) = f(x_k + \alpha d_k) = f(x_k) + \alpha \langle A x_k - b, d_k \rangle + \frac{1}{2}\alpha^2 \langle A d_k, d_k \rangle$$

$$\varphi'(\alpha) = \langle A x_k - b, d_k \rangle + \alpha \langle A d_k, d_k \rangle$$

$$\varphi'(\alpha) = 0 \quad \Leftrightarrow \quad \alpha_k = -\frac{\langle A x_k - b, d_k \rangle}{\langle A d_k, d_k \rangle}$$

$$d_k := -\nabla f(x_k) =: -g_k \quad \Rightarrow \quad \boxed{\alpha_k = \frac{\langle g_k, g_k \rangle}{\langle A g_k, g_k \rangle}}$$

Steepest descent method

$$\boxed{\|e_{k+1}\|_A \le \sqrt{1 - \frac{1}{\kappa(A)}}\, \|e_k\|_A}$$

$$e_k := x_k - x^*, \quad \|x\|_A := \sqrt{\langle A x, x \rangle}$$

# Iterative solvers

$$x_0 \in \mathbb{R}^n, \quad x_{k+1} = x_k + \alpha_k d_k$$

Krylov subspace methods

$$\boxed{\begin{aligned} &d_k \text{ is } A\text{-orthogonal to } \{d_0, \ldots, d_{k-1}\}, d_0 := -\nabla f(x_0) \\ &\alpha_k := \arg\min_{\alpha} f(x_k + \alpha d_k) \end{aligned}}$$

$$\alpha_k = -\frac{\langle g_k, d_k \rangle}{\langle A d_k, d_k \rangle}$$

- Generated using modified Gram-Schmidt orthogonalization process
- Feature: orthogonalize only with respect to last step direction (wow)

- A-orthogonal -> linearly independent -> basis of Krylov subspace
- Algorithm generates coordinates of initial error in the A-orthogonal basis

$$\mathcal{K}_k(\mathbf{A}, b) \equiv \operatorname{span}\{b, \mathbf{A}b, \ldots, \mathbf{A}^{k-1}b\}$$

$$\|x - x_k\|_{\mathbf{A}} = \min_{y \in \mathcal{K}_k} \|x - y\|_{\mathbf{A}}$$

Conjugate gradient method

$$\boxed{\|e_{k+1}\|_A \leq 2 \left( \frac{\kappa(A) - 1}{\kappa(A) + 1} \right)^k \|e_0\|_A}$$

# 4.) Iterative solvers for constrained QP

# KKT in general

$$x^* = \arg\min_{x\in\Omega} \frac{1}{2}x^T Ax - b^T x$$

$$\begin{aligned} \Omega &= \Omega_E \cap \Omega_I \\ \Omega_E &= \{x \in \mathbb{R}^n : Bx = c\} \\ \Omega_I &= \{x \in \mathbb{R}^n : x \geq 0\} \end{aligned}$$

# KKT in general

$$\begin{aligned}\Omega &= \Omega_E \cap \Omega_I \\ \Omega_E &= \{x \in \mathbb{R}^n : Bx = c\} \\ \Omega_I &= \{x \in \mathbb{R}^n : x \geq 0\}\end{aligned}$$

$$x^* = \arg\min_{x \in \Omega} \frac{1}{2} x^T A x - b^T x$$

$$L(x, \lambda_E, \lambda_I) = \frac{1}{2} x^T A x - b^T x + \lambda_E^T (Bx - c) - \lambda_I^T x$$

$$\begin{array}{rclcl}\nabla_x L &=& Ax - b + B^T \lambda_E - \lambda_I &=& 0 \\ \nabla_{\lambda_E} L &=& Bx - c &=& 0 \\ \nabla_{\lambda_I} L &=& -x &\leq& 0 \\ && \lambda_I &\geq& 0 \\ \forall i : && [\lambda_I]_i x_i &=& 0\end{array}$$

