A VU-Point of View of Nonsmooth Optimization

Claudia Sagastizábal
IMECC-UNICAMP BRAZIL, adjunct researcher
sagastiz@unicamp.br

ENSTA, January 14th and 15th, 2019

(supported by ENSTA, PGMO, and Fondation X)

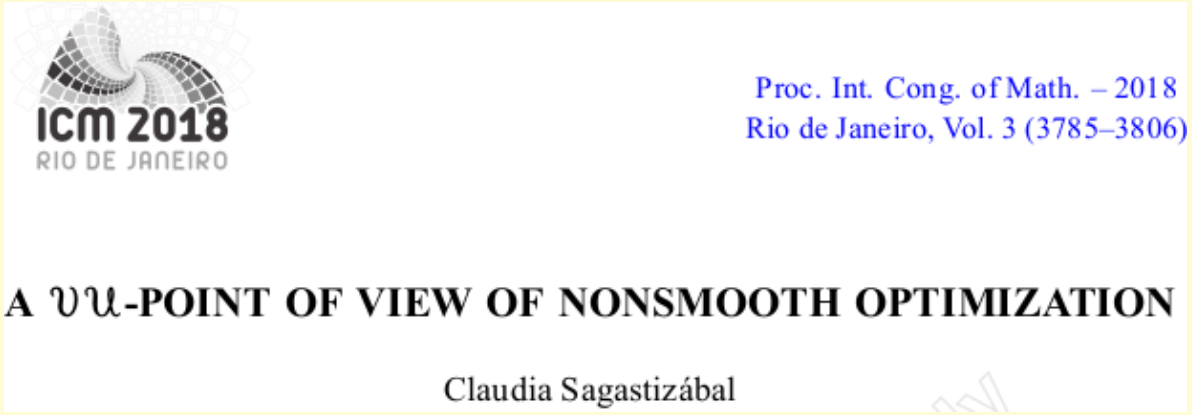
Program

1. Yesterday morning:
   Introduction to nonsmooth convex optimization
2. Yesterday afternoon:
   Models and the proximal point algorithm
3. Today morning:
   Bundle methods and the Moreau-Yosida regularization
4. Today afternoon:
   Beyond first order: VU-decomposition methods

Bibliography and one comment

References in the conference paper

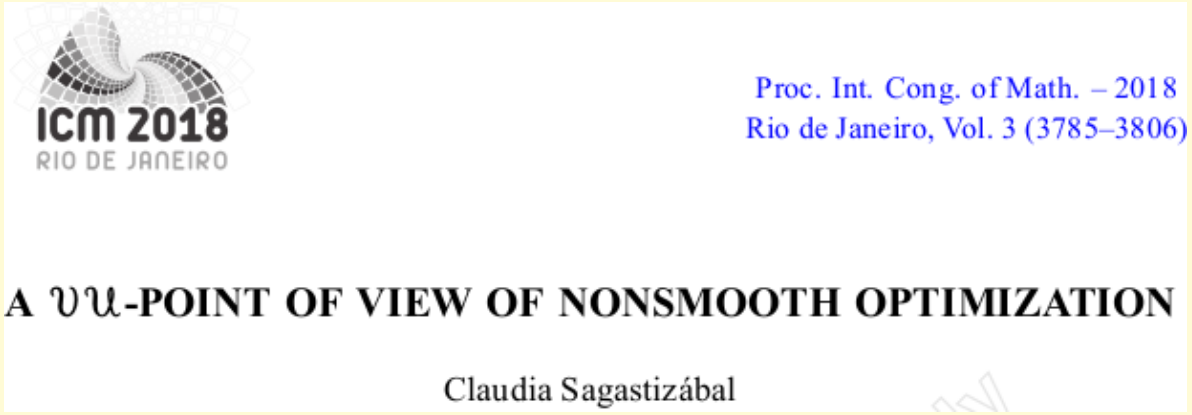
https://eta.impa.br/dl/064.pdf



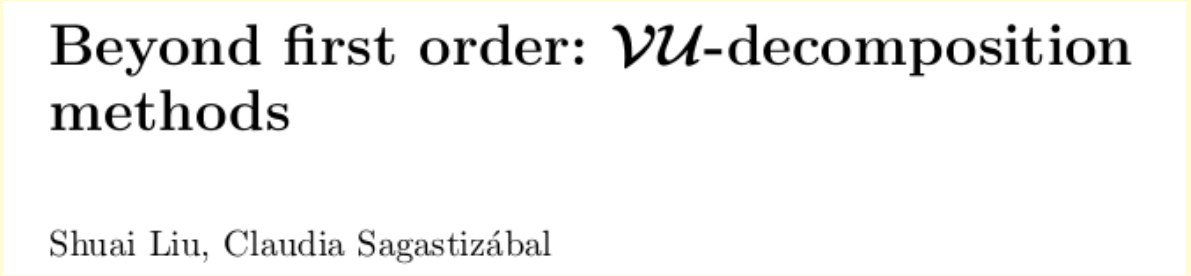
Bibliography and one comment

References in the conference paper

`https://eta.impa.br/dl/064.pdf`



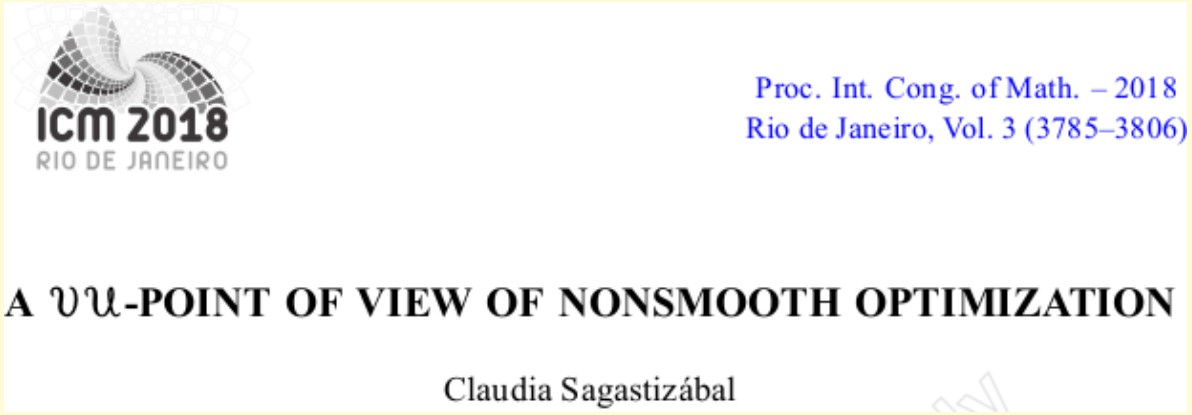
Also the forthcoming book chapter



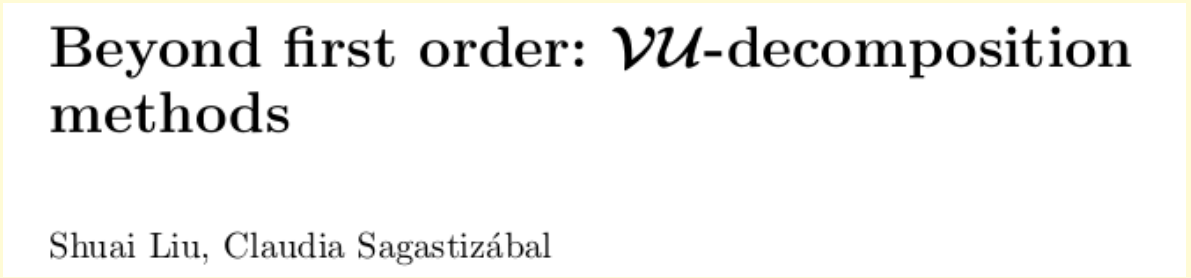
Bibliography and one comment

References in the conference paper

`https://eta.impa.br/dl/064.pdf`



Also the forthcoming book chapter



Beware that requiring the knowledge of the Lipschitz constant of a function is not far from require to know the full subdifferential

ε-subgradient descent methods

- **PPA**

$$x^{k+1} = p^f_{t_k}(x^k) = x^k - t_k G^{k+1} \text{ with } G^{k+1} \in \partial_{\varepsilon_{k+1}} f(x^k)$$

ε-subgradient descent methods

- **PPA**

$$x^{k+1} = p^f_{t_k}(x^k) = x^k - t_k G^{k+1} \text{ with } G^{k+1} \in \partial_{\varepsilon_{k+1}} f(x^k)$$

(DR) is satisfied at all iterations (no hats needed)

ε-subgradient descent methods

- **PPA**

$$x^{k+1} = p^f_{t_k}(x^k) = x^k - t_k G^{k+1} \text{ with } G^{k+1} \in \partial_{\varepsilon_{k+1}} f(x^k)$$

(DR) is satisfied at all iterations (no hats needed)

may not be implementable

- **Bundle method as inexact PPA**

$$x^{k+1} = p^{\mathbf{M}_k}_{t_k}(\hat{x}^k) = \hat{x}^k - t_k G^k \text{ with } G^k \in \partial_{\varepsilon_k} f(\hat{x}^k)$$

ε-subgradient descent methods

- **PPA**

$$x^{k+1} = p^f_{t_k}(x^k) = x^k - t_k G^{k+1} \text{ with } G^{k+1} \in \partial_{\varepsilon_{k+1}} f(x^k)$$

(DR) is satisfied at all iterations (no hats needed)

may not be implementable

- **Bundle method as inexact PPA**

$$x^{k+1} = p^{\mathbf{M}_k}_{t_k}(\hat{x}^k) = \hat{x}^k - t_k G^k \text{ with } G^k \in \partial_{\varepsilon_k} f(\hat{x}^k)$$

