Abs-Linearization for
Piecewise Smooth Optimization

Andrea Walther\textsuperscript{1} and Andreas Griewank\textsuperscript{2}

\textsuperscript{1}Institut für Mathematik, Universität Paderborn
\textsuperscript{2}School of Mathematical Sciences and Information Techno., Yachay Tech

125th NII Shonan Meeting

Shonan, June 25, 2018
Outline

1. Piecewise Smooth Problems and Their Properties
2. Optimization for PS functions
3. Abs-Linearisation
4. The SALOP Algorithm
5. Relation to Other Derivative Concepts
6. Conclusion and Outlook
Definition (Piecewise Smoothness, Piecewise Linearity)

Let $\mathcal{D} \subseteq \mathbb{R}^n$ be open and $f_i : \mathcal{D} \to \mathbb{R}^m$, $i = 1, \ldots, k$ with $k \in \mathbb{N}$ be given.

- $f : \mathcal{D} \to \mathbb{R}^m$ is called a continuous selection of the collection $f_1, \ldots, f_k$ on the set $U \subseteq \mathcal{D}$ if $f$ is continuous on $U$ and $f(x) \in \{f_1(x), \ldots, f_k(x)\}$ for all $x \in U$. 

- A $PC_r$-function with $r \in \mathbb{N} \cup \{\infty\}$ if for every $x \in \mathcal{D}$ there exists an open neighborhood $U \subseteq \mathcal{D}$ and a finite number of $C^r$-functions $f_i : U \to \mathbb{R}^m$, $i = 1, \ldots, k$, such that $f$ is a continuous selection of $f_1, \ldots, f_k$ on $U$.

A $PC_r$-function with $r \geq 1$ is also called piecewise smooth.

A continuous selection $f : U \to \mathbb{R}^m$ is called piecewise linear if all elements of the collection $f_1, \ldots, f_k$ are affine functions.
Definition (Piecewise Smoothness, Piecewise Linearity)

Let $\mathcal{D} \subseteq \mathbb{R}^n$ be open and $f_i : \mathcal{D} \to \mathbb{R}^m$, $i = 1, \ldots, k$ with $k \in \mathbb{N}$ be given.

- $f : \mathcal{D} \to \mathbb{R}^m$ is called a continuous selection of the collection $f_1, \ldots, f_k$ on the set $U \subseteq \mathcal{D}$ if $f$ is continuous on $U$ and $f(x) \in \{f_1(x), \ldots, f_k(x)\}$ for all $x \in U$.

- $f : \mathcal{D} \to \mathbb{R}^m$ is called $PC^r$-function with $r \in \mathbb{N} \cup \{\infty\}$ if for every $x \in \mathcal{D}$ there exists an open neighborhood $U \subseteq \mathcal{D}$ and a finite number of $C^r$-functions $f_i : U \to \mathbb{R}^m$, $i = 1, \ldots, k$, such that $f$ is a continuous selection of $f_1, \ldots, f_k$ on $U$. 
Piecewise Smooth (PS) Functions

Definition (Piecewise Smoothness, Piecewise Linearity)

Let $D \subseteq \mathbb{R}^n$ be open and $f_i : D \rightarrow \mathbb{R}^m$, $i = 1, \ldots, k$ with $k \in \mathbb{N}$ be given.

1. $f : D \rightarrow \mathbb{R}^m$ is called a continuous selection of the collection $f_1, \ldots, f_k$ on the set $U \subseteq D$ if $f$ is continuous on $U$ and $f(x) \in \{f_1(x), \ldots, f_k(x)\}$ $\forall x \in U$.

2. $f : D \rightarrow \mathbb{R}^m$ is called $P C^r$-function with $r \in \mathbb{N} \cup \{\infty\}$ if for every $x \in D$ there exists an open neighborhood $U \subseteq D$ and a finite number of $C^r$-functions $f_i : U \rightarrow \mathbb{R}^m$, $i = 1, \ldots, k$, such that $f$ is a continuous selection of $f_1, \ldots, f_k$ on $U$.

3. A $P C^r$-function with $r \geq 1$ is also called piecewise smooth.
Piecewise Smooth (PS) Functions

Definition (Piecewise Smoothness, Piecewise Linearity)

Let $D \subseteq \mathbb{R}^n$ be open and $f_i : D \rightarrow \mathbb{R}^m$, $i = 1, \ldots, k$ with $k \in \mathbb{N}$ be given.

- $f : D \rightarrow \mathbb{R}^m$ is called a continuous selection of the collection $f_1, \ldots, f_k$ on the set $U \subseteq D$ if $f$ is continuous on $U$ and $f(x) \in \{f_1(x), \ldots, f_k(x)\} \quad \forall x \in U$.

- $f : D \rightarrow \mathbb{R}^m$ is called $PC^r$-function with $r \in \mathbb{N} \cup \{\infty\}$ if for every $x \in D$ there exists an open neighbourhood $U \subseteq D$ and a finite number of $C^r$-functions $f_i : U \rightarrow \mathbb{R}^m$, $i = 1, \ldots, k$, such that $f$ is a continuous selection of $f_1, \ldots, f_k$ on $U$.

- A $PC^r$-function with $r \geq 1$ is also called piecewise smooth.

- A continuous selection $f : U \rightarrow \mathbb{R}^m$ is called piecewise linear if all elements of the collection $f_1, \ldots, f_k$ are affine functions.

Piecewise Smooth Example Problems

Exact $\ell_1$ penalty functions
Constrained optimization problem

$$\min_x f(x) \quad \text{s.t.} \quad c_i(x) = 0, \ i \in \mathcal{E}, \ c_i(x) \geq 0, \ i \in \mathcal{I}$$

equivalent to unconstrained optimization problem with $\ell_1$-penalty

$$\phi(x; \mu) = f(x) + \mu \sum_{i \in \mathcal{E}} |c_i(x)| + \mu \sum_{i \in \mathcal{I}} \max\{0, -c_i(x)\}$$
**Piecewise Smooth Example Problems**

**Exact $\ell_1$ penalty functions**

Constrained optimization problem

$$\min_{x} f(x) \quad \text{s.t.} \quad c_i(x) = 0, \; i \in \mathcal{E}, \quad c_i(x) \geq 0, \; i \in \mathcal{I}$$

equivalent to unconstrained optimization problem with $\ell_1$-penalty

$$\phi(x; \mu) = f(x) + \mu \sum_{i \in \mathcal{E}} |c_i(x)| + \mu \sum_{i \in \mathcal{I}} \max\{0, -c_i(x)\}$$

**Robust Optimization**

Often formulated as min-max problems
Piecewise Smooth Example Problems

Exact $\ell_1$ penalty functions
Constrained optimization problem

$$\min_x f(x) \quad \text{s.t.} \quad c_i(x) = 0, \; i \in \mathcal{E}, \quad c_i(x) \geq 0, \; i \in \mathcal{I}$$

equivalent to unconstrained optimization problem with $\ell_1$-penalty

$$\phi(x; \mu) = f(x) + \mu \sum_{i \in \mathcal{E}} |c_i(x)| + \mu \sum_{i \in \mathcal{I}} \max\{0, -c_i(x)\}$$

Robust Optimization

Often formulated as min-max problems

Train timetabling

yields piecewise linear optimization problem

F. Fischer, C. Helmberg: Dynamic Graph Generation and Dynamic Rolling Horizon Techniques in Large Scale Train Timetabling, 2010
Fuzzy Pattern Tree I

(together with Eyke Hüllermeier, Uni Pb)

= model class for classification and regression in machine learning
Fuzzy Pattern Tree I

(together with Eyke Hüllermeier, Uni Pb)

= model class for classification and regression in machine learning

Application: Determine wine quality
Fuzzy Pattern Tree I

(together with Eyke Hüllermeier, Uni Pb)

= model class for classification and regression in machine learning

Application: Determine wine quality via a target function defined by

\[
(\theta^*, \gamma^*, \sigma^*, c^*) = \arg\min_{\theta, \gamma, \sigma, c} \sum_{i=1}^{N} (F_{\theta, \gamma, \sigma, c}(x_i) - y_i)^2 \text{ with }
\]

\[
F_{\theta, \gamma, \sigma, c}(x) = T_{\theta}(\mu_{c_1}(x_{11}), C_{\gamma}(S_{\sigma}(\mu_{c_2}(x_2), \mu_{c_3}(x_{10})), \mu_{c_4}(x_2)))
\]
Fuzzy Pattern Tree II