# KKT for equalities

$$\Omega = \Omega_E \cancel{\cap \Omega_I}$$
$$\Omega_E = \{x \in \mathbb{R}^n : Bx = c\}$$
$$\cancel{\Omega_I = \{x \in \mathbb{R}^n : x \geq 0\}}$$

$$x^* = \arg\min_{x\in\Omega} \frac{1}{2}x^T A x - b^T x$$

$$L(x, \lambda_E, \cancel{\lambda_I}) = \frac{1}{2}x^T A x - b^T x + \lambda_E^T(Bx - c) \cancel{- \lambda_I^T x}$$

$$\begin{array}{rcrcl}
\nabla_x L & = & Ax - b + B^T\lambda_E \cancel{- \lambda_I} & = & 0 \\
\nabla_{\lambda_E} L & = & Bx - c & = & 0 \\
\cancel{\nabla_{\lambda_I} L} & \cancel{=} & \cancel{-x} & \cancel{\leq} & \cancel{0} \\
 & & \cancel{\lambda_I} & \cancel{\geq} & \cancel{0} \\
\cancel{\forall i :} & & \cancel{[\lambda_I]_i x_i} & \cancel{=} & \cancel{0}
\end{array}$$

Saddle-point problem

$$\begin{array}{rcl}
Ax - b + B^T\lambda & = & 0 \\
Bx - c & = & 0
\end{array}$$

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix}$$

quite naïve ☹, actually:

$$x = A^+(b - B^T\lambda) + R\alpha, \alpha \in \mathbb{R}^{\dim \operatorname{Ker} A}$$

$$\text{dual problem} \quad \lambda = \arg\min \frac{1}{2}\lambda^T B A^{-1} B^T \lambda - (BA^{-1}b - c)^T \lambda$$

# 4b.) Penalty methods

(how to deal with equalities)

# Equality constrained QP

$$x^* = \arg\min_{Bx=c} \frac{1}{2}x^T Ax - b^T x$$

Penalty method:

$$x^* = \arg\min_{x\in\Omega} f(x) \quad \approx \quad x^* = \arg\min_{x\in\mathbb{R}^n} f(x) + \underbrace{\frac{\rho}{2}\|Bx - c\|^2}_{\text{Enforce the equality constraint using penalty term}}$$

~ Tikhonov regularization?

# Equality constrained QP

$$x^* = \arg \min_{Bx=c} \frac{1}{2} x^T A x - b^T x$$

Penalty method:

$$x^* = \arg \min_{x \in \Omega} f(x) \quad \approx \quad x^* = \arg \min_{x \in \mathbb{R}^n} f(x) + \frac{\rho}{2} \|Bx - c\|^2$$

*Let $A \in \mathbb{R}^{n,n}$ be SPD matrix, $B \in \mathbb{R}^{m,n}$ be nonzero matrix, and $b \in \mathbb{R}^n$. We assume that $B$ is not necessarily a full rank matrix. Let $\varepsilon \geq 0$ and $\rho > 0$. Let $\hat{x}$ is an approximate solution of unconstrained optimization problem*

$$\min_{x \in \mathbb{R}^n} f_\rho(x) \qquad f_\rho(x) := f(x) + \frac{\rho}{2} \|Bx\|^2 \tag{1.41}$$

*such that the necessary optimality condition $\nabla_x f_\rho(x) = 0$ is satisfied approximately with respect to relative precision $\varepsilon \|b\|$, i.e.*

$$\|\nabla_x f_\rho(x)\| \leq \varepsilon \|b\|,$$

*then*

$$\|Bx\| \leq \frac{1 + \varepsilon}{\sqrt{\lambda_{\min}^A \rho}} \|b\|.$$


- *Dostál Z.:* Optimal Quadratic Programming Algorithms, with Applications to Variational Inequalities, *SOIA, first ed., vol. 23, Springer, US, New York, (2009).*

# Equality constrained QP

$$x^* = \arg\min_{Bx=c} \frac{1}{2} x^T A x - b^T x$$

Penalty method:

$$x^* = \arg\min_{x \in \Omega} f(x) \quad \approx \quad x^* = \arg\min_{x \in \mathbb{R}^n} f(x) + \frac{\rho}{2} \|Bx - c\|^2$$