$\hat{x}^{k+1} = \hat{x}^k$ **whenever (DR) is satisfied**

Bundle methods

0 Choose x^1, $t_1 > 0$, and set $\hat{x}^1 = x^1$, $k = 1$.

1 Given $\hat{x}^k$, $\mathbf{M}_k$ and t_k, compute
$x^{k+1} = \arg\min \mathbf{M}_k(x) + \frac{1}{2t_k}\|x - \hat{x}^k\|^2$ and

$$\delta_{k+1} = f(\hat{x}^k) - \mathbf{M}_k(x^{k+1}) = \varepsilon^k + t_k\|G^k\|^2$$

2 Call the oracle at x^{k+1}. If $\delta_{k+1} \leq tol$ **STOP**

3 `(Descent Rule)`

$$f(x^{k+1}) \leq f(\hat{x}^k) - m\delta_k? \quad \begin{cases} \text{yes} & \text{SS: } \hat{x}^{k+1} = x^{k+1} \\ \text{no} & \text{NS: } \hat{x}^{k+1} = \hat{x}^k \end{cases}$$

4 Choose a new model : $f(\cdot) \geq \mathbf{M}_{k+1}(\cdot)$ and $\mathbf{M}_{k+1}(\cdot) \geq \max\left(\mathbf{M}_k(x^{k+1}) + G^{k\top}(\cdot - x^{k+1}), f^{k+1} + g^{k+1\top}(\cdot - x^{k+1})\right)$
Choose a bounded below stepsize t_{k+1} that, if NS, is nondecreasing

5 Set $k = k + 1$, loop to 1.

Relatives in the family of bundle methods

- Proximal BM, parameter t_k
- Trust-region BM, parameter Δ_k
- Level BM, parameter ℓ_k

Relatives in the family of bundle methods

- Proximal BM, parameter t_k
- Trust-region BM, parameter Δ_k
- Level BM, parameter ℓ_k

Using `(DR)` and the theory in [CL93], we showed convergence for the proximal family

Relatives in the family of bundle methods

- Proximal BM, parameter t_k
- Trust-region BM, parameter Δ_k
- Level BM, parameter ℓ_k

Using `(DR)` and the theory in [CL93], we showed convergence for the proximal family

What about speed?

Math. Program., Ser. A 86: 41–50 (1999)

Digital Object Identifier (DOI) 10.1007/s101079900088

Stephen M. Robinson

Linear convergence of epsilon-subgradient descent methods for a class of convex functions*

Math. Program., Ser. A 86: 41–50 (1999) Springer-Verlag 1999

Digital Object Identifier (DOI) 10.1007/s101079900088

Stephen M. Robinson

Linear convergence of epsilon-subgradient descent methods for a class of convex functions*

Consider a method defining iterates

$x^{k+1} = x^k - t_k g^k$ for $g^k \in \partial_\varepsilon f(x^k)$ and such that

- (DR) holds with $\delta_k = \varepsilon + t_k \|g^k\|^2$
- $t_k \in [t_{\min}, t_{\max}]$ with $0 < t_{\min} \leq t_{\max} < \infty$

Math. Program., Ser. A 86: 41–50 (1999) Springer-Verlag 1999

Digital Object Identifier (DOI) 10.1007/s101079900088

Stephen M. Robinson

Linear convergence of epsilon-subgradient descent methods for a class of convex functions*

Consider a method defining iterates

$$x^{k+1} = x^k - t_k g^k \quad \text{for } g^k \in \partial_\varepsilon f(x^k) \text{ and such that}$$

- (DR) holds with $\delta_k = \varepsilon + t_k \|g^k\|^2$
- $t_k \in [t_{\min}, t_{\max}]$ with $0 < t_{\min} \leq t_{\max} < \infty$

If f is not too "flat" around its set of minimizers $\bar{X}$,

$$f(x) \geq \inf f + c\,dist(x, \bar{X}) \text{ for any } x \in \bar{X} + \eta\mathbb{B},$$

Math. Program., Ser. A 86: 41–50 (1999) Springer-Verlag 1999

Digital Object Identifier (DOI) 10.1007/s101079900088

Stephen M. Robinson

Linear convergence of epsilon-subgradient descent methods for a class of convex functions*

Consider a method defining iterates

$$x^{k+1} = x^k - t_k g^k \quad \text{for } g^k \in \partial_\varepsilon f(x^k) \text{ and such that}$$

- (DR) holds with $\delta_k = \varepsilon + t_k \|g^k\|^2$
- $t_k \in [t_{\min}, t_{\max}]$ with $0 < t_{\min} \leq t_{\max} < \infty$

If f is not too "flat" around its set of minimizers $\bar{X}$,

$$f(x) \geq \inf f + c\,dist(x, \bar{X}) \text{ for any } x \in \bar{X} + \eta\mathbb{B},$$

the rate of convergence if R-linear

Math. Program., Ser. A 86: 41–50 (1999) Springer-Verlag 1999

Digital Object Identifier (DOI) 10.1007/s101079900088

Stephen M. Robinson

Linear convergence of epsilon-subgradient descent methods for a class of convex functions*

Consider a method defining iterates

$$x^{k+1} = x^k - t_k g^k \quad \text{for } g^k \in \partial_\varepsilon f(x^k) \text{ and such that}$$

- (DR) holds with $\delta_k = \varepsilon + t_k \|g^k\|^2$
- $t_k \in [t_{\min}, t_{\max}]$ with $0 < t_{\min} \leq t_{\max} < \infty$

If f is not too "flat" around its set of minimizers $\bar{X}$,

$$f(x) \geq \inf f + c\,dist(x, \bar{X}) \text{ for any } x \in \bar{X} + \eta\mathbb{B},$$

the rate of convergence if R-linear

note: condition $\iff$ **"inverse growth condition"**
$\iff$ **Kurdyka-Lojasiewicz inequality**

Jean-Jacques Moreau and its envelope

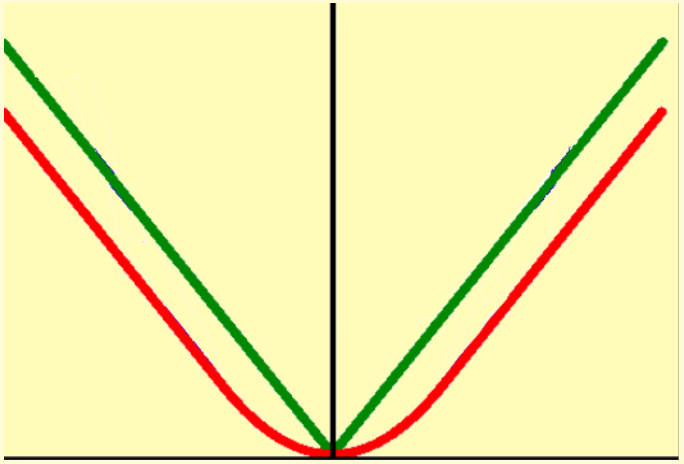
The Moreau-envelope of f is a $C^{1,1}$-smoothing of f

$$F_t(x) := \min\left\{ f(y) + \frac{1}{2t}\|y - x\|^2 \right\}$$

Jean-Jacques Moreau and its envelope

The Moreau-envelope of f is a $C^{1,1}$-smoothing of f

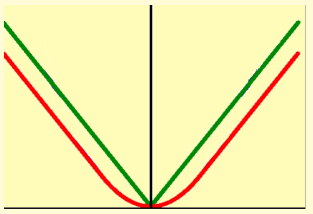
$$F_t(x) := \min\left\{ f(y) + \frac{1}{2t}\|y - x\|^2 \right\}$$



- the unique minimizer is the proximal point mapping $p_t^f(x)$

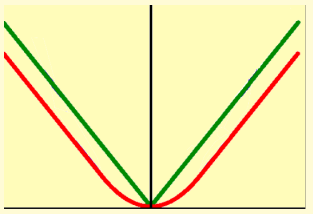
- the envelope's gradient is $\nabla F_t(x) = \frac{1}{t}\left(x - p_t^f(x)\right)$

Proximal point relations



$\min f(x)$ **is equivalent to** $x^* = p_t(x^*)$

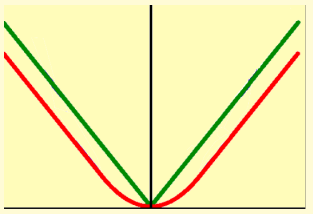
Proximal point relations



$$\min f(x) \quad \textbf{is equivalent to} \quad x^* = p_t(x^*)$$

Picard's iteration $\quad x^{k+1} = p_t(x^k)$

Proximal point relations

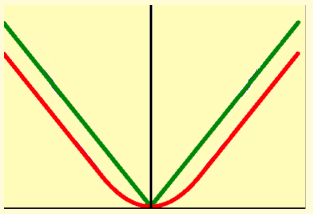


$$\min f(x) \quad \textbf{is equivalent to} \quad x^* = p_t(x^*)$$

Picard's iteration

$$\begin{aligned} x^{k+1} &= p_t(x^k) \\ &= x^k - \frac{t}{t}\left(x^k - p_t(x^k)\right) \end{aligned}$$

Proximal point relations

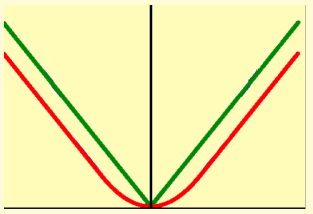


$$\min f(x) \quad \textbf{is equivalent to} \quad x^* = p_t(x^*)$$

Picard's iteration

$$\begin{aligned} x^{k+1} &= p_t(x^k) \\ &= x^k - \frac{t}{t}\left(x^k - p_t(x^k)\right) \\ &= x^k - t\nabla F_t(x^k) \end{aligned}$$