Here:

\[ \mu_{c_i}(x) = \begin{cases} \frac{x}{c_i} & \text{if } 0 \leq x \leq c_i \\ \frac{1-x}{1-c_i} & \text{if } c_i \leq x \leq 1 \end{cases} \]

allow non-monotonicity

\[ T_\theta(u, v) = \frac{uv}{\max\{u, v, \theta\}} \]

= Dubois-Prade family

\[ S_\sigma(u, v) = 1 - T_\sigma(1-u, 1-v) \]

= corr. dual t-conorm

\[ C_\gamma(u, v) = \begin{cases} \gamma u + (1 - \gamma)v & \text{if } u > v \\ (1 - \gamma)u + \gamma v & \text{if } u \leq v \end{cases} \]

= ordered weighted operator
Fuzzy Pattern Tree II

Here:

\[
\mu_{c_i}(x) = \begin{cases} 
\frac{x}{c_i} & \text{if } 0 \leq x \leq c_i \\
\frac{1-x}{1-c_i} & \text{if } c_i \leq x \leq 1
\end{cases}
\]

allow non-monotonicity

\[
T_\theta(u, v) = \frac{uv}{\max\{u, v, \theta\}}
\]

= Dubois-Prade family

\[
S_\sigma(u, v) = 1 - T_\sigma(1 - u, 1 - v)
\]

= corr. dual t-conorm

\[
C_\gamma(u, v) = \begin{cases} 
\gamma u + (1 - \gamma)v & \text{if } u > v \\
(1 - \gamma)u + \gamma v & \text{if } u \leq v
\end{cases}
\]

= ordered weighted operator

⇒ Piecewise smooth target function

\[
(\theta^*, \gamma^*, \sigma^*, c^*) = \arg\min_{\theta, \gamma, \sigma, c} \sum_{i=1}^{N} (F_{\theta, \gamma, \sigma, c}(x_i) - y_i)^2 \quad \text{with}
\]

\[
F_{\theta, \gamma, \sigma, c}(x) = T_\theta(\mu_{c_1}(x_{11}), C_\gamma(S_\sigma(\mu_{c_2}(x_2), \mu_{c_3}(x_{10})), \mu_{c_4}(x_2)))
\]
**Optimality Conditions**

Generalized derivative concept required:

- directional derivative
- Clarke generalized gradient

\[
\partial_C \varphi(x) := \text{conv}\left\{ \lim_{i \to \infty} \nabla \varphi(x_i) : x_i \mapsto x, \nabla \varphi(x_i) \text{ exists} \right\} = \text{conv}\{ \partial^L \varphi(x) \}
\]

F. Clarke: Optimization and Nonsmooth Analysis, SIAM, 1990

- Mordukhovich subgradient \( \partial_M \varphi(x) \)

Optimality Conditions

Generalized derivative concept required:

- directional derivative
- Clarke generalized gradient

\[ \partial_C \varphi(x) := \text{conv}\left\{ \lim_{i \to \infty} \nabla \varphi(x_i) : x_i \mapsto x, \nabla \varphi(x_i) \text{ exists} \right\} = \text{conv}\{\partial^L \varphi(x)\} \]

F. Clarke: Optimization and Nonsmooth Analysis, SIAM, 1990

- Mordukhovich subgradient \( \partial_M \varphi(x) \)
  

Necessary optimality conditions:

- \( \varphi'(x; d) \geq 0 \) for all \( d \in \mathbb{R}^n \)
- Clarke stationarity: \( 0 \in \partial_C \varphi(x) \)
Optimality Conditions

Generalized derivative concept required:
- directional derivative
- Clarke generalized gradient

\[
\partial C \varphi(x) := \text{conv} \left\{ \lim_{i \to \infty} \nabla \varphi(x_i) : x_i \mapsto x, \nabla \varphi(x_i) \text{ exists} \right\} = \text{conv} \{ \partial^L \varphi(x) \}
\]

F. Clarke: Optimization and Nonsmooth Analysis, SIAM, 1990

- Mordukhovich subgradient \( \partial_M \varphi(x) \)

Necessary optimality conditions:
- \( \varphi'(x; d) \geq 0 \) for all \( d \in \mathbb{R}^n \)
- Clarke stationarity: \( 0 \in \partial C \varphi(x) \) \( \quad ? \quad \partial C(|x|) = \partial C(-|x|) \quad ! \)
Optimality Conditions

Generalized derivative concept required:
- directional derivative
- Clarke generalized gradient

\[ \partial_C \varphi(x) := \text{conv} \left\{ \lim_{i \to \infty} \nabla \varphi(x_i) : x_i \to x, \nabla \varphi(x_i) \text{ exists} \right\} = \text{conv} \{ \partial^L \varphi(x) \} \]

F. Clarke: Optimization and Nonsmooth Analysis, SIAM, 1990

- Mordukhovich subgradient \( \partial_M \varphi(x) \)

Necessary optimality conditions:
- \( \varphi'(x; d) \geq 0 \) for all \( d \in \mathbb{R}^n \)
- Clarke stationarity: \( 0 \in \partial_C \varphi(x) \) ? \( \partial_C(|x|) = \partial_C(-|x|) \) !
- a little stronger: Mordukhovich stationarity: \( 0 \in \partial_M \varphi(x) \)
Current (= Black Box) Approaches

- Use methods for smooth problems
  May fail, no convergence theory

- Subgradient method
  Very (!) slow convergence

- Bundle methods
  Lots of parameters, erratic convergence behaviour
  involves oracle

- Derivative-free methods
  No structure exploitation,
  difficult when number of optimization variables large
Hierarchy of Problems

locally Lipschitz continuous (LL)
∪
piecewise smooth (PS)
∪
piecewise linear (PL)
∪
piecewise linear and convex (PL+C)
Observations

Solving $\min \varphi(x)$ with $\varphi$ PL+C not easy:

- Global minimization is NP-hard
- Steepest descent with exact line search may fail
- Zeno behaviour possible, i.e., solution trajectory with infinite number of direction changes in a finite amount of time

New (= Gray Box) Approach

Goal: Locate stationary (?!?) point of piecewise smooth function $\varphi(.)$ by

- successive approximation by piecewise linear (PL) models and
- explicit handling of kink structure in PL model.
New (= Gray Box) Approach

Goal: Locate stationary (?!?) point of piecewise smooth function $\varphi(.)$ by

- successive approximation by piecewise linear (PL) models and
- explicit handling of kink structure in PL model.

Example: Half-Pipe function

$$\varphi : \mathbb{R}^2 \mapsto \mathbb{R}, \quad \varphi(x) = \max\{x_2^2 - \max\{x_1, 0\}, 0\}$$

Nonlinear function $\varphi(.)$ and its piecewise linearization at $\hat{x} = (1, 1)$
Abs-Linearisation I

**Given:** Target function $\varphi : \mathbb{R}^n \mapsto \mathbb{R}$ piecewise smooth

**Assumption:** Non-smoothness caused by univariate piecewise linear elements like min, max or abs!
Abs-Linearisation I

Given: Target function $\varphi : \mathbb{R}^n \mapsto \mathbb{R}$ piecewise smooth

Assumption: Non-smoothness caused by univariate piecewise linear elements like min, max or abs!

For example:

$$\varphi(x) = \min_{x \in \mathbb{R}^n} \max_{1 \leq i \leq m} f_i(x)$$

$$= \min \max \text{ regret problem}$$
Abs-Linearisation I

**Given:** Target function $\varphi : \mathbb{R}^n \mapsto \mathbb{R}$ piecewise smooth

**Assumption:** Non-smoothness caused by univariate piecewise linear elements like min, max or abs!

**Then:** $\varphi$ can be written using switching variables

$$z_i, \quad i = 1, \ldots, s$$

as arguments of $\text{abs}(.)$.  

A. Walther and A. Griewank 11 / 42 Abs-Linearization for PS Optimization Shonan, June 25, 2018
Abs-Linearisation I

**Given:** Target function \( \varphi : \mathbb{R}^n \mapsto \mathbb{R} \) piecewise smooth

**Assumption:** Non-smoothness caused by univariate piecewise linear elements like min, max or abs!

**Then:** \( \varphi \) can be written using switching variables

\[ z_i, \quad i = 1, \ldots, s \]

as arguments of abs(.)