*Let* $A \in \mathbb{R}^{n \times n}$ *be a symmetric positive semidefinite matrix, let* $B \in \mathbb{R}^{m \times n}$, $\rho > 0$, *and let* $\operatorname{Ker} A \cap \operatorname{Ker} B = \{0\}$. *Then matrix*

$$A_\rho = A + \rho B^T B$$

*is symmetric positive definite and*

$$\kappa(A_\rho) \geq \hat{\kappa}(A) \; .$$

*Moreover,*

- *if* $\hat{\kappa}(B^T B) \leq \hat{\kappa}(A)$ *and*

$$\rho \in \left[ \frac{\hat{\lambda}^A_{\min}}{\hat{\lambda}^{B^T B}_{\min}}, \frac{\lambda^A_{\max}}{\lambda^{B^T B}_{\max}} \right]$$

  *then* $\kappa(A_\rho) = \hat{\kappa}(A)$,
- *if* $\hat{\kappa}(B^T B) > \hat{\kappa}(A)$ *then* $\forall \rho > 0 : \kappa(A_\rho) > \hat{\kappa}(A)$.


- *Pospíšil L.:* Development of Algorithms for Solving Minimizing Problems with Convex Quadratic Function on Special Convex Sets and Applications, *PhD thesis, VSB-TU Ostrava, supervised by Z. Dostál (2015)*

# Equality constrained QP

$$x^* = \arg\min_{Bx=c} \frac{1}{2} x^T A x - b^T x$$

It is not necessary to solve inner problem exactly in every iteration – adaptive accuracy

"Uzawa" update

Adaptive penalty update

**Algorithm 4.5. Semimonotonic augmented Lagrangians (SMALE).**

*Given a symmetric positive definite matrix* $\mathsf{A} \in \mathbb{R}^{n\times n}$, $\mathsf{B} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^n$, *and* $\mathbf{c} \in \mathrm{Im}\mathsf{B}$.

*Step 0.* {*Initialization.*}
*Choose* $\eta > 0,\ \beta > 1,\ M > 0,\ \varrho_0 > 0,\ \boldsymbol{\lambda}^0 \in \mathbb{R}^m$
**for** $k$=*0,1,2,...*

*Step 1.* {*Inner iteration with adaptive precision control.*}
*Find* $\mathbf{x}^k$ *such that*

$$\|\mathbf{g}(\mathbf{x}^k, \boldsymbol{\lambda}^k, \varrho_k)\| \le \min\{M\|\mathsf{B}\mathbf{x}^k - \mathbf{c}\|, \eta\}. \tag{4.80}$$

*Step 2.* {*Updating the Lagrange multipliers.*}

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \varrho_k(\mathsf{B}\mathbf{x}^k - \mathbf{c}) \tag{4.81}$$

*Step 3.* {*Update* $\varrho$ *provided the increase of the Lagrangian is not sufficient.*}
**if** $k > 0$ *and*

$$L(\mathbf{x}^k, \boldsymbol{\lambda}^k, \varrho_k) < L(\mathbf{x}^{k-1}, \boldsymbol{\lambda}^{k-1}, \varrho_{k-1}) + \frac{\varrho_k}{2}\|\mathsf{B}\mathbf{x}^k - \mathbf{c}\|^2 \tag{4.82}$$

$\varrho_{k+1} = \beta\varrho_k$
**else**
$\varrho_{k+1} = \varrho_k.$
**end if**
**end for**


- *Dostál Z.:* Optimal Quadratic Programming Algorithms, with Applications to Variational Inequalities, *SOIA, first ed., vol. 23, Springer, US, New York, (2009).*

# 4c.) Active-set methods

(how to deal with inequalities)