Proximal point relations



$$\min f(x) \quad \textbf{is equivalent to} \quad x^* = p_t(x^*)$$

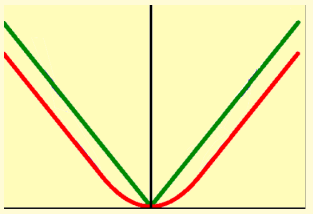
$$\textbf{is equivalent to} \quad \min F_t(x)$$

Picard's iteration

$$\begin{aligned} x^{k+1} &= p_t(x^k) \\ &= x^k - \frac{t}{t}\left(x^k - p_t(x^k)\right) \\ &= x^k - t\nabla F_t(x^k) \end{aligned}$$

a gradient method to minimize Moreau's envelope

Proximal point relations



$$\min f(x) \quad \textbf{is equivalent to} \quad x^* = p_t(x^*)$$

$$\textbf{is equivalent to} \quad \min F_t(x)$$

Picard's iteration

$$\begin{aligned} x^{k+1} &= p_t(x^k) \\ &= x^k - \frac{t}{t}\left(x^k - p_t(x^k)\right) \\ &= x^k - t\nabla F_t(x^k) \end{aligned}$$

a gradient method to minimize Moreau's envelope

note: the "stepsize" t is NOT computed by a linesearch

(curve-search on the metric)

Generalizing Moreau's envelope

Replace the proximal parameter by a *matrix*:

Instead of $\frac{1}{t}$

Generalizing Moreau's envelope

Replace the proximal parameter by a *matrix*:

Instead of $\frac{1}{t}$ **use a positive definite matrix M**:

$$F_M(x) := \min\left\{ f(y) + \frac{1}{2}\|y - x\|_M^2 \right\}$$

Generalizing Moreau's envelope

Replace the proximal parameter by a *matrix*:

Instead of $\frac{1}{t}$ **use a positive definite matrix M**:

$$F_M(x) := \min\left\{ f(y) + \frac{1}{2}\|y - x\|_M^2 \right\}$$

- There is a unique minimizer $p_M(x)$
- The proximal point operator is Lipschitzian
- The relation $\nabla F_M(x) = M(x - p_M(x))$ holds
- $\min f \iff \min F_M \iff x = p_M(x)$

Generalizing Moreau's envelope

Replace the proximal parameter by a *matrix*:

Instead of $\frac{1}{t}$ use a positive definite matrix M:

$$F_M(x) := \min\left\{ f(y) + \frac{1}{2}\|y - x\|_M^2 \right\}$$

- There is a unique minimizer $p_M(x)$
- The proximal point operator is Lipschitzian
- The relation $\nabla F_M(x) = M(x - p_M(x))$ holds
- $\min f \iff \min F_M \iff x = p_M(x)$

Interest: now we perceive better the role of M

The role of the metric

Picard's iteration

$$\begin{aligned} x^{k+1} &= p_M(x^k) \\ &= x^k - M^{-1} M\Big(x^k - p_t(x^k)\Big) \\ &= x^k - M^{-1}\nabla F_M(x^k) \end{aligned}$$

a gradient method to minimize Moreau's envelope, preconditioned by the matrix $M^{-1} \approx \nabla^{-2} F_M(x^k)$

The role of the metric

Picard's iteration

$$\begin{aligned} x^{k+1} &= p_M(x^k) \\ &= x^k - M^{-1} M\left(x^k - p_t(x^k)\right) \\ &= x^k - M^{-1}\nabla F_M(x^k) \end{aligned}$$

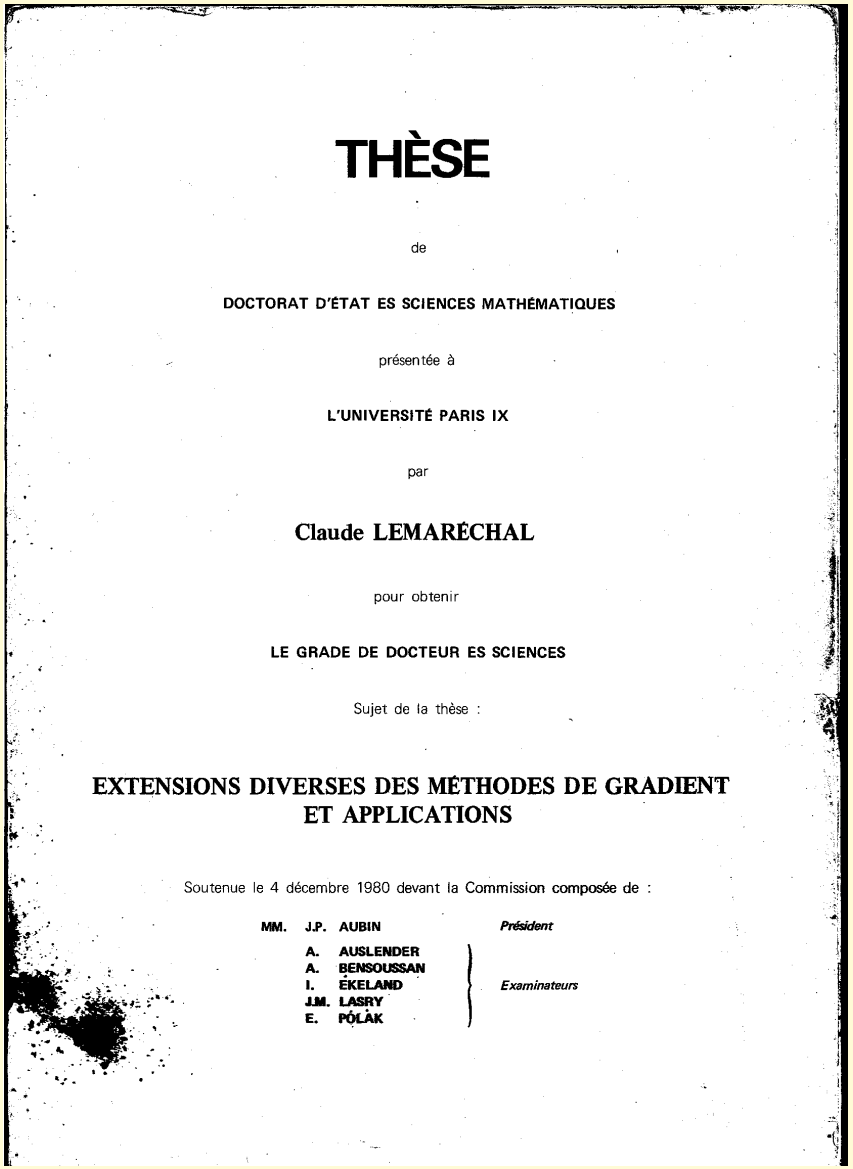
a gradient method to minimize Moreau's envelope, preconditioned by the matrix $M^{-1} \approx \nabla^{-2} F_M(x^k)$

We know $F_M \in C^{1,1}$ . . . when does the Hessian exist?

Existe-t-il une généralisation adéquate de la notion de Hessien?. . . Cette question est la plus passionnante qui se pose actuellement, et une réponse satisfaisante marquerait probablement pour longtemps une étape décisive dans les recherches fondamentales en programmation mathématique.

“Does an adequate generalization for the notion of a Hessian exist?. . . This is today’s most interesting question, to which a satisfactory answer would probably start a new, long-lasting and decisive era for basic research in Mathematical Programming.”

Chapter 5 in:



THÈSE

de

DOCTORAT D'ÉTAT ES SCIENCES MATHÉMATIQUES

présentée à

L'UNIVERSITÉ PARIS IX

par

Claude LEMARÉCHAL

pour obtenir

LE GRADE DE DOCTEUR ES SCIENCES

Sujet de la thèse :

EXTENSIONS DIVERSES DES MÉTHODES DE GRADIENT ET APPLICATIONS

Soutenue le 4 décembre 1980 devant la Commission composée de :

MM.	J.P.	AUBIN	*Président*
	A.	AUSLENDER	*Examinateurs*
	A.	BENSOUSSAN	
	I.	EKELAND	
	J.M.	LASRY	
	E.	PÓLAK	

December 4th, 1980

How structural nonsmoothness has been exploited

Improving speed only possible if algorithm incorporates **structure** information