Hence:

**Definition (Abs-normal form of PS function \( \varphi : \mathbb{R}^n \mapsto \mathbb{R} \))**

\[
F : \mathbb{R}^{n+s} \rightarrow \mathbb{R}^s, \quad z = F(x, |z|) \\
f : \mathbb{R}^{n+s} \rightarrow \mathbb{R}, \quad y = f(x, |z|) = \varphi(x)
\]

with \( F \) and \( f \) at least twice differentiable.
Abs-Linearisation II

Defining

\[ L = \frac{\partial}{\partial |z|} F(x, |z|) \in \mathbb{R}^{s \times s} \] 

strictly lower triangular

\[ Z = \frac{\partial}{\partial x} F(x, |z|) \in \mathbb{R}^{s \times n} \]

\[ a = \frac{\partial}{\partial x} f(x, |z|) \in \mathbb{R}^n, \quad b = \frac{\partial}{\partial |z|} f(x, |z|) \in \mathbb{R}^s \]
Abs-Linearisation II

Defining

\[ L = \frac{\partial}{\partial |z|} F(x, |z|) \in \mathbb{R}^{s \times s} \] strictly lower triangular

\[ Z = \frac{\partial}{\partial x} F(x, |z|) \in \mathbb{R}^{s \times n} \]

\[ a = \frac{\partial}{\partial x} f(x, |z|) \in \mathbb{R}^n, \quad b = \frac{\partial}{\partial |z|} f(x, |z|) \in \mathbb{R}^s \]

one obtains

**Definition (Abs-linear form of abs-normal \( \varphi : \mathbb{R}^n \to \mathbb{R} \) in \( x \))**

\[
\begin{bmatrix}
  z \\
  \Delta y
\end{bmatrix} =
\begin{bmatrix}
  c_1 \\
  c_2
\end{bmatrix} +
\begin{bmatrix}
  Z & L \\
  a & b
\end{bmatrix}
\begin{bmatrix}
  \Delta x \\
  \Sigma \cdot z
\end{bmatrix}
\]

with

\[ c_1 \in \mathbb{R}^s, \quad c_2 \in \mathbb{R}, \quad \sigma = \sigma(x) \equiv \text{sign}(z(x)) \in \{-1, 0, 1\}^s, \quad \Sigma \equiv \text{diag}(\sigma) \]

as piecewise linearisation \( \Delta \varphi \) of \( \varphi \) in \( x \).
Abs-Linearisation II

Defining

\[ L = \frac{\partial}{\partial |z|} F(x, |z|) \in \mathbb{R}^{s \times s} \]  
strictly lower triangular

\[ Z = \frac{\partial}{\partial x} F(x, |z|) \in \mathbb{R}^{s \times n} \]

\[ a = \frac{\partial}{\partial x} f(x, |z|) \in \mathbb{R}^n, \quad b = \frac{\partial}{\partial |z|} f(x, |z|) \in \mathbb{R}^s \]

one obtains

**Definition (Abs-linear form of abs-normal \( \varphi : \mathbb{R}^n \rightarrow \mathbb{R} \) in \( x \))**

\[
\begin{bmatrix}
  z \\
  \Delta y
\end{bmatrix} =
\begin{bmatrix}
  c_1 \\
  c_2
\end{bmatrix} +
\begin{bmatrix}
  Z & L \\
  a & b
\end{bmatrix}
\begin{bmatrix}
  \Delta x \\
  \Sigma \cdot z
\end{bmatrix}
\]

with

\[ c_1 \in \mathbb{R}^s, \ c_2 \in \mathbb{R}, \ \sigma = \sigma(x) \equiv \text{sign}(z(x)) \in \{-1, 0, 1\}^s, \Sigma \equiv \text{diag}(\sigma) \]

as piecewise linearisation \( \Delta \varphi \) of \( \varphi \) in \( x \).

Abs-normal form can be generated using appropriate variant of AD!
Example: Nesterov-Rosenbrock Function

Smooth variant:

\[ \varphi_0(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} (x_{i+1} - 2x_i^2 + 1)^2 \]
Example: Nesterov-Rosenbrock Function

PS variant:

\[
\varphi_1(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} \left| x_{i+1} - 2x_i^2 + 1 \right|
\]

Example: Nesterov-Rosenbrock Function

PS variant:

\[ \varphi_1(x) = \frac{1}{4} (x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1| \]

Abs-normal form:

\[ z_i = F_i(x, |z|) = x_{i+1} - 2x_i^2 + 1, \quad 1 \leq i \leq n - 1, \]

\[ y = f(x, |z|) = \frac{1}{4} (x_1 - 1)^2 + \sum_{i=1}^{n-1} |z_i| \quad \Rightarrow \]
**Example: Nesterov-Rosenbrock Function**

PS variant:

\[ \varphi_1(x) = \frac{1}{4} (x_1 - 1)^2 + \sum_{i=1}^{n-1} |x_{i+1} - 2x_i^2 + 1| \]

Abs-normal form:

\[ z_i = F_i(x, |z|) = x_{i+1} - 2x_i^2 + 1, \quad 1 \leq i \leq n - 1, \]

\[ y = f(x, |z|) = \frac{1}{4} (x_1 - 1)^2 + \sum_{i=1}^{n-1} |z_i| \quad \Rightarrow \]

\[ Z = \begin{bmatrix} -4x_1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -4x_2 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -4x_s & 1 \end{bmatrix} \in \mathbb{R}^{(n-1) \times n} \]

\[ L = 0 \in \mathbb{R}^{(n-1) \times (n-1)}, \quad a = \left( \frac{(x_1 - 1)}{2}, 0, \ldots, 0 \right) \in \mathbb{R}^n, \quad b = \mathbf{1} \in \mathbb{R}^{n-1} \]
Open Questions I

Gap between class of abs-normal functions and PS functions?
Original Evaluation Procedure

For smooth functions, AD is based on

\[
\begin{align*}
  v_{i-r} &= x_i & i &= 1 \ldots n \\
  v_i &= \varphi_i(v_j)_{j \prec i} & i &= 1 \ldots l \\
  y &= v_l
\end{align*}
\]
Adapted Evaluation Procedure

For abs-normal functions, consider

\[
\begin{align*}
    v_{i-n} & = x_i & i = 1 \ldots n \\
    z_i & = \psi_i(v_j)_{j<i} \\
    \sigma_i & = \text{sign}(z_i) \\
    v_i & = \sigma_i z_i = \text{abs}(z_i) \\
    y & = v_{s+1} = \psi_{s+1}(v_j)_{j<s+1}
\end{align*}
\]

- Declare \( z_i \) as independent variables
- adapt evaluation of abs() correspondingly
Abs-Linearisation via AD I

**AD approach:** tangent approximation for each elemental function

\[ v_i(x + \Delta x) - v_i(x) \approx \Delta v_i \equiv \Delta v_i(\Delta x) \]

For smooth elementals:

\[ \Delta v_i = \Delta v_j \pm \Delta v_k \quad \text{for } v_i = v_j \pm v_k, \]
\[ \Delta v_i = v_j \cdot \Delta v_k + v_k \cdot \Delta v_j \quad \text{for } v_i = v_j \cdot v_k, \]
\[ \Delta v_i = \varphi'(v_j)_{j \prec i} \cdot \Delta (v_j)_{j \prec i} \quad \text{for } v_i = \varphi_i(v_j)_{j \prec i} \neq \text{abs}(v_j), \]
\[ \Delta v_i = \text{abs}(v_j + \Delta v_j) - v_i \quad \text{for } v_i = \text{abs}(v_j). \]

⇒ If \( y = F(x) \) involves no call of \text{abs}():

\[ \Delta y = \Delta F(x; \Delta x) = F'(x)\Delta x, \quad F'(x) \in \mathbb{R}^{m \times n} = \text{Jacobian} \]

standard AD!
Abs-Linearisation via AD II

For the absolute value function $v_i = \text{abs}(v_j)$:

$$\Delta v_i = \text{abs}(v_j(\hat{x}) + \Delta v_j) - v_j(\hat{x})$$

$$\Rightarrow \Delta y(\Delta x) = \Delta F(\hat{x}; \Delta x) : \mathbb{R}^n \mapsto \mathbb{R}^m$$

is a piecewise linear continuous function for each fixed $x \in D$. 

A. Walther and A. Griewank 18 / 42 Abs-Linearization for PS Optimization Shonan, June 25, 2018
Abs-Linearisation via AD II

For the absolute value function $v_i = \text{abs}(v_j)$:

$$\Delta v_i = \text{abs}(v_j(\hat{x}) + \Delta v_j) - v_j(\hat{x})$$

$$\Rightarrow \Delta y(\Delta x) = \Delta F(\hat{x}; \Delta x) : \mathbb{R}^n \mapsto \mathbb{R}^m$$

is a piecewise linear continuous function for each fixed $x \in D$.