# KKT for inequalities

$$
\begin{aligned}
\Omega &= \Omega_E \cancel{\cap \Omega_I} \\
\cancel{\Omega_E} &\cancel{= \{x \in \mathbb{R}^n : Bx = c\}} \\
\Omega_I &= \{x \in \mathbb{R}^n : x \geq 0\}
\end{aligned}
$$

$$x^* = \arg\min_{x \in \Omega} \frac{1}{2} x^T A x - b^T x$$

$$L(x, \lambda_E, \cancel{\lambda_I}) = \frac{1}{2} x^T A x - b^T x \cancel{+ \lambda_E^T (Bx - c)} - \lambda_I^T x$$

$$
\begin{aligned}
\nabla_x L &= Ax - b + \cancel{B^T \lambda_E} - \lambda_I &= 0 \\
\cancel{\nabla_{\lambda_E} L} &\cancel{= Bx - c} &\cancel{= 0} \\
\nabla_{\lambda_I} L &= -x &\leq 0 \\
& \qquad \lambda_I &\geq 0 \\
\forall i : & \qquad [\lambda_I]_i x_i &= 0
\end{aligned}
$$

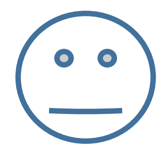
# KKT for inequalities

$$
\begin{aligned}
\Omega &= \Omega_E \cancel{\cap \Omega_I} \\
\cancel{\Omega_E} &\cancel{= \{x \in \mathbb{R}^n : Bx = c\}} \\
\Omega_I &= \{x \in \mathbb{R}^n : x \geq 0\}
\end{aligned}
$$

$$x^* = \arg\min_{x \in \Omega} \frac{1}{2} x^T A x - b^T x$$

$$L(x, \lambda_E, \cancel{\lambda_I}) = \frac{1}{2} x^T A x - b^T x \cancel{+ \lambda_E^T (Bx - c)} - \lambda_I^T x$$

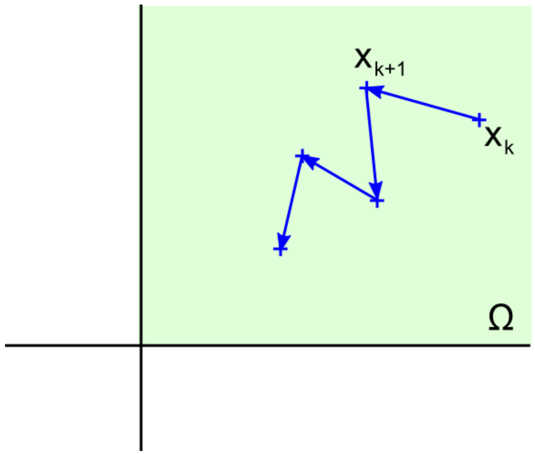
$$
\begin{aligned}
\nabla_x L &= Ax - b + \cancel{B^T \lambda_E} - \lambda_I &&= 0 \\
\cancel{\nabla_{\lambda_E} L} &\cancel{= \quad Bx - c} &&\cancel{= 0} \\
\nabla_{\lambda_I} L &= -x &&\leq 0 \\
& \lambda_I &&\geq 0 \\
\forall i : \quad & [\lambda_I]_i x_i &&= 0
\end{aligned}
$$



“try” to set some of $x_i$ equal to 0

-> active/free constraint -> <u>Active set methods</u>

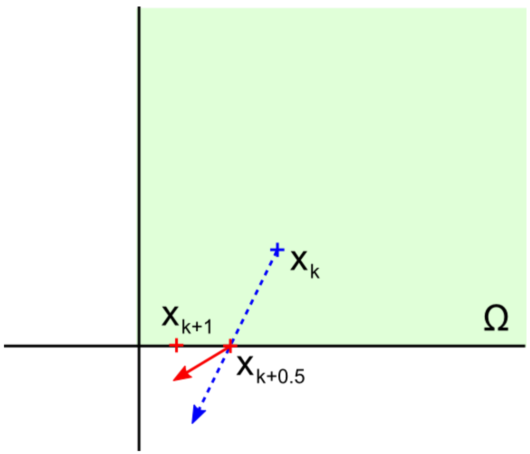
# Modified Proportioning with Gradient projection (MPGP)