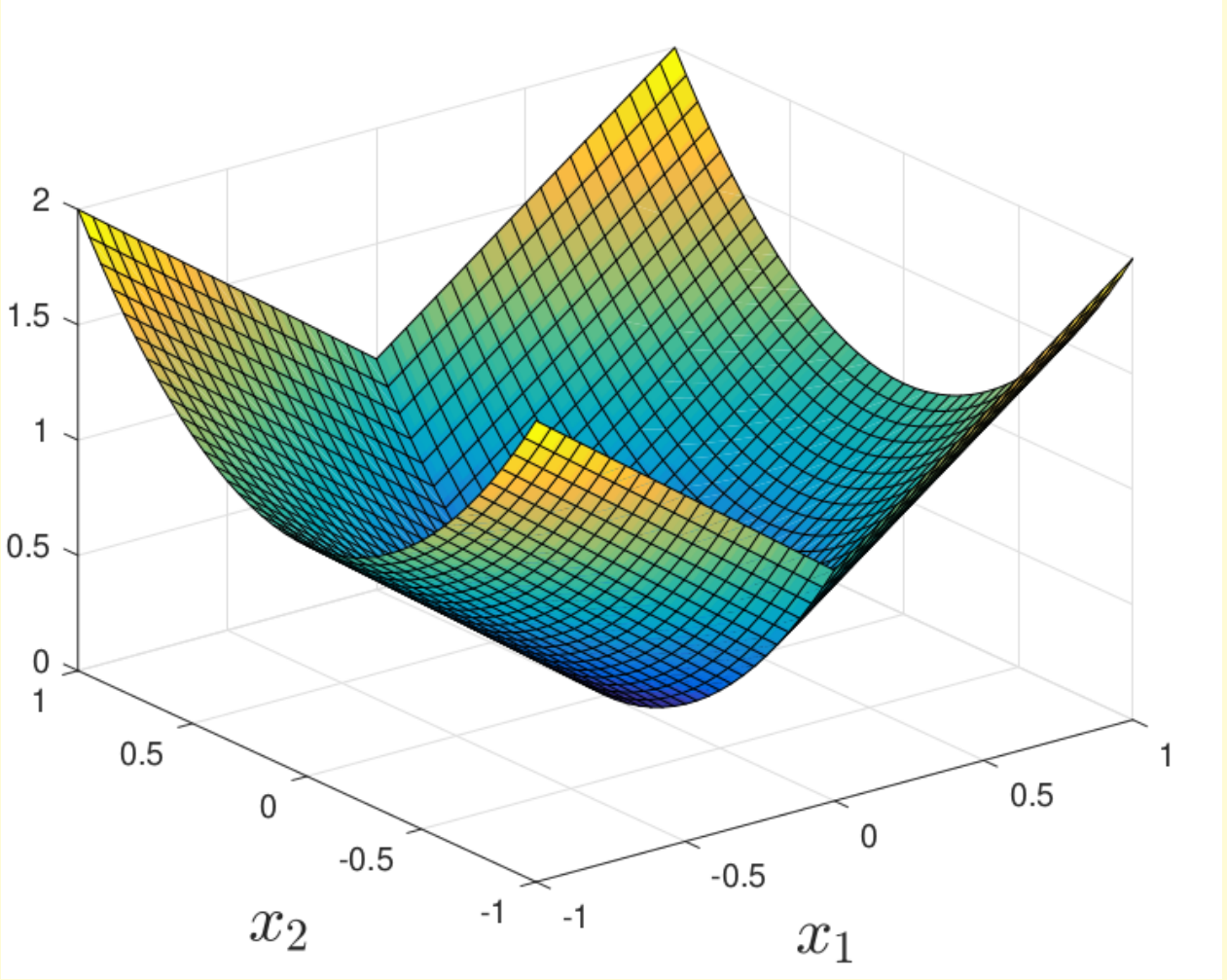
- 1995-2000: $\mathcal{U}$-Lagrangian (Lemaréchal, Oustry, Sagastiz)
- 1999: $\mathcal{VU}$-decomposition (Mifflin, Sagastiz)
- 2002: $\mathcal{M}$-manifolds, partly smooth functions (Lewis, Hare)
- 2003: Composite objective functions (Shapiro)

Later on, special minimization of composite objective functions revisited: Lewis & Wright, Nesterov, Planiden, Hare & Sagastiz, Liu, Sagastiz & Solodov

Illustrative examples

- The half-and-half function in $\mathbb{R}^2$

  $f(x_1, x_2) = |x_1| + bx_2^2$



Proximal point: calculus rules

- separable sum:
  $f(x,y) = g(x) + h(y) \Longrightarrow$
  $p_t^f(x) = \left(p_t^g(x), p_t^h(y)\right)$
- scalar factor ($\alpha \neq 0$) and translation ($v \neq 0$):
  $f(x) = g(\alpha x + v) \Longrightarrow$
  $p_t^f(x) = \frac{1}{\alpha}\left(p_t^{\alpha^2 g}(\alpha x + v) - v\right)$
- "perspective" ($\alpha > 0$):
  $f(x) = \alpha g(\frac{1}{\alpha}x) \Longrightarrow p_t^f(x) = \alpha p_t^{g/\alpha}(\frac{x}{\alpha})$

Proximal point: special functions

- + linear term ($v \neq 0$):
  $f(x) = g(x) + \langle v, x \rangle \Longrightarrow p_t^f(x) = p_t^g(x - v)$
- + convex quadratic term ($t > 0$):
  $f(x) = g(x) + \dfrac{1}{2t}\|x - v\|^2 \Longrightarrow$
  $p_t^f(x) = p_t^{\lambda g}(\lambda x + (1 - \lambda)v)$ for $\lambda = \dfrac{t}{t+1}$
- composition with linear term such that $A^\top A = \frac{1}{\alpha} I$, ($\alpha \neq 0$):
  $f(x) = g(Ax + v) \Longrightarrow$
  $p_t^f(x) = (I - \alpha A^\top A)x + \alpha A^\top \left[p_t^{g/\alpha}(Ax + v) - v \right]$

Proximal calculations: half-and-half function

Since $f(x_1, x_2) = |x_1| + \frac{b}{2}x_2^2$ is a separable sum, $f(x_1, x_2) = f_1(x_1) + f_2(x_2)$, the prox can be computed separately for each component:

$$p_t^f(x) = \left(p_t^{f_1}(x_1), p_t^{f_2}(x_2)\right)$$

Proximal calculations: half-and-half function

Since $f(x_1, x_2) = |x_1| + \frac{b}{2}x_2^2$ is a separable sum, $f(x_1, x_2) = f_1(x_1) + f_2(x_2)$, the prox can be computed separately for each component:

$$p_t^f(x) = \left(p_t^{f_1}(x_1), p_t^{f_2}(x_2)\right)$$

$$p_t^{f_1}(x_1) = x_1 - proj_{[-t,t]}(x_1)$$

Proximal calculations: half-and-half function

Since $f(x_1, x_2) = |x_1| + \frac{b}{2}x_2^2$ is a separable sum, $f(x_1, x_2) = f_1(x_1) + f_2(x_2)$, the prox can be computed separately for each component:

$$p_t^f(x) = \left(p_t^{f_1}(x_1), p_t^{f_2}(x_2)\right)$$

$$p_t^{f_1}(x_1) = x_1 - proj_{[-t,t]}(x_1)$$

$$p_t^{f_2}(x_2) = \frac{1}{1+bt}x_2$$

Proximal calculations: half-and-half function

Since $f(x_1, x_2) = |x_1| + \frac{b}{2}x_2^2$ is a separable sum, $f(x_1, x_2) = f_1(x_1) + f_2(x_2)$, the prox can be computed separately for each component:

$$p_t^f(x) = \left(p_t^{f_1}(x_1), p_t^{f_2}(x_2)\right)$$

$$p_t^{f_1}(x_1) = x_1 - proj_{[-t,t]}(x_1)$$

$$p_t^{f_2}(x_2) = \frac{1}{1+bt}x_2$$

$F_t(x) = ?$ **and** $\nabla F_t(x) = ?$

Proximal calculations: half-and-half function

Since $f(x_1, x_2) = |x_1| + \frac{b}{2}x_2^2$ is a separable sum, $f(x_1, x_2) = f_1(x_1) + f_2(x_2)$, the prox can be computed separately for each component:

$$p_t^f(x) = \left(p_t^{f_1}(x_1), p_t^{f_2}(x_2)\right)$$

$$p_t^{f_1}(x_1) = x_1 - proj_{[-t,t]}(x_1)$$

$$p_t^{f_2}(x_2) = \frac{1}{1+bt}x_2$$

$F_t(x) = ?$ **and** $\nabla F_t(x) = ?$

What about a Hessian?

quasi-Newton update of $\{M^k = \mu_k \mathrm{I}\}$

- Let M be the current matrix
- for example, let $u = \Delta x$, and $v = \Delta g$

quasi-Newton update of $\{M^k = \mu_k \mathrm{I}\}$

- Let M be the current matrix
- for example, let $u = \Delta x$, and $v = \Delta g$

The *quasi-Newton* equation

$M_+ u = v$ should hold for the update.

quasi-Newton update of $\{M^k = \mu_k \mathrm{I}\}$

- Let M be the current matrix
- for example, let $u = \Delta x$, and $v = \Delta g$

The *quasi-Newton* equation
$M_+ u = v$ should hold for the update.
If $M = \mu\,\mathrm{I}$, a scalar multiple of the identity satisfying the qN equation may be imposible, as it writes down

$$\mu_{k+1} u = v$$

with u and v vectors.
Instead, μ_{k+1} solves

$$\min_{\mu} \frac{1}{2} \|v/\mu - u\|^2$$

$\mu = ?$

quasi-Newton update of $\{M^k = \mu_k \mathrm{I}\}$

- Let M be the current matrix
- for example, let $u = \Delta x$, and $v = \Delta g \leftarrow$ **options!**

The *quasi-Newton* equation
$M_+ u = v$ should hold for the update.
If $M = \mu\,\mathrm{I}$, a scalar multiple of the identity satisfying the qN equation may be imposible, as it writes down

$$\mu_{k+1} u = v$$

with u and v vectors.
Instead, μ_{k+1} solves

$$\min_{\mu} \frac{1}{2} \|v/\mu - u\|^2$$

$\mu = ?$

Reversal quasi-Newton update

- Given u and Δg, invert the prox
- compute v accordingly

Reversal quasi-Newton update

- Given u and Δg, invert the prox
- compute v accordingly

$\mu = ?$

Reversal quasi-Newton update

- Given u and Δg, invert the prox
- compute v accordingly

$\mu = ?$ As in Lemaréchal C., Sagastizábal C. (1994) An approach to variable metric bundle methods. In: Henry J., Yvon JP. (eds) System Modelling and Optimization, LNCIS vol 197. Springer, Berlin, Heidelberg

$$\frac{1}{\mu_{n+1}} = \frac{1}{\mu_n} + \frac{\langle v, \Delta x \rangle}{|v|^2}$$

whenever $\langle v, u \rangle > 0$.

Theorem 10. *If ∇f is locally Lipschitzian, then $\mu_n \to 0$. Make the following additional assumptions: f has a (unique) minimal point $\bar{x}$ and a quadratic growth condition holds: for some $\alpha > 0$,*

$$f(x) \geq f(\bar{x}) + \alpha |x - \bar{x}|^2 .$$

Then $f(x_n)$ tends to $f(\bar{x})$ q-superlinearly.