**Theorem**

Suppose $F$ is elementwise Lipschitz continuously differentiable on $D \subset K \subset \mathbb{R}^n$, $D$ open, $K$ closed and convex. Then there exists $\gamma > 0$ such that for all $x, \hat{x} \in K$

$$\|F(x) - F(\hat{x}) - \Delta F(\hat{x}; x - \hat{x})\| = \gamma \|x - \hat{x}\|^2$$

Abs-Linearisation via AD II

For the absolute value function \( v_i = \text{abs}(v_j) \):

\[
\Delta v_i = \text{abs}(v_j(\hat{x}) + \Delta v_j) - v_j(\hat{x})
\]

\( \Rightarrow \Delta y(\Delta x) = \Delta F(\hat{x}; \Delta x) : \mathbb{R}^n \rightarrow \mathbb{R}^m \)

is a piecewise linear continuous function for each fixed \( x \in D \).

Theorem

Suppose \( F \) is elementwise Lipschitz continuously differentiable on \( D \subset K \subset \mathbb{R}^n \), \( D \) open, \( K \) closed and convex. Then there exists \( \gamma > 0 \) such that for all \( x, \hat{x} \in K \)

\[
\| F(x) - F(\hat{x}) - \Delta F(\hat{x}; x - \hat{x}) \| = \gamma \| x - \hat{x} \|^2
\]


Derivatives \( a, b, c, Z, L \) required by abs-linear form provided by AD!
Open Questions II

Drivers/Interfaces of AD tools for abs-linearisation?
Very brief description of the algorithm:

\[ x_{k+1} = x_k + \arg \min_{\Delta x} \{ \Delta \varphi(x_k; \Delta x) + \frac{q}{2} \| \Delta x \|^2 \} \]

= Successive Abs-Linear OPtimization with a proximal term
The SALOP Algorithm

SALOP

Initialization $x_0, k = 0, q > 0$

Outer optimization loop

Create piecewise linear model at $x_k$

Inner optimization loop

Add quadratic penalty using $q$ as multiplier

Solve inner problem $\rightarrow \Delta x_k$

Check optimality of $x_k + \Delta x_k$

Update $x_{k+1}$ and $q$, $k = k + 1$

STOP
Example

\[ \varphi : \mathbb{R}^2 \to \mathbb{R}, \quad \varphi(x_1, x_2) = \max\{x_2^2 - \max\{x_1, 0\}, 0\} \]

\( k = 0 \)
Example

\[ \varphi : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad \varphi(x_1, x_2) = \max\{x_2^2 - \max\{x_1, 0\}, 0\} \]

\[ k = 0 \]

local QP in \( x_0 \) based on linearization

\[ \rightarrow \]
Example

\[ \varphi : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad \varphi(x_1, x_2) = \max\{x_2^2 - \max\{x_1, 0\}, 0\} \]

\( k = 0 \)

local QP in \( x_0 \) based on linearization

\[ x_1 = x_0 + \Delta x_0 \]
Example

\[ \varphi : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad \varphi(x_1, x_2) = \max\{x_2^2 - \max\{x_1, 0\}, 0\} \]

\( k = 0 \)

Local QP in \( x_0 \) based on linearization

New iterate

\( x_1 = x_0 + \Delta x_0 \)

\( k = 1 \)
Example

\[ \phi : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad \phi(x_1, x_2) = \max\{x_2^2 - \max\{x_1, 0\}, 0\} \]

\[ k = 0 \]

\[ \text{local QP in } x_0 \]
\[ \text{based on linearization} \]
\[ \rightarrow \]
\[ \text{New iterate} \]
\[ x_1 = x_0 + \Delta x_0 \]

\[ k = 1 \]

\[ \text{Local QP in } x_1 \]
\[ \text{based on linearization} \]
\[ \rightarrow \]
Convergence of SALOP

Finite convergence of inner loop:
- Argument space divided into finitely many polyhedra
- Function value decreased when switching polyhedra
- No polyhedron visited twice

⇒ stationary point reached after finitely many steps
Convergence of SALOP

Finite convergence of inner loop:
- Argument space divided into finitely many polyhedra
- Function value decreased when switching polyhedra
- No polyhedron visited twice
⇒ stationary point reached after finitely many steps

Convergence of outer loop:

**Theorem**

Assume that \( \varphi : \mathbb{R}^n \to \mathbb{R} \) is a PS function as considered here with a bounded level set \( \mathcal{N}_0 = \{ x \in \mathbb{R}^n \mid f(x) \leq f(x_0) \} \). Let \( x_0 \) be the starting point of the generated sequence of iterates \( \{x_k\}_{k \in \mathbb{N}} \) generated by SALOP. Then a cluster point \( x_* \) of the infinite sequence \( \{x_k\}_{k \in \mathbb{N}} \) exists and all clusters points are Clarke stationary.
Convergence of SALOP

Finite convergence of inner loop:
- Argument space divided into finitely many polyhedra
- Function value decreased when switching polyhedra
- No polyhedron visited twice

⇒ stationary point reached after finitely many steps

Convergence of outer loop:

**Theorem**

Assume that $\varphi : \mathbb{R}^n \to \mathbb{R}$ is a PS function as considered here with a bounded level set $\mathcal{N}_0 = \{ x \in \mathbb{R}^n \mid f(x) \leq f(x_0) \}$. Let $x_0$ be the starting point of the generated sequence of iterates $\{ x_k \}_{k \in \mathbb{N}}$ generated by SALOP. Then a cluster point $x_*$ of the infinite sequence $\{ x_k \}_{k \in \mathbb{N}}$ exists and all clusters points are Clarke stationary.

S. Fiege, A. Walther, A. Griewank: An algorithm for nonsmooth optimization by successive piecewise linearization. Mathematical Programming, 2018
The Inner Optimisation Loop

Improved solver for inner loop:
- adaption of new optimality conditions for inner loop
- corresponding modification of QP solver

⇒ Active Signature Method (ASM)
  for the first time convergence to local minimizers!

Example: Nesterov-Rosenbrock function with $2^{n} - 1$ Clarke-stationary points

\[ \varphi_{n} : \mathbb{R}^{n} \to \mathbb{R}, \varphi_{n}(x) = \frac{1}{4} |x_{1} - 1| + \sum_{i=1}^{n-1} |x_{i+1} - 2| x_{i} + 1 \]
The Inner Optimisation Loop

Improved solver for inner loop:
- adaption of new optimality conditions for inner loop
- corresponding modification of QP solver
⇒ Active Signature Method (ASM)
  for the first time convergence to local minimizers!

Exam.: Nesterov-Rosenbrock function with \(2^{n-1}\) Clarke-stationary points

\[
\varphi_2 : \mathbb{R}^n \mapsto \mathbb{R}, \quad \varphi(x) = \frac{1}{4} |x_1 - 1| + \sum_{i=1,\ldots,n-1} |x_{i+1} - 2|x_i| + 1|
\]
The Inner Optimisation Loop

Improved solver for inner loop:

- adaptation of new optimality conditions for inner loop
- corresponding modification of QP solver

⇒ Active Signature Method (ASM)
for the first time convergence to local minimizers!