CG-step

$$\alpha_{cg} = g_k^T p_k / p_k^T A p_k$$
$$x_{k+1} = x_k - \alpha_{cg} p_k$$

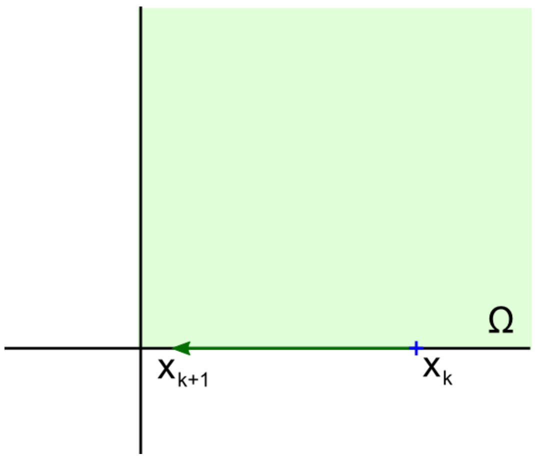
$$\gamma = \varphi\left(x_{k+1}\right)^T A p_k / p_k^T A p_k$$
$$p_{k+1} = \varphi\left(x_{k+1}\right) - \gamma p_k$$




CG-halfstep
+ expansion step

$$\alpha_f = \max\left\{\alpha \in \mathbb{R}^+ :\ x^k - \alpha p \in \Omega\right\}$$
$$x_{k+0.5} = x_k - \alpha_f p_k$$
$$x_{k+1} = P_\Omega(x_{k+0.5} - \bar{\alpha} g_{k+0.5})$$

$$\bar{\alpha} \in (0, 2/\|A\|\rangle$$




optimal step on Active set

$$d = \beta(x_k)$$
$$\alpha_{SD} = g_k^T d_k / d_k^T A d_k$$
$$x_{k+1} = x_k - \alpha_{SD} d_k$$

- *Dostál Z., Pospíšil L.:* Minimization of the quadratic function with semidefinite Hessian subject to the bound constraints, *Computers and Mathematics with Applications, (2015).*

# 4d.) Interior point methods

(how to deal with inequalities approximately)

# Barrier function method

$$x^* = \arg\min_{x \geq 0} f(x) \quad \approx \quad x^* = \arg\min_{x \in \mathbb{R}^n} f(x) - \mu \sum_{i=1}^{n} \log(x_i) \qquad (\ x > 0\ )$$

# Barrier function method

$$x^* = \arg\min_{x \geq 0} f(x) \quad \approx \quad x^* = \arg\min_{x \in \mathbb{R}^n} f(x) - \mu \sum_{i=1}^{n} \log(x_i) \qquad (\, x > 0 \,)$$

First-order optimality condition:

$$Ax - b - \mu \begin{bmatrix} 1/x_1 \\ \vdots \\ 1/x_n \end{bmatrix} = 0$$

# Barrier function method

$$x^* = \arg\min_{x \geq 0} f(x) \quad \approx \quad x^* = \arg\min_{x \in \mathbb{R}^n} f(x) - \mu \sum_{i=1}^{n} \log(x_i) \qquad (\ x > 0\ )$$

First-order optimality condition:

$$Ax - b - \mu \begin{bmatrix} 1/x_1 \\ \vdots \\ 1/x_n \end{bmatrix} = 0$$

$$\begin{aligned} Ax - b - \lambda &= 0 \\ x &> 0 \\ \lambda &> 0 \\ \forall i: \quad \lambda_i x_i &= \mu \end{aligned}$$

Introduce notation:

$$\lambda \in \mathbb{R}^n : \quad \lambda_i := \frac{\mu}{x_i}$$

# Barrier function method

$$x^* = \arg\min_{x \geq 0} f(x) \quad \approx \quad x^* = \arg\min_{x \in \mathbb{R}^n} f(x) - \mu \sum_{i=1}^{n} \log(x_i) \qquad (\; x > 0 \;)$$