How smooth is the regularization?

drop M in F_M

For Newton's-type methods to apply, we need F to have an invertible Hessian at $\bar{x}$

How smooth is the regularization?

drop M in F_M

For Newton's-type methods to apply, we need F to have an invertible Hessian at $\bar{x}$

A convex function φ has a *generalized Hessian* $H\varphi(x_0)$ at x_0 if the gradient $\nabla\varphi(x_0)$ exists and there exists a symmetric positive semidefinite operator $H\varphi(x_0)$ such that

$$\partial\varphi(x_0+d) \subset \nabla\varphi(x_0) + H\varphi(x_0)d + \|d\|\mathbb{B}$$

- If f at a generalized Hessian at $p(x_0)$, then the Hessian of F exists at x_0 and

$$\nabla^2 F(x_0) = M - M[Hf(p(x_0)) + M]^{-1}M$$

How smooth is the regularization?

drop M in F_M

For Newton's-type methods to apply, we need F to have an invertible Hessian at $\bar{x}$

How smooth is the regularization?

drop M in F_M

For Newton's-type methods to apply, we need F to have an invertible Hessian at $\bar{x}$

- We know that

  f convex and lower sci $\implies$ F is $C^{1,1}$ everywhere

How smooth is the regularization?

drop M in F_M

For Newton's-type methods to apply, we need F to have an invertible Hessian at $\bar{x}$

- We know that

  f convex and lower sci $\implies$ F is $C^{1,1}$ everywhere

- **BUT:** F is C^2 everywhere $\implies$ f is C^2 everywhere, too!

How smooth is the regularization?

drop M in F_M

For Newton's-type methods to apply, we need F to have an invertible Hessian at $\bar{x}$

- We know that

  f convex and lower sci $\implies$ F is $C^{1,1}$ everywhere

- **BUT:** F is C^2 everywhere $\implies$ f is C^2 everywhere, too!
- The Hessian of F exists at x_0 if and only if p has a Jacobian: $\nabla^2 F(x_0) = M(\mathrm{I} - p'(x_0))$.

How smooth is the regularization?

drop M in F_M

For Newton's-type methods to apply, we need F to have an invertible Hessian at $\bar{x}$

- We know that

  f convex and lower sci $\implies$ F is $C^{1,1}$ everywhere

- **BUT:** F is C^2 everywhere $\implies$ f is C^2 everywhere, too!
- The Hessian of F exists at x_0 if and only if p has a Jacobian: $\nabla^2 F(x_0) = M(\mathrm{I} - p'(x_0))$.
- When the Jacobian exists, its image lies in a very specific subspace

How smooth is the regularization?

drop M in F_M

For Newton's-type methods to apply, we need F to have an invertible Hessian at $\bar{x}$

- We know that

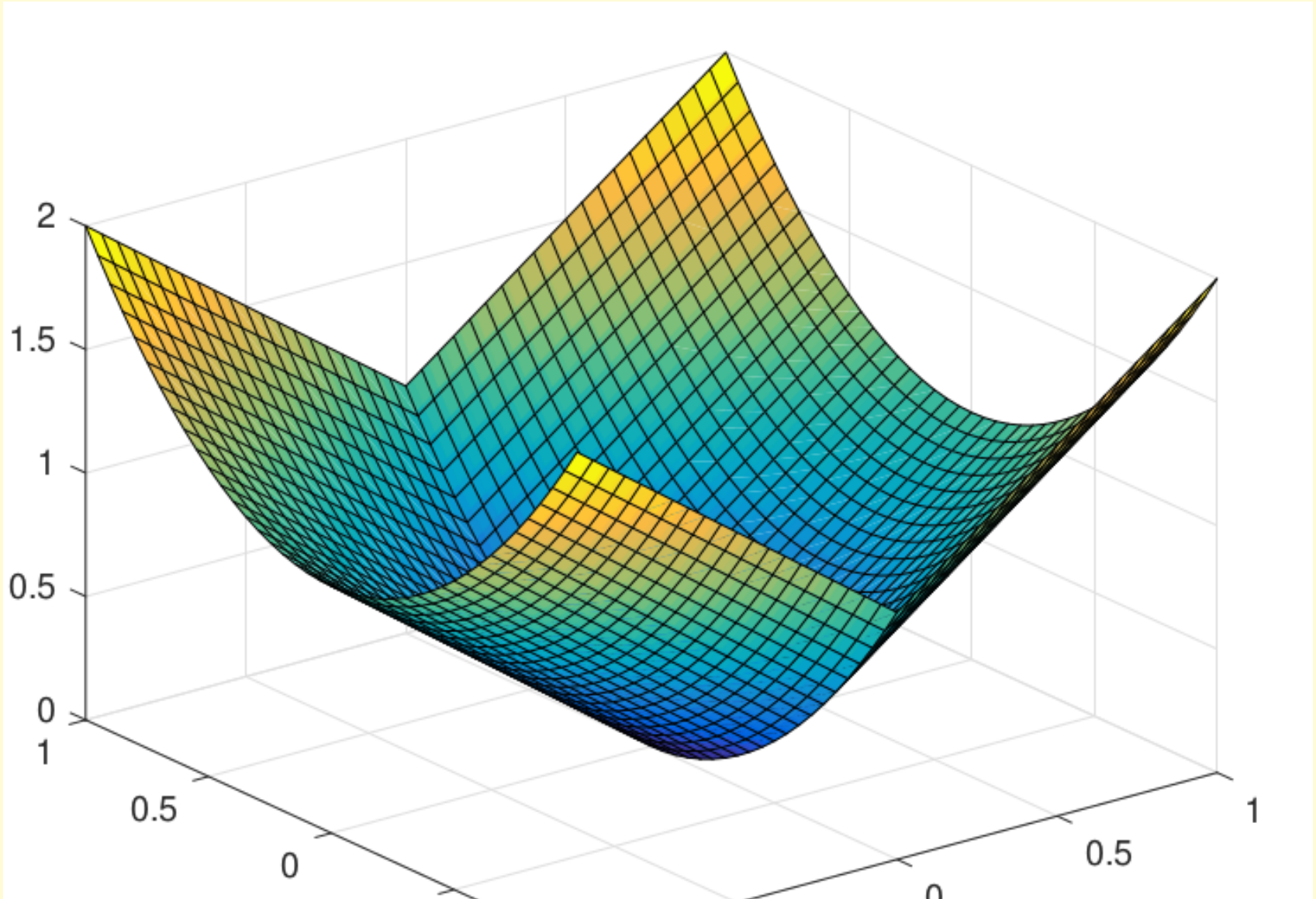
  f convex and lower sci $\implies$ F is $C^{1,1}$ everywhere

- **BUT:** F is C^2 everywhere $\implies$ f is C^2 everywhere, too!
- The Hessian of F exists at x_0 if and only if p has a Jacobian: $\nabla^2 F(x_0) = M(\mathrm{I} - p'(x_0))$.
- When the Jacobian exists, its image lies in a very specific subspace
- By considering trajectories not far from that subspace we can define a 2nd-order object of f, even if f is not differentiable