Exam.: Nesterov-Rosenbrock function with $2^{n-1}$ Clarke-stationary points

$$\varphi_2 : \mathbb{R}^n \mapsto \mathbb{R}, \quad \varphi(x) = \frac{1}{4} |x_1 - 1| + \sum_{i=1, \ldots, n-1} |x_{i+1} - 2|x_i| + 1|$$

Iterations numbers:

<table>
<thead>
<tr>
<th>n</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASM</td>
<td></td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
<td>256</td>
<td>512</td>
</tr>
<tr>
<td>HANSO</td>
<td></td>
<td>3</td>
<td>61</td>
<td>494*</td>
<td>1341*</td>
<td>2521*</td>
<td>329*</td>
<td>357*</td>
<td>326*</td>
<td>307*</td>
</tr>
<tr>
<td>MPBNGC</td>
<td>3</td>
<td>52</td>
<td>9859</td>
<td>9978*</td>
<td>3561*</td>
<td>4166*</td>
<td>2547*</td>
<td>1959*</td>
<td>9420*</td>
<td>9807*</td>
</tr>
</tbody>
</table>

A. Griewank, A. Walther: Finite convergence of an active signature method to local minima of piecewise linear functions. In revision + Matlab Implementierung von ASM
The LASSO Problem

In statistics and machine learning: Least Absolute Shrinkage and Selection Operator (LASSO)

= regression approach for variable selection and regularization to enhance prediction accuracy and interpretability of statistical model it produces
The LASSO Problem

In statistics and machine learning: Least Absolute Shrinkage and Selection Operator (LASSO)

= regression approach for variable selection and regularization to enhance prediction accuracy and interpretability of statistical model it produces

For given data $w \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m \times n}$, the LASSO function is

$$\varphi : \mathbb{R}^n \mapsto \mathbb{R}, \quad \varphi(x) = \frac{1}{m} \|w - Ax\|_2^2 + \rho \|x\|_1$$

with the penalty factor $\rho > 0$. 
The LASSO Problem

In statistics and machine learning: Least Absolute Shrinkage and Selection Operator (LASSO)

= regression approach for variable selection and regularization to enhance prediction accuracy and interpretability of statistical model it produces

For given data \( w \in \mathbb{R}^m \) and \( A \in \mathbb{R}^{m \times n} \), the LASSO function is

\[
\varphi: \mathbb{R}^n \mapsto \mathbb{R}, \quad \varphi(x) = \frac{1}{m} \| w - Ax \|_2^2 + \rho \| x \|_1
\]

with the penalty factor \( \rho > 0 \).

ASM with adapted quadratic term!!
## LASSO: Iteration Numbers

<table>
<thead>
<tr>
<th>Löser</th>
<th>ρ = 100 opt. value</th>
<th># iter.</th>
<th>ρ = 17.353616 opt. value</th>
<th># iter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Signature Method</td>
<td>13035.7</td>
<td>3</td>
<td>11452.1</td>
<td>3</td>
</tr>
<tr>
<td>LassoBlockCoordinate</td>
<td>13035.7</td>
<td>30</td>
<td>11452.1</td>
<td>29</td>
</tr>
<tr>
<td>LassoConstrained</td>
<td>13035.7</td>
<td>8</td>
<td>11452.1</td>
<td>6</td>
</tr>
<tr>
<td>LassoGaussSeidel</td>
<td>13035.7</td>
<td>12</td>
<td>11452.1</td>
<td>11</td>
</tr>
<tr>
<td>LassoGrafting</td>
<td>13087.2</td>
<td>10</td>
<td>11452.1</td>
<td>11</td>
</tr>
<tr>
<td>LassoIteratedRidge</td>
<td>13087.2</td>
<td>102</td>
<td>11452.1</td>
<td>102</td>
</tr>
<tr>
<td>LassoNonNegativeSquared</td>
<td>13035.7</td>
<td>64</td>
<td>11452.1</td>
<td>58</td>
</tr>
<tr>
<td>LassoPrimalDualLogBarrier</td>
<td>13035.7</td>
<td>9</td>
<td>11452.1</td>
<td>7</td>
</tr>
<tr>
<td>LassoProjection</td>
<td>13035.7</td>
<td>3</td>
<td>11452.1</td>
<td>5</td>
</tr>
<tr>
<td>LassoShooting</td>
<td>13035.7</td>
<td>54</td>
<td>11452.1</td>
<td>51</td>
</tr>
<tr>
<td>LassoSubGradient</td>
<td>13035.7</td>
<td>52</td>
<td>11452.1</td>
<td>23</td>
</tr>
<tr>
<td>LassoUnconstrainedApx v1</td>
<td>13035.7</td>
<td>50</td>
<td>11452.1</td>
<td>40</td>
</tr>
<tr>
<td>LassoUnconstrainedApx v2</td>
<td>13035.7</td>
<td>94</td>
<td>11452.1</td>
<td>27</td>
</tr>
<tr>
<td>LassoActiveSet</td>
<td>13288.9</td>
<td>14</td>
<td>11602.1</td>
<td>12</td>
</tr>
<tr>
<td>LassoLARS</td>
<td>13296.7</td>
<td>18</td>
<td>11602.1</td>
<td>14</td>
</tr>
<tr>
<td>LassoSignConstraints</td>
<td>13288.9</td>
<td>1</td>
<td>11602.1</td>
<td>4</td>
</tr>
</tbody>
</table>

Quadratic Convergence

**Proposition**

If $x_*$ is a sharp minimizer of $\varphi$ then SALOP with $q \geq \gamma$ converges quadratically to $x_*$ from all $x_0$ in some ball $B_\rho(x_*)$. 

Proof.

$$\|x_{k+1} - x_*\| \leq \varphi(x_{k+1}) - \varphi(x_*) = \varphi(x_{k+1}) - \varphi(x_k) - \left(\varphi(x_*) - \varphi(x_k)\right) \leq \Delta \varphi(x_k; x_{k+1} - x_k) - \Delta \varphi(x_k; x_* - x_k) + \gamma^2 \left(\|x_{k+1} - x_k\|^2 + \|x_* - x_k\|^2\right) \leq \gamma \|x_k - x_*\|^2.$$
Quadratic Convergence

Proposition

If $x_*$ is a sharp minimizer of $\varphi$ then SALOP with $q \geq \gamma$ converges quadratically to $x_*$ from all $x_0$ in some ball $B_\rho(x_*)$.

Proof.

$$c\|x_k - x_*\| \leq \varphi(x_k+1) - \varphi(x_*) = \varphi(x_k+1) - \varphi(x_k) - (\varphi(x_*) - \varphi(x_k))$$

$$\leq \Delta \varphi(x_k; x_k+1 - x_k) - \Delta \varphi(x_k; x_* - x_k)$$

$$+ \frac{\gamma}{2} (\|x_k+1 - x_k\|^2 + \|x_* - x_k\|^2)$$

$$\leq \frac{\gamma + q}{2} \|x_k+1 - x_k\|^2 + \frac{\gamma - q}{2} \|x_k - x_*\|^2 \leq \gamma \|x_k - x_*\|^2.$$
Chained CB3 I

$$\varphi : \mathbb{R}^n \mapsto \mathbb{R}, \varphi(x) = \sum_{i=1}^{n-1} \max \{x_i^4 + x_{i+1}^2, (2 - x_i)^2 + (2 - x_{i+1})^2, 2e^{-x_i + x_{i+1}} \}$$

$$s = 2(n - 1), \quad x_* = (1 \ldots 1)^\top \in \mathbb{R}^n \text{ is sharp}$$
The SALOP Algorithm

Chained CB3 I

\[ \varphi : \mathbb{R}^n \mapsto \mathbb{R}, \varphi(x) = \sum_{i=1}^{n-1} \max\{x_i^4 + x_{i+1}^2, (2 - x_i)^2 + (2 - x_{i+1})^2, 2e^{-x_i + x_{i+1}}\} \]

\[ s = 2(n - 1), \quad x_0 = (1 \ldots 1)^	op \in \mathbb{R}^n \text{ is sharp} \]

Implementation LiPsMin of SALOP yields for \( n = 10 \)

\[ \log(||x_{i+1} - 1||)/\log(||x_i - 1||) \]

![Graph showing the log ratio of distances between consecutive iterations for \( n = 10 \).]
The SALOP Algorithm

Linear Convergence

Proposition

Suppose $x_*$ satisfies SSC with strict complementarity under LIKQ for $\varphi(.)$. Assume $q > \max(\gamma, \|\tilde{U}^*_\top \tilde{H}_* \tilde{U}_*\|)$ for the proximal parameter $q$. Then SALOP yields local and linear convergence with $R$-factor

$$\|I - \frac{1}{q} \tilde{U}^*_\top \tilde{H}_* \tilde{U}_*\| \geq 1 - (\kappa(\tilde{U}^*_\top \tilde{H}_* \tilde{U}_*))^{-1},$$

where $\kappa$ denotes the condition number with respect to the spectral norm.
Linear Convergence

Proposition

Suppose $x_*$ satisfies SSC with strict complementarity under LIKQ for $\varphi(.)$. Assume $q > \max(\gamma, \|\tilde{U}_*^\top \tilde{H}_* \tilde{U}_*\|)$ for the proximal parameter $q$. Then SALOP yields local and linear convergence with R-factor

$$\|I - \frac{1}{q} \tilde{U}_*^\top \tilde{H}_* \tilde{U}_*\| \geq 1 - (\kappa(\tilde{U}_*^\top \tilde{H}_* \tilde{U}_*))^{-1},$$

where $\kappa$ denotes the condition number with respect to the spectral norm.