First-order optimality condition:

$$Ax - b - \mu \begin{bmatrix} 1/x_1 \\ \vdots \\ 1/x_n \end{bmatrix} = 0$$

Introduce notation:

$$\lambda \in \mathbb{R}^n : \qquad \lambda_i := \frac{\mu}{x_i}$$

$$\begin{array}{rcl} Ax - b - \lambda & = & 0 \\ x & > & 0 \\ \lambda & > & 0 \\ \forall i : \quad \lambda_i x_i & = & \mu \end{array} \quad \approx \quad \begin{array}{rcrcl} \nabla_x L & = & Ax - b + \cancel{B^T \lambda_E} - \lambda_I & = & 0 \\ \cancel{\nabla_{\lambda_E} L} & \cancel{=} & \cancel{Bx - c} & \cancel{=} & \cancel{0} \\ \nabla_{\lambda_I} L & = & -x & \leq & 0 \\ & & \lambda_I & \geq & 0 \\ \forall i : & & [\lambda_I]_i x_i & = & 0 \end{array}$$

Key idea: Construct a sequence of subproblems where $\mu \to 0$

# Barrier function method

$$x^* = \arg\min_{x \geq 0} f(x) \quad \approx \quad x^* = \arg\min_{x \in \mathbb{R}^n} f(x) - \mu \sum_{i=1}^{n} \log(x_i) \qquad (\, x > 0 \,)$$

First-order optimality condition:

$$Ax - b - \mu \begin{bmatrix} 1/x_1 \\ \vdots \\ 1/x_n \end{bmatrix} = 0$$

$$\begin{array}{rcl} Ax - b - \lambda & = & 0 \\ x & > & 0 \\ \lambda & > & 0 \\ \forall i : \quad \lambda_i x_i & = & \mu \end{array}$$

Introduce notation:

$$\lambda \in \mathbb{R}^n : \quad \lambda_i := \frac{\mu}{x_i}$$

Solve system of non-linear eqs:

$$\begin{array}{rcl} Ax - b - \lambda & = & 0 \\ \forall i : \quad \lambda_i x_i & = & \mu \end{array}$$

such that $x, \lambda > 0$

using Newton method

with controlled step-size

(for example predictor-corrector method)

- *Stephen J. Wright: Primal-Dual Interior-Points Methods, SIAM, 1997.*

# 4a.) Projected gradient descent methods

(when the projection is known)

# Projected gradient descent methods

$$x^* = \arg\min_{x \in \Omega} \frac{1}{2} x^T A x - b^T x$$

$$x_0 \in \mathbb{R}^n, \quad x_{k+1} = x_k + \alpha_k d_k$$ (unconstrained)

$$\big\downarrow$$

$$x_0 \in \Omega, \quad x_{k+1} = P(x_k - \alpha_k \nabla f(x_k))$$ (constrained)

$$P(x) = \arg\min_{y \in \Omega} \|x - y\|$$

# Projected gradient descent methods

$$x^* = \arg\min_{x\in\Omega} \frac{1}{2} x^T A x - b^T x$$

$$x_0 \in \mathbb{R}^n, \quad x_{k+1} = x_k + \alpha_k d_k$$ (unconstrained)

$$\cancel{x_0 \in \Omega, \quad x_{k+1} = P(x_k - \alpha_k \nabla f(x_k))}$$ (constrained)

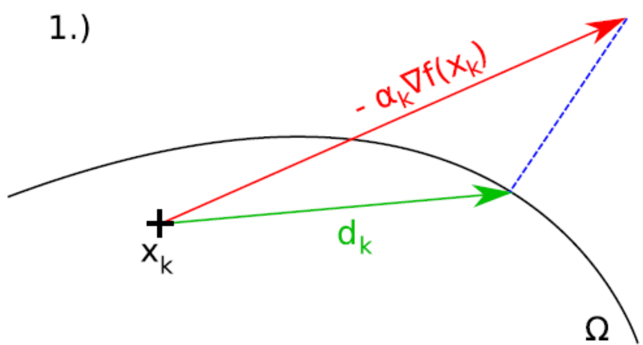
$$P(x) = \arg\min_{y\in\Omega} \|x - y\|$$

$$\boxed{\begin{aligned} & x_0 \in \Omega \\ & d_k = P_\Omega(x_k - \alpha_k \nabla f(x_k)) - x_k \\ & x_{k+1} = x_k + \beta_k d_k \end{aligned}}$$