Views of some functions

The half-and-half

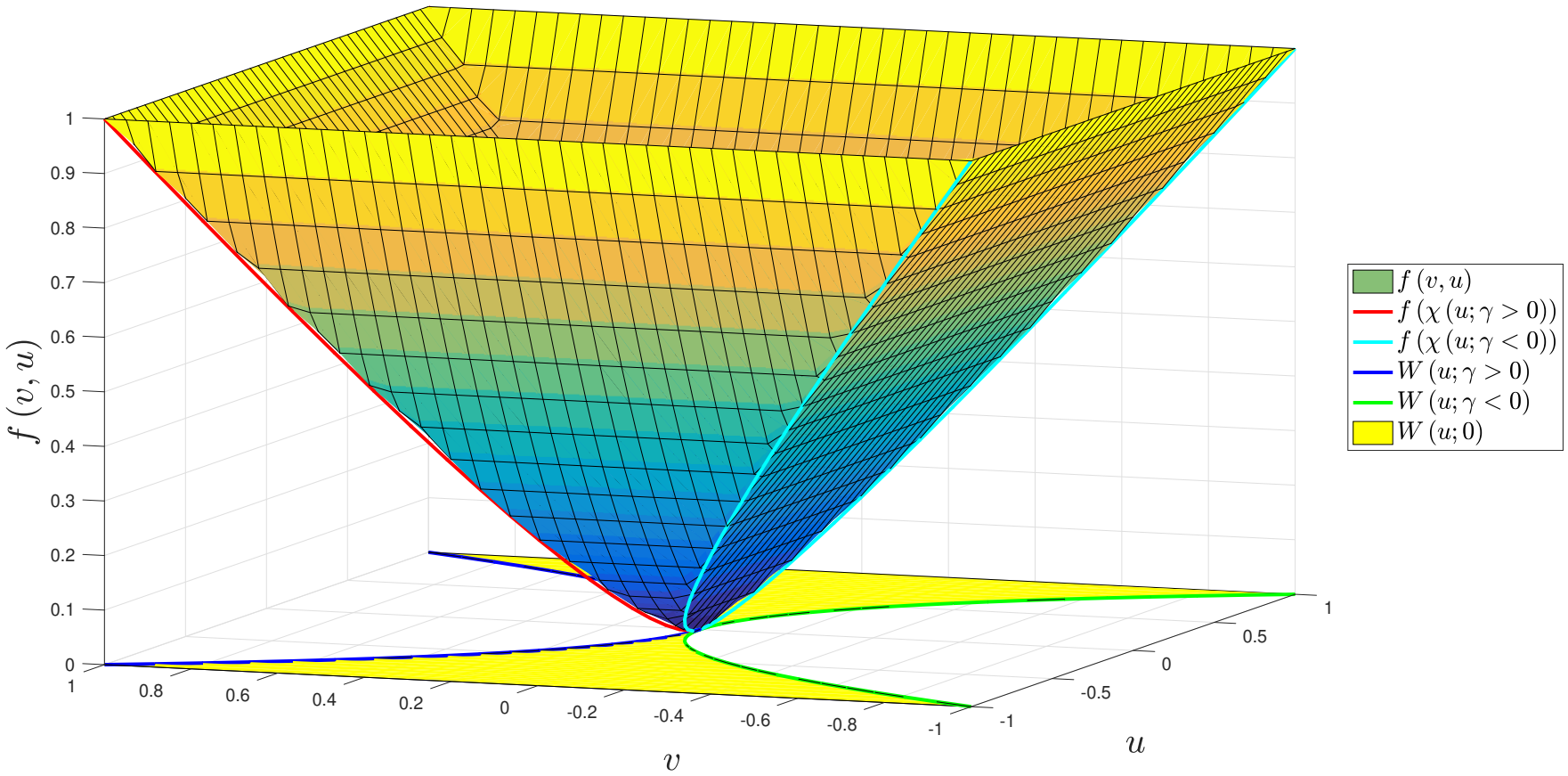
$f(x_1, x_2) = |x_1| + bx_2^2$



Views of some functions

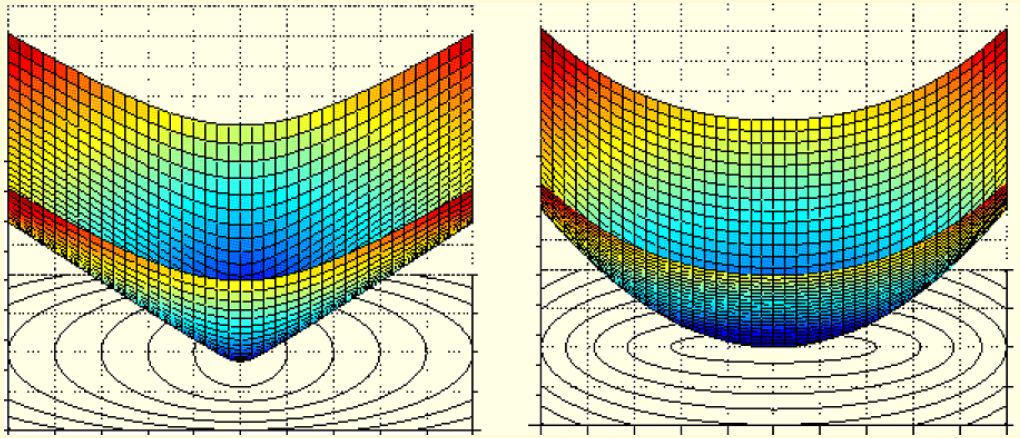
A max-variation

$f(x_1, x_2) = \max(|x_1|, bx_2^2)$



The key is in decomposing the space

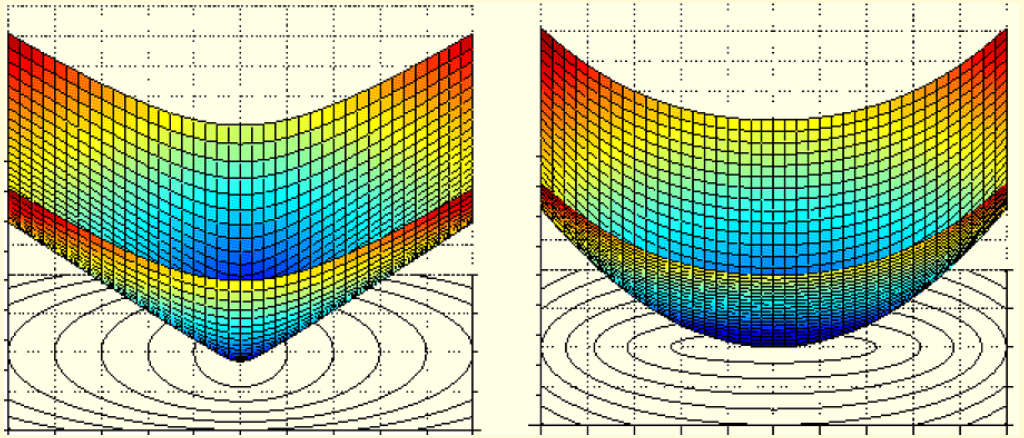
- $\mathbb{R}^n = \mathcal{V} \oplus \mathcal{U}$
- On the $\mathcal{V}$-subspace the function is sharp
- On the $\mathcal{U}$-subspace the function looks smooth



The subspace definition depends on x

The key is in decomposing the space

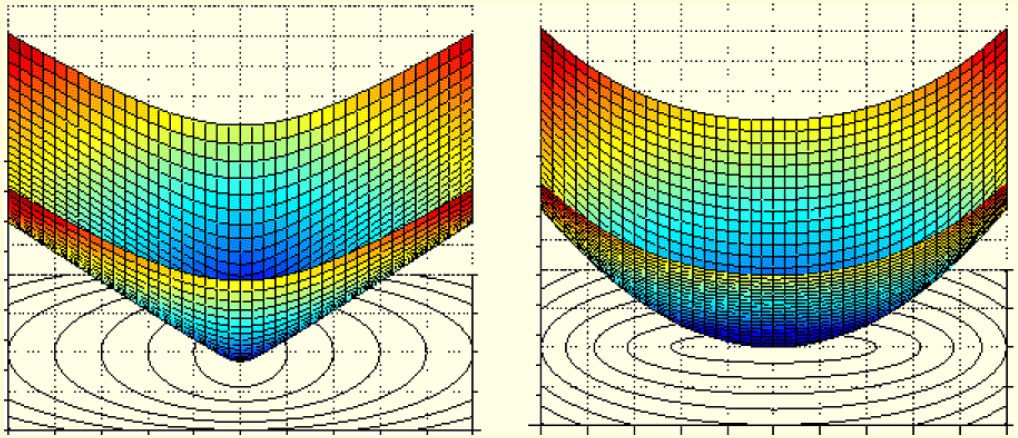
- $\mathbb{R}^n = \mathcal{V} \oplus \mathcal{U} + \bar{x}$
- On the $\mathcal{V}$-subspace the function is sharp
- On the $\mathcal{U}$-subspace the function looks smooth



The subspace definition depends on $\bar{x}$

The key is in decomposing the space

- $\mathbb{R}^n = \mathcal{V} \oplus \mathcal{U} + \bar{x}$
- On the $\mathcal{V}$-subspace the function is sharp
- On the $\mathcal{U}$-subspace the function looks smooth



The subspace definition depends on $\bar{x}$

Nonsmoothness appears in a structured manner

The n-dimensional half-and-half function

For $x \in \mathbb{R}^n$, given matrices A with nontrivial kernel, $B \succ 0$,

$$f(x) = \max\left(\left|x^\top A x\right|, x^\top B x\right)$$

has a unique minimizer at $\bar{x}* = 0$.

The n-dimensional half-and-half function

For $x \in \mathbb{R}^n$, given matrices A with nontrivial kernel, $B \succ 0$,

$$f(x) = \max\left(\left|x^\top A x\right|, x^\top B x\right)$$

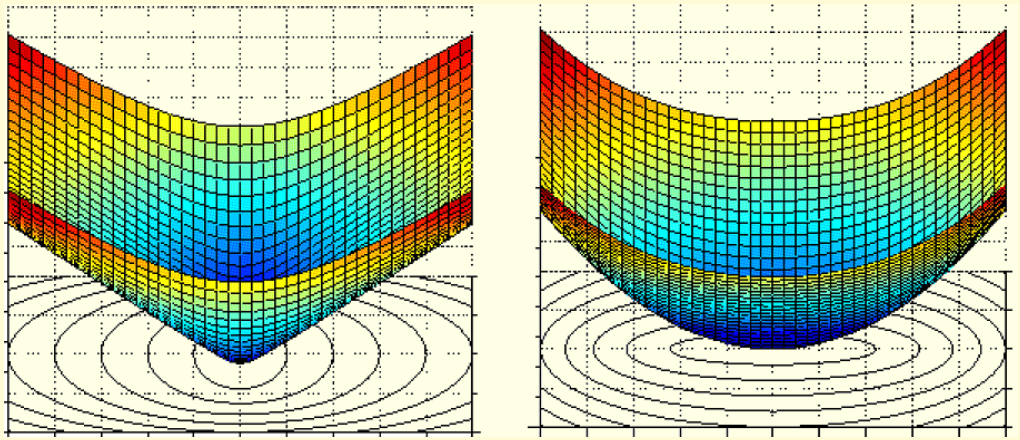
has a unique minimizer at $\bar{x}* = 0$. On $\mathcal{N}(A)$ the function is not differentiable, and the first term vanishes: $f|_{\mathcal{N}(A)}$ looks smooth

The n-dimensional half-and-half function

For $x \in \mathbb{R}^n$, given matrices A with nontrivial kernel, $B \succ 0$,

$$f(x) = \max\left(\left|x^\top A x\right|, x^\top B x\right)$$

has a unique minimizer at $\bar{x}* = 0$. On $\mathcal{N}(A)$ the function is not differentiable, and the first term vanishes: $f|_{\mathcal{N}(A)}$ looks smooth