Proof.

- take care of nonlocalization
- formulation as fixed point problem, analysis of contraction rate

A. Griewank and A. Walther: Relaxing kink qualifications and proving convergence rates in piecewise smooth optimization, in revision
Chained Crescent I

\[ \varphi : \mathbb{R}^n \mapsto \mathbb{R}, \quad \varphi(x) = \max \{ \varphi_1(x), \varphi_2(x) \} \]

\[ \varphi_1(x) = \sum_{i=1}^{n-1} (x_i^2 + (x_{i+1} - 1)^2 + x_{i+1} - 1), \]

\[ \varphi_2(x) = \sum_{i=1}^{n-1} (-x_i^2 - (x_{i+1} - 1)^2 + x_{i+1} + 1), \]

⇒ PS, nonconvex function
isolated but not sharp minimizer \(x_\ast = (1 \ldots 1) \top \in \mathbb{R}^n, s = 1,\)

\[ Z = (0 \ 4 \ \ldots \ 4), \quad L = 0 \in \mathbb{R}, \quad a = (0 \ 1 \ \ldots \ 1), \quad b = 0.5, \]

only switching variable is active at \(x_\ast, \text{LIKQ holds}\)
Chained Crescent I: Convergence

\[ ||x_{i+1}|| / ||x_i|| \]
FPT Problem: Wine Quality

- data set contains 4000 entries
- C implementation with old inner loop algo. could handle 200 entries
- Matlab implementation: up to 4000 entries feasible!
- $n = 7, s = 18014$ for $m = 2000$ entries
  - $n = 7, s = 27014$ for $m = 3000$ entries $\Rightarrow$ large, sparse matrices!
FPT Problem: Wine Quality

- data set contains 4000 entries
- C implementation with old inner loop algo. could handle 200 entries
- Matlab implementation: up to 4000 entries feasible!
- $n = 7$, $s = 18014$ for $m = 2000$ entries
  $n = 7$, $s = 27014$ for $m = 3000$ entries $\rightarrow$ large, sparse matrices!
FPT Problem: Wine Quality

- data set contains 4,000 entries
- C implementation with old inner loop algo. could handle 200 entries
- Matlab implementation: up to 4,000 entries feasible!
- $n = 7$, $s = 18014$ for $m = 2,000$ entries
  $n = 7$, $s = 27014$ for $m = 3,000$ entries $\implies$ large, sparse matrices!

![Function Value vs. Outer Iteration for m = 3000](image.png)

Function Value

Outer Iteration, $m = 3000$
Open Questions III

- linear convergence with fewer assumptions?
- superlinear convergence?
- larger class of functions?
Signature Vectors

The signature vector

\[ \sigma(x) = \text{sign}(z(x)) \]

and the corresponding diagonal matrix

\[ \Sigma = \text{diag}(\sigma) \]

define active switch set

\[ \alpha = \alpha(x) \equiv \{ 1 \leq i \leq s \mid \sigma_i(x) = 0 \} \quad |\alpha(x)| = s - |\sigma(x)|. \]
**Signature Vectors**

The signature vector

$$\sigma(x) = \text{sign}(z(x))$$

and the corresponding diagonal matrix

$$\Sigma = \text{diag}(\sigma)$$

define active switch set

$$\alpha = \alpha(x) \equiv \{1 \leq i \leq s \mid \sigma_i(x) = 0\} \quad |\alpha(x)| = s - |\sigma(x)|.$$ 

Furthermore, for fixed $\sigma$ and hence also $\Sigma$

$$z = F(x, \Sigma z)$$

has unique solution $z^\sigma$ with $\nabla z^\sigma = \frac{\partial}{\partial x} z^\sigma = \left( I - L\Sigma \right)^{-1} z$. 
Signature Vectors

The signature vector

$$\sigma(x) = \text{sign}(z(x))$$

and the corresponding diagonal matrix

$$\Sigma = \text{diag}(\sigma)$$

define active switch set

$$\alpha = \alpha(x) \equiv \{1 \leq i \leq s \mid \sigma_i(x) = 0\} \quad |\alpha(x)| = s - |\sigma(x)|.$$  

Furthermore, for fixed $\sigma$ and hence also $\Sigma$

$$z = F(x, \Sigma z)$$

has unique solution $z^\sigma$ with $\nabla z^\sigma = \frac{\partial}{\partial x} z^\sigma = (I - L \Sigma)^{-1} z$. 
Linear Independence Kink Qualification

**Definition**

We say that the linear independence kink qualification (LIKQ) is satisfied at a point \( x \in \mathbb{R}^n \) if for \( \sigma = \sigma(x) \) the active Jacobian

\[
J(x) \equiv \nabla z_\alpha^\sigma(x) \equiv (e_i^T \nabla z^{\sigma}(x))_{i \in \alpha} \in \mathbb{R}^{\left|\alpha\right| \times n}
\]

has full row rank \( \left|\alpha\right| \), which requires in particular that \( \left|\sigma\right| \geq s - n \).
Relation to Other Derivative Concepts

Linear Independence Kink Qualification

Definition

We say that the linear independence kink qualification (LIKQ) is satisfied at a point \( x \in \mathbb{R}^n \) if for \( \sigma = \sigma(x) \) the active Jacobian

\[
J(x) \equiv \nabla z_\alpha^\sigma(x) \equiv (e_i^T \nabla z^\sigma(x))_{i \in \alpha} \in \mathbb{R}^{|\alpha| \times n}
\]

has full row rank \(|\alpha|\), which requires in particular that \(|\sigma| \geq s - n\).

Generalization of LICQ!
Generalized Gradients by AD

Definition

For a PS function $\varphi : \mathbb{R}^n \mapsto \mathbb{R}$ as considered here and a point $x \in \mathbb{R}^n$ the set of conical gradients is given by

$$\partial^K \varphi(x) = \{ g \in \mathbb{R}^n | g \in \partial^L \Delta \varphi(x; \Delta x)|_{\Delta x=0} \}.$$

Griewank (2013), considered also by Barton and Khan, see publications in 2013 and 2015.
Generalized Gradients by AD

**Definition**

For a PS function $\varphi : \mathbb{R}^n \mapsto \mathbb{R}$ as considered here and a point $x \in \mathbb{R}^n$ the set of conical gradients is given by

$$\partial^K \varphi(x) = \left\{ g \in \mathbb{R}^n \middle| g \in \partial^L \Delta \varphi(x; \Delta x) \bigg|_{\Delta x = 0} \right\}.$$

- Griewank (2013), considered also by Barton and Khan, see publications in 2013 and 2015
- Can be computed from the abs-normal form, i.e., they are available using AD
- A directional active gradient computed by AD is an element of the limiting gradients, i.e., $g \in \partial^L \varphi(x)$. 
The Half-Pipe Example

\( \varphi : \mathbb{R}^2 \mapsto \mathbb{R} \),

\[ \varphi(x_1, x_2) = \max(x_2^2 - \max(x_1, 0), 0) \]

\[ = \begin{cases} 
  x_2^2 & \text{if } x_1 \leq 0 \\
  x_2^2 - x_1 & \text{if } 0 \leq x_1 \leq x_2^2 \\
  0 & \text{if } 0 \leq x_2^2 \leq x_1 
\end{cases} \]
The Half-Pipe Example

\[ \varphi : \mathbb{R}^2 \mapsto \mathbb{R}, \quad \varphi(x_1, x_2) = \max(x_2^2 - \max(x_1, 0), 0) \]

\[
\begin{align*}
\varphi(x_1, x_2) &= \begin{cases} 
x_2^2 & \text{if } x_1 \leq 0 
x_2^2 - x_1 & \text{if } 0 \leq x_1 \leq x_2^2 
0 & \text{if } 0 \leq x_2^2 \leq x_1
\end{cases},
\end{align*}
\]

Here, one has that

\[ \hat{\partial}^M \varphi(0) = \{(0, 0)\} \subsetneq \partial^M \varphi(0) = \{(0, 0), (-1, 0)\} = \partial^L \varphi(0), \]

\[ \Rightarrow \quad \partial^C \varphi(0) = \{(v, 0) \mid v \in [-1, 0]\}, \]

\[ \partial^K \varphi(0) = \partial^L \Delta \varphi(0; 0) = \{(0, 0)\} \]
The Half-Pipe Example

\[ \varphi : \mathbb{R}^2 \mapsto \mathbb{R}, \quad \varphi(x_1, x_2) = \max(x_2^2 - \max(x_1, 0), 0) \]

\[ = \begin{cases} 
  x_2^2 & \text{if } x_1 \leq 0 \\
  x_2^2 - x_1 & \text{if } 0 \leq x_1 \leq x_2^2 \\
  0 & \text{if } 0 \leq x_2^2 \leq x_1 
\end{cases} , \]