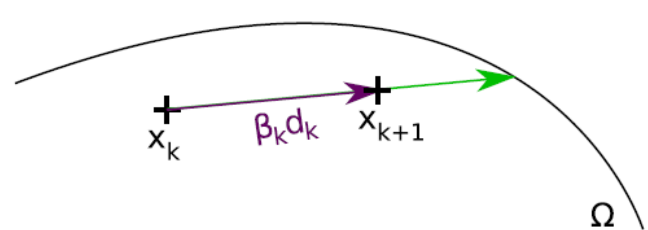
"Projected gradient"

To obtain the decrease of objective function
Using "line-search" techniques

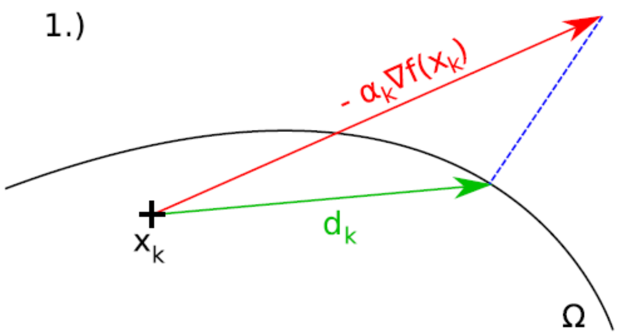
1.)




2.)



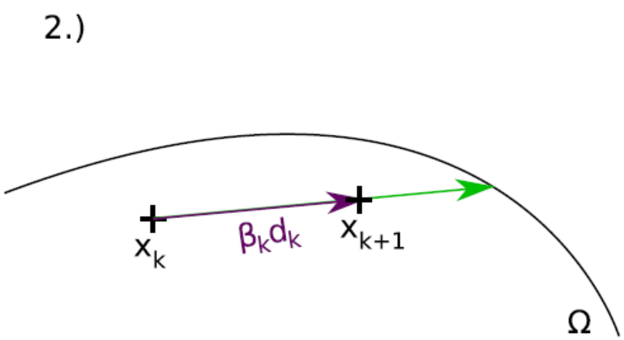
# Example: Spectral Projected Gradient method (SPG)




$$d^k = P(x^k - \alpha_k \nabla f(x^k)) - x^k$$

$$P(x) = \arg\min_{y \in \Omega} \|x - y\|$$

$\alpha_k$ - Barzilai-Borwein step-length




find $\beta_k \in (0, 1]$ such that

$$f(x^k + \beta_k d^k) \leq f_{\max} + \gamma \beta_k \langle \nabla f(x^k), d^k \rangle$$
$$f_{\max} = \max\{f(x^k), f(x^{k-1}), \ldots, f(x^{k-M})\}$$
$$x^{k+1} = x^k + \beta_k d^k$$


- *Birgin E.G., Raydan M., Martínez J.M.:* Spectral Projected Gradient Methods: Review and Perspectives, *2014*
- *Pospíšil L., Gagliardini P., Sawyer W., Horenko I.*: On a scalable nonparametric denoising of time series signals. *Commun. Appl. Math. Comput. Sci. 13 (2018), no. 1, 107--138. doi:10.2140/camcos.2018.13.107*
- *Pospíšil L.:* Development of Algorithms for Solving Minimizing Problems with Convex Quadratic Function on Special Convex Sets and Applications, *PhD thesis, supervised by Z. Dostál (2015)*

# Which QP algorithm should we use?

- first try **quadprog** in Matlab (trust-region / interior-point) for small, dense problems (if it is working & sufficiently fast -> do not touch it :)

- solve KKT or formulate dual problem, use the structure of the problem (e.g., sparsity), analyze properties of the problem (e.g., regularity, condition number)

- inequalities exactly -> active set methods

- inequalities approximately -> interior point methods

- if projection is known & cheap (for example separable feasible set) -> projected gradient methods