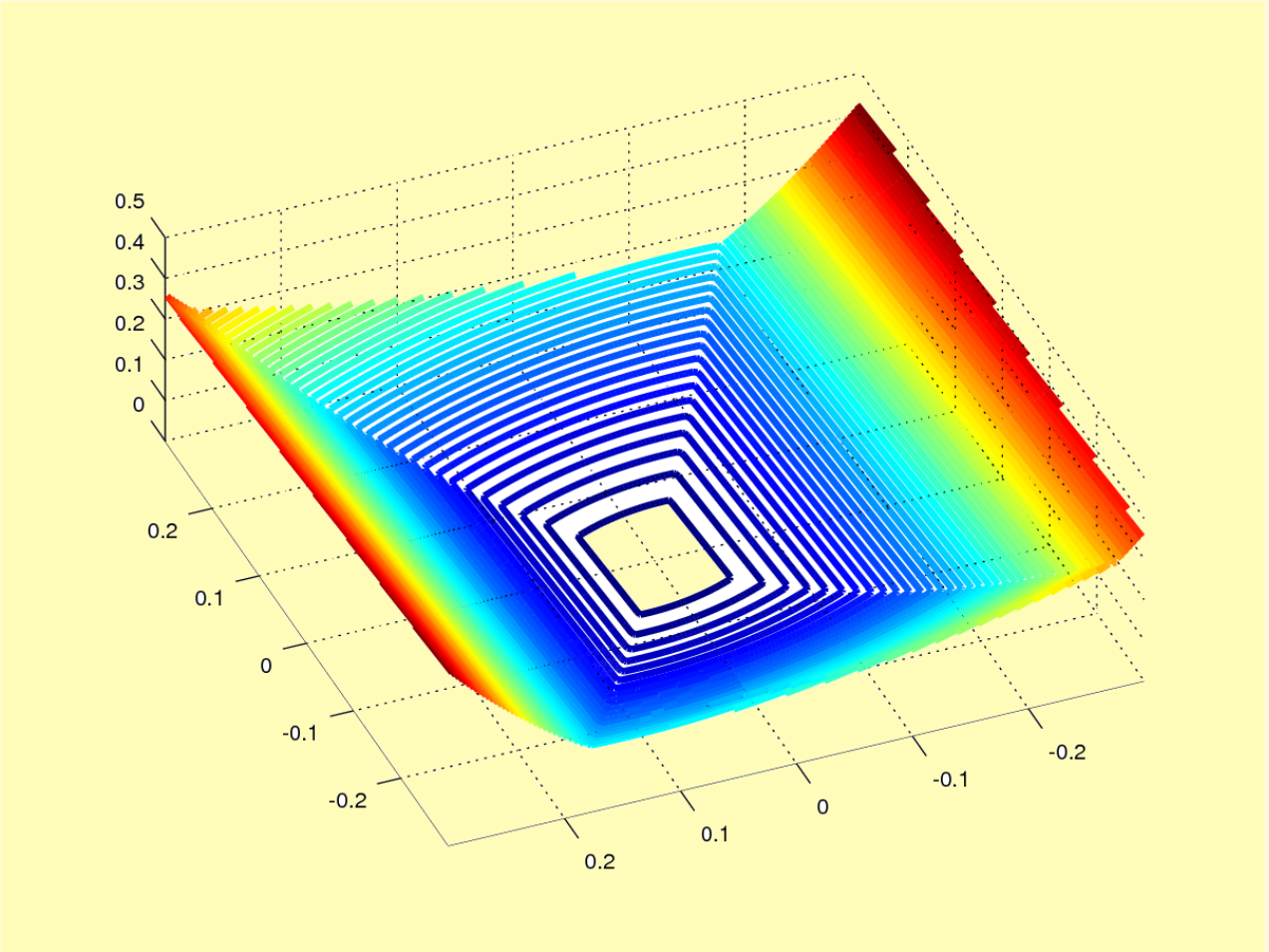


$\mathcal{R}(A)$ $\qquad$ $\mathcal{N}(A)$

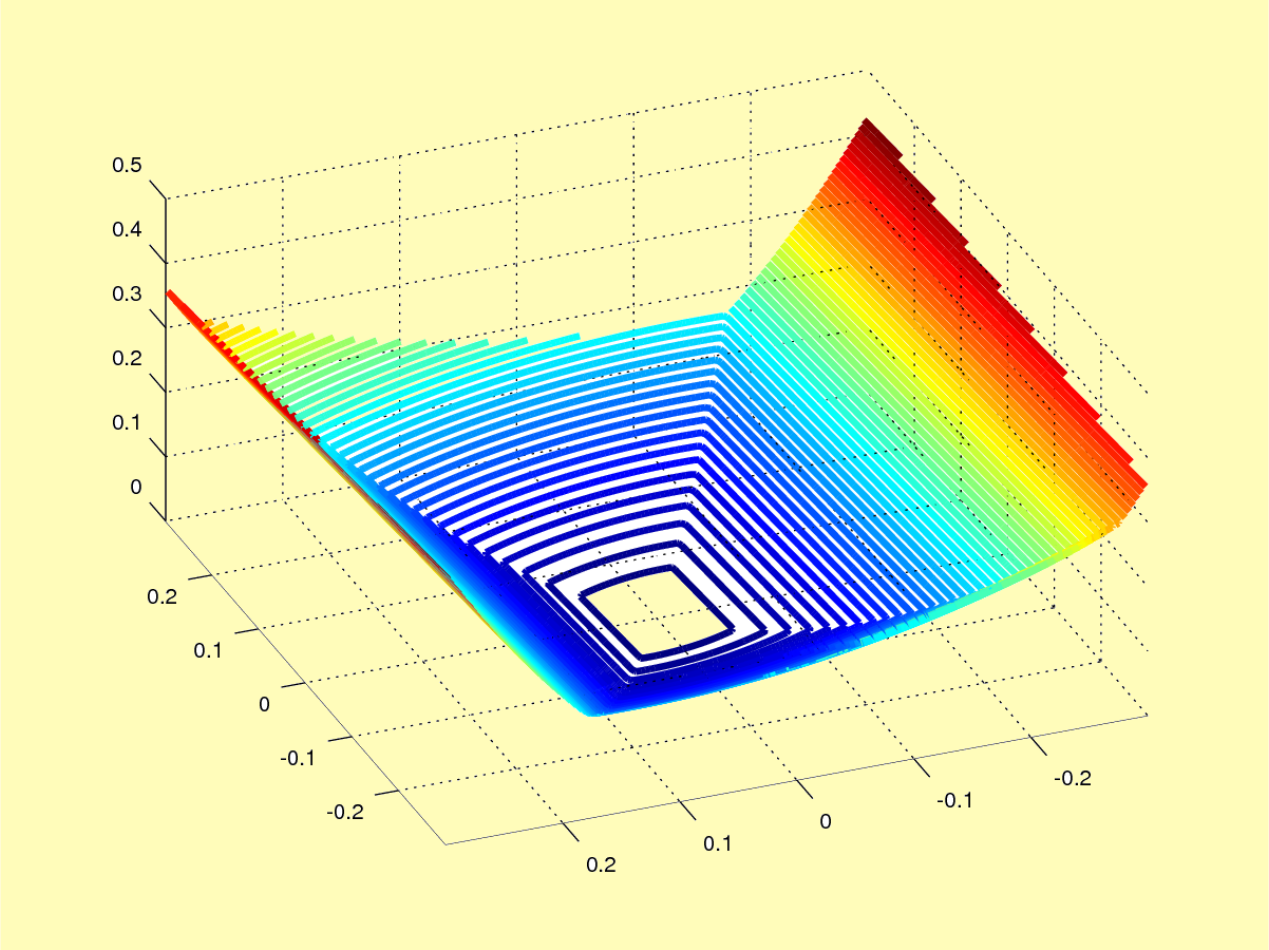
$\mathcal{V}$ parallel to $\partial f(\bar{x})$ $\qquad$ $\mathcal{U} \perp \mathcal{V}$

$\mathcal{VU}$s of the function

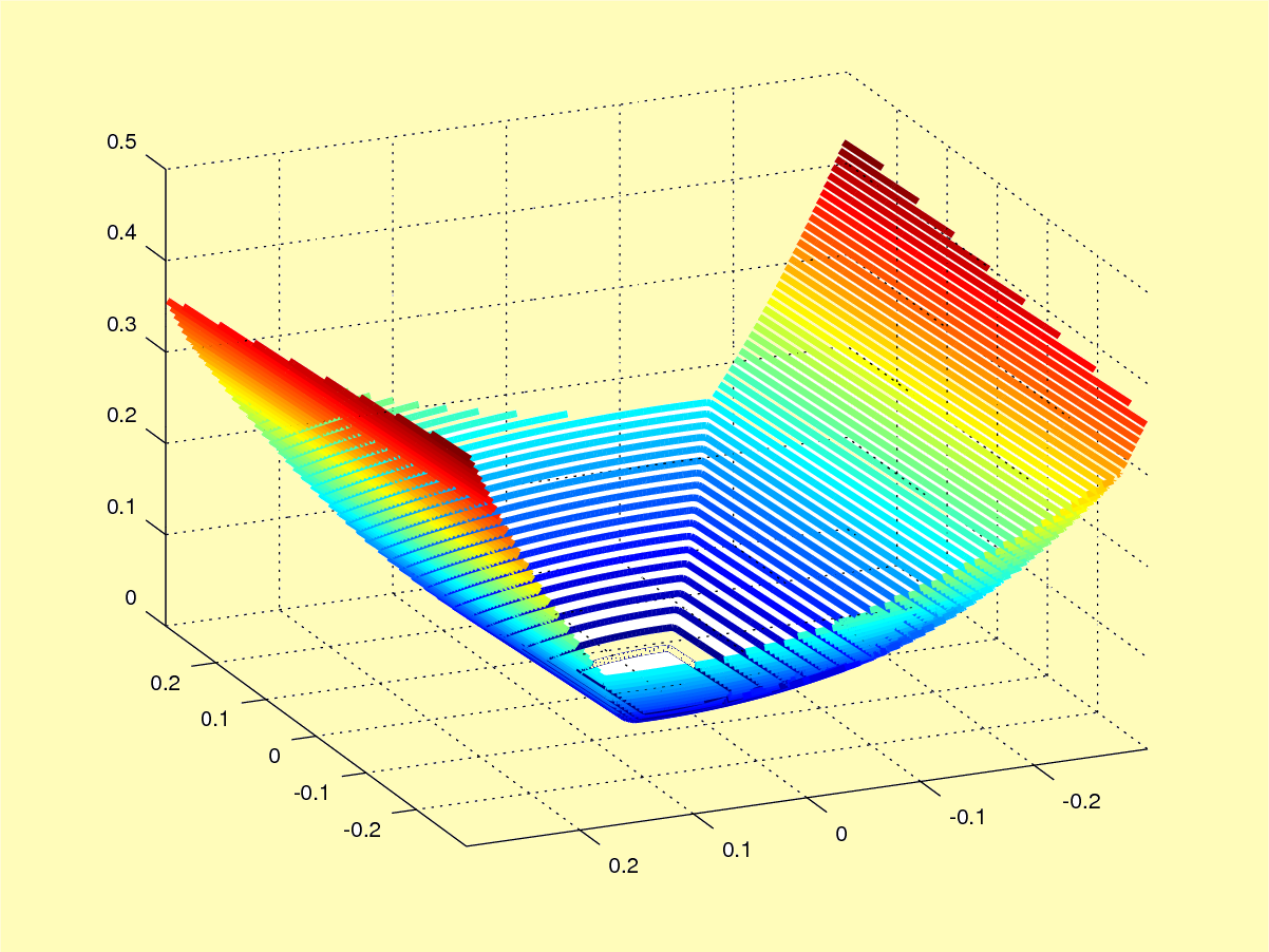
$$f(x) = \max\left(\left|x^\top Ax\right|, x^\top Bx\right)$$