Here, one has that

\[ \hat{\partial}^M \varphi(0) = \{ (0, 0) \} \subsetneq \partial^M \varphi(0) = \{ (0, 0), (-1, 0) \} = \partial^L \varphi(0), \]

\[ \Rightarrow \partial^C \varphi(0) = \{ (v, 0) \mid v \in [-1, 0] \}, \]

\[ \partial^K \varphi(0) = \partial^L \Delta \varphi(0; 0) = \{ (0, 0) \} \]

yielding

\[ \{ \nabla \varphi(0) \} = \hat{\partial}^M \varphi(0) \subsetneq \partial^M \varphi(0) = \partial^L \varphi(0), \]
The Half-Pipe Example

\[ \varphi : \mathbb{R}^2 \mapsto \mathbb{R}, \quad \varphi(x_1, x_2) = \max(x_2^2 - \max(x_1, 0), 0) \]

\[ = \begin{cases} 
  x_2^2 & \text{if } x_1 \leq 0 \\
  x_2^2 - x_1 & \text{if } 0 \leq x_1 \leq x_2^2 \\
  0 & \text{if } 0 \leq x_2^2 \leq x_1
\end{cases} \]

Here, one has that

\[ \hat{\partial}^M \varphi(0) = \{(0,0)\} \subsetneq \partial^M \varphi(0) = \{(0,0), (-1,0)\} = \partial^L \varphi(0), \]

\[ \Rightarrow \partial^C \varphi(0) = \{(v,0) \mid v \in [-1,0]\}, \]

\[ \partial^K \varphi(0) = \partial^L \Delta \varphi(0;0) = \{(0,0)\} \]

yielding

\[ \{\nabla \varphi(0)\} = \hat{\partial}^M \varphi(0) \subsetneq \partial^M \varphi(0) = \partial^L \varphi(0), \quad \partial^L \varphi(0) \neq \partial^K \varphi(0) \subset \partial^C \varphi(0). \]
Gradient Cube Example \((n = 2)\)

\[
\varphi : \mathbb{R}^2 \mapsto \mathbb{R}, \quad \varphi(x_1, x_2) = |x_2 - |x_1|| + \varepsilon |x_1|, \quad \varepsilon \in \mathbb{R}
\]

\[
= \begin{cases} 
\varphi_1(x_1, x_2) = x_2 - x_1 + \varepsilon x_1 & \text{if } x_2 \geq x_1 \geq 0 \\
\varphi_2(x_1, x_2) = x_2 + x_1 - \varepsilon x_1 & \text{if } x_2 \geq -x_1, x_1 < 0 \\
\varphi_3(x_1, x_2) = -x_2 - x_1 - \varepsilon x_1 & \text{if } x_2 < -x_1, x_1 < 0 \\
\varphi_4(x_1, x_2) = -x_2 + x_1 + \varepsilon x_1 & \text{if } x_1 > x_2, x_1 \geq 0
\end{cases}
\]
Gradient Cube Example \((n = 2)\)

\[
\varphi : \mathbb{R}^2 \mapsto \mathbb{R}, \quad \varphi(x_1, x_2) = |x_2| - |x_1| + \varepsilon |x_1|, \quad \varepsilon \in \mathbb{R}
\]

\[
= \begin{cases}
\varphi_1(x_1, x_2) = x_2 - x_1 + \varepsilon x_1 & \text{if } x_2 \geq x_1 \geq 0 \\
\varphi_2(x_1, x_2) = x_2 + x_1 - \varepsilon x_1 & \text{if } x_2 \geq -x_1, x_1 < 0 \\
\varphi_3(x_1, x_2) = -x_2 - x_1 - \varepsilon x_1 & \text{if } x_2 < -x_1, x_1 < 0 \\
\varphi_4(x_1, x_2) = -x_2 + x_1 + \varepsilon x_1 & \text{if } x_1 > x_2, x_1 \geq 0
\end{cases}
\]

If \(\varepsilon \geq 1\)

\[
\partial^M \varphi(0) = \partial^M \varphi(0) = \partial^C \varphi(0) = \text{conv } \{g_1, g_2, g_3, g_4\}
\]

\[
\partial^K \varphi(0) = \partial^L \varphi(0) = \{g_1, g_2, g_3, g_4\}
\]
Relation to Other Derivative Concepts

Gradient Cube Example \((n = 2)\)

\[\varphi : \mathbb{R}^2 \mapsto \mathbb{R}, \quad \varphi(x_1, x_2) = \|x_2 - x_1\| + \varepsilon|x_1|, \quad \varepsilon \in \mathbb{R}\]

\[
= \begin{cases}
\varphi_1(x_1, x_2) = x_2 - x_1 + \varepsilon x_1 & \text{if } x_2 \geq x_1 \geq 0 \\
\varphi_2(x_1, x_2) = x_2 + x_1 - \varepsilon x_1 & \text{if } x_2 \geq -x_1, x_1 < 0 \\
\varphi_3(x_1, x_2) = -x_2 - x_1 - \varepsilon x_1 & \text{if } x_2 < -x_1, x_1 < 0 \\
\varphi_4(x_1, x_2) = -x_2 + x_1 + \varepsilon x_1 & \text{if } x_1 > x_2, x_1 \geq 0
\end{cases}
\]

If \(\varepsilon \geq 1\)

\[\hat{\partial}^M \varphi(0) = \partial^M \varphi(0) = \partial^C \varphi(0) = \text{conv} \{g_1, g_2, g_3, g_4\}\]

\[\partial^K \varphi(0) = \partial^L \varphi(0) = \{g_1, g_2, g_3, g_4\}\]

If \(\varepsilon < -1\)

\[\hat{\partial}^M \varphi(0) = \emptyset \quad \text{and} \quad \partial^M \varphi(0) = \text{conv} \{g_1, g_4\} \cup \text{conv} \{g_2, g_3\}\]

\[\partial^K \varphi(0) = \partial^L \varphi(0) = \{g_1, g_2, g_3, g_4\}\]
Gradient Cube Example \((n = 2)\)

\[ \varphi : \mathbb{R}^2 \mapsto \mathbb{R}, \quad \varphi(x_1, x_2) = |x_2 - |x_1|| + \varepsilon |x_1|, \quad \varepsilon \in \mathbb{R} \]

\[
\varphi = \begin{cases} 
\varphi_1(x_1, x_2) = x_2 - x_1 + \varepsilon x_1 & \text{if } x_2 \geq x_1 \geq 0 \\
\varphi_2(x_1, x_2) = x_2 + x_1 - \varepsilon x_1 & \text{if } x_2 \geq -x_1, x_1 < 0 \\
\varphi_3(x_1, x_2) = -x_2 - x_1 - \varepsilon x_1 & \text{if } x_2 < -x_1, x_1 < 0 \\
\varphi_4(x_1, x_2) = -x_2 + x_1 + \varepsilon x_1 & \text{if } x_1 > x_2, x_1 \geq 0 
\end{cases}
\]

If \(\varepsilon \geq 1\)

\[ \hat{\partial}^M \varphi(0) = \partial^M \varphi(0) = \partial^C \varphi(0) = \text{conv} \{g_1, g_2, g_3, g_4\} \]
\[ \partial^K \varphi(0) = \partial^L \varphi(0) = \{g_1, g_2, g_3, g_4\} \]

If \(\varepsilon < -1\)

\[ \hat{\partial}^M \varphi(0) = \emptyset \quad \text{and} \quad \partial^M \varphi(0) = \text{conv} \{g_1, g_4\} \cup \text{conv} \{g_2, g_3\} \]
\[ \partial^K \varphi(0) = \partial^L \varphi(0) = \{g_1, g_2, g_3, g_4\} \]

Relations of different generalized gradients?
Mangasarin-Fromovitz-Kink Qualification

Definition

The Mangasarin-Fromovitz-Kink Qualification (MFKQ) holds at a point $\hat{x}$ if

- for all $\sigma \succeq \sigma$ the vector inequality $J_\sigma v > 0$ is solvable for some $v \in \mathbb{R}^n$, where
  $$\sigma \succeq \sigma \quad \text{in that} \quad \sigma_j \sigma_j \geq \sigma_j^2 \quad \text{for} \quad j = 1, \ldots, n$$
  
  and $J_\sigma \equiv (\sigma_i \nabla z^\sigma_i)_{i \in \bar{\alpha}}$, or

- if $J_\sigma v \geq 0$ has only the trivial solution $v = 0 \in \mathbb{R}^n$
Mangasarin-Fromovitz-Kink Qualification

**Definition**

The Mangasarin-Fromovitz-Kink Qualification (MFKQ) holds at a point \( \hat{x} \) if

- for all \( \sigma \succeq \hat{\sigma} \) the vector inequality \( J_\sigma v > 0 \) is solvable for some \( v \in \mathbb{R}^n \), where
  
  \[
  \sigma \succeq \hat{\sigma} \quad \text{in that} \quad \sigma_j \hat{\sigma}_j \geq \hat{\sigma}_j^2 \quad \text{for} \quad j = 1, \ldots, n
  \]

  and \( J_\sigma \equiv (\sigma_i \nabla z^\sigma_i)_{i \in \alpha} \), or

- if \( J_\sigma v \geq 0 \) has only the trivial solution \( v = 0 \in \mathbb{R}^n \)

- strongly related to constraint qualification MFCQ

- much weaker than LIKQ
Kink Qualifikations for the Examples

One can check quite easily:

- Half-Pipe example:
  LIKQ and MFQK do not hold
Kink Qualifikations for the Examples

One can check quite easily:

- Half-Pipe example:
  LIKQ and MFQK do not hold

- Gradient Cube example:
  LIKQ and MFQK do hold
Kink Qualifikations for the Examples

One can check quite easily:

- Half-Pipe example:
  LIKQ and MFQK do not hold
- Gradient Cube example:
  LIKQ and MFQK do hold
- Lemon squeezer example:
Kink Qualifikations for the Examples

One can check quite easily:

- Half-Pipe example:
 LIKQ and MFQK do not hold

- Gradient Cube example:
 LIKQ and MFQK do hold

- Lemon squeezer example:
 LIKQ does not hold
  MFQK does hold

What can we prove with these properties?
Proposition (Limiting, Mordukovich and Clark subdiff’tials)

For the abs-normal function \( \varphi : \mathbb{R}^n \to \mathbb{R} \) and \( x \in \mathbb{R}^n \), the inclusions

\[
\partial^L \varphi(x) \subset \partial_M \varphi(x) \subset \partial_C \varphi(x)
\]

hold. Furthermore, the function \( \varphi(.) \) is regular in \( \hat{x} \in \mathbb{R}^n \) if and only if

\[
\partial_M \varphi(\hat{x}) = \partial_C \varphi(\hat{x})
\]
Proposition (Limiting, Mordukovich and Clark subdiff’tials)

For the abs-normal function \( \varphi : \mathbb{R}^n \to \mathbb{R} \) and \( x \in \mathbb{R}^n \), the inclusions

\[
\partial^L \varphi(x) \subset \partial_M \varphi(x) \subset \partial_C \varphi(x)
\]

hold. Furthermore, the function \( \varphi(.) \) is regular in \( \hat{x} \in \mathbb{R}^n \) if and only if

\[
\partial_M \varphi(\hat{x}) = \partial_C \varphi(\hat{x})
\]

Proposition (Conical and limiting gradients)

For the abs-normal function \( \varphi : \mathbb{R}^n \to \mathbb{R} \), one has

\[
\partial^K \varphi(x) \subset \partial^L \varphi(x)
\]

for all \( x \in \mathbb{R}^n \). Furthermore, if MFKQ holds at \( \hat{x} \in \mathbb{R}^n \), then

\[
\partial^K \varphi(\hat{x}) = \partial^L \varphi(\hat{x}).
\]
**Definition (First order convexity (FOC))**

The PS function \( \varphi : \mathbb{R}^n \rightarrow \mathbb{R} \) is said to be *convex of first order* at a point \( \hat{x} \) if its piecewise linearization \( \Delta \varphi(\hat{x}; \cdot) : \mathbb{R}^n \rightarrow \mathbb{R} \) is convex on some ball about the argument \( \Delta x = 0 \).
**Definition (First order convexity (FOC))**

The PS function $\varphi : \mathbb{R}^n \to \mathbb{R}$ is said to be *convex of first order* at a point $\hat{x}$ if its piecewise linearization $\Delta \varphi(\hat{x}; \cdot) : \mathbb{R}^n \to \mathbb{R}$ is convex on some ball about the argument $\Delta x = 0$.

**Theorem (Regularity and FOC)**

*For the abs-normal function $\varphi : \mathbb{R}^n \to \mathbb{R}$, one has that $\varphi(\cdot)$ is first order convex in some ball about $x$ if $\varphi(\cdot)$ is regular in $x$. Furthermore, if MFKQ holds at $x \in \mathbb{R}^n$, then $\varphi(\cdot)$ is first-order convex in some ball about $x$ if and only if $\varphi(\cdot)$ is regular in $x$.***
**Definition (First order convexity (FOC))**

The PS function \( \varphi : \mathbb{R}^n \to \mathbb{R} \) is said to be *convex of first order* at a point \( \hat{x} \) if its piecewise linearization \( \Delta \varphi(\hat{x}; \cdot) : \mathbb{R}^n \to \mathbb{R} \) is convex on some ball about the argument \( \Delta x = 0 \).

**Theorem (Regularity and FOC)**

*For the abs-normal function \( \varphi : \mathbb{R}^n \to \mathbb{R} \), one has that \( \varphi(.) \) is first order convex in some ball about \( x \) if \( \varphi(.) \) is regular in \( x \). Furthermore, if MFKQ holds at \( x \in \mathbb{R}^n \), then \( \varphi(.) \) is first-order convex in some ball about \( x \) if and only if \( \varphi(.) \) is regular in \( x \).*

**Theorem (Complexity of convexity test)**

*The convexity test is co-NP complete.*
**Definition (First order convexity (FOC))**

The PS function $\varphi : \mathbb{R}^n \to \mathbb{R}$ is said to be *convex of first order* at a point $\hat{x}$ if its piecewise linearization $\Delta \varphi(\hat{x}; \cdot) : \mathbb{R}^n \to \mathbb{R}$ is convex on some ball about the argument $\Delta x = 0$.

**Theorem (Regularity and FOC)**

*For the abs-normal function $\varphi : \mathbb{R}^n \to \mathbb{R}$, one has that $\varphi(.)$ is first order convex in some ball about $x$ if $\varphi(.)$ is regular in $x$. Furthermore, if MFKQ holds at $x \in \mathbb{R}^n$, then $\varphi(.)$ is first-order convex in some ball about $x$ if and only if $\varphi(.)$ is regular in $x$.***

**Theorem (Complexity of convexity test)**

*The convexity test is co-NP complete.*

A. Walther, A. Griewank. Characterizing and testing subdifferential regularity for piecewise smooth objective functions, in revision
Conclusion and Outlook

- Practically all nonsmooth problems are piecewise smooth in abs-normal form.

- Extended tools for algorithmic differentiation yield abs-linearization in form of $Z, L, a, b, c$.
  \[ \Rightarrow \text{generation of abs-normal form can be automated!} \]
Conclusion and Outlook

- Practically all nonsmooth problems are piecewise smooth in abs-normal form.

- Extended tools for algorithmic differentiation yield abs-linearization in form of $Z$, $L$, $a$, $b$, $c$.
  $\Rightarrow$ generation of abs-normal form can be automated!

- LIKQ $\Rightarrow$ First order minimality can be tested with polynomial effort

- SALOP yields typically linear, quadratic, or superlinear convergence
  Inner loop: PL functions can be minimized effectively by adapted QP solver

- Wine quality test feasible for 3,000 out of 4,000 data sets
  $\Rightarrow$ training of model possible!
Conclusion and Outlook

- Practically all nonsmooth problems are piecewise smooth in abs-normal form.

- Extended tools for algorithmic differentiation yield abs-linearization in form of $Z, L, a, b, c$. 
  $\Rightarrow$ generation of abs-normal form can be automated!

- LIKQ $\Rightarrow$ First order minimality can be tested with polynomial effort

- SALOP yields typically linear, quadratic, or superlinear convergence
  Inner loop: PL functions can be minimized effectively by adapted QP solver

- Wine quality test feasible for 3 000 out of 4 000 data sets 
  $\Rightarrow$ training of model possible!

- Relation to other derivatives concepts analysed