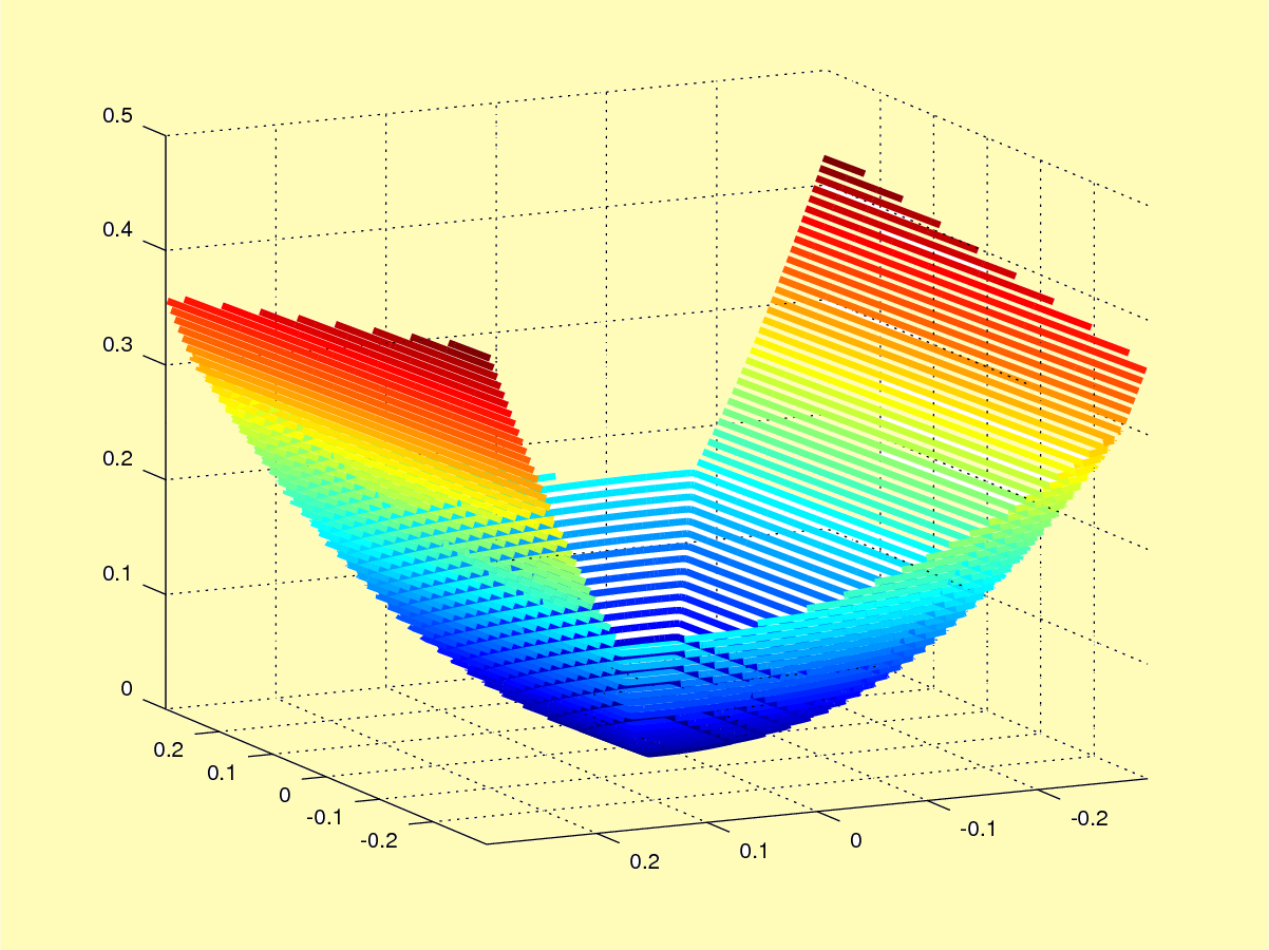
$\mathcal{VU}$s of the function $f(x) = \max\left(\left|x^\top Ax\right|, x^\top Bx\right)$



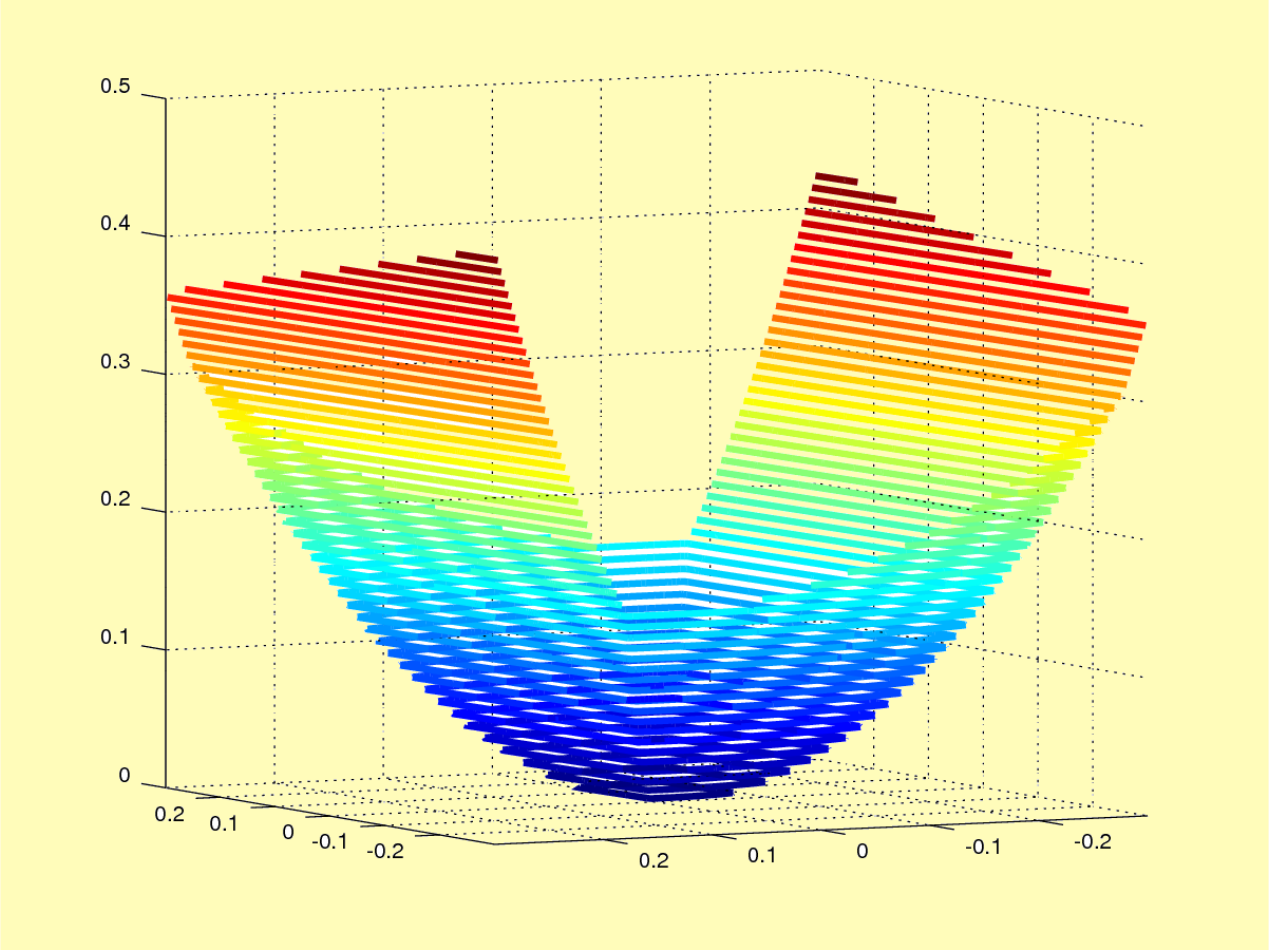
$\mathcal{VU}$s of the function $f(x) = \max\left(\left|x^\top Ax\right|, x^\top Bx\right)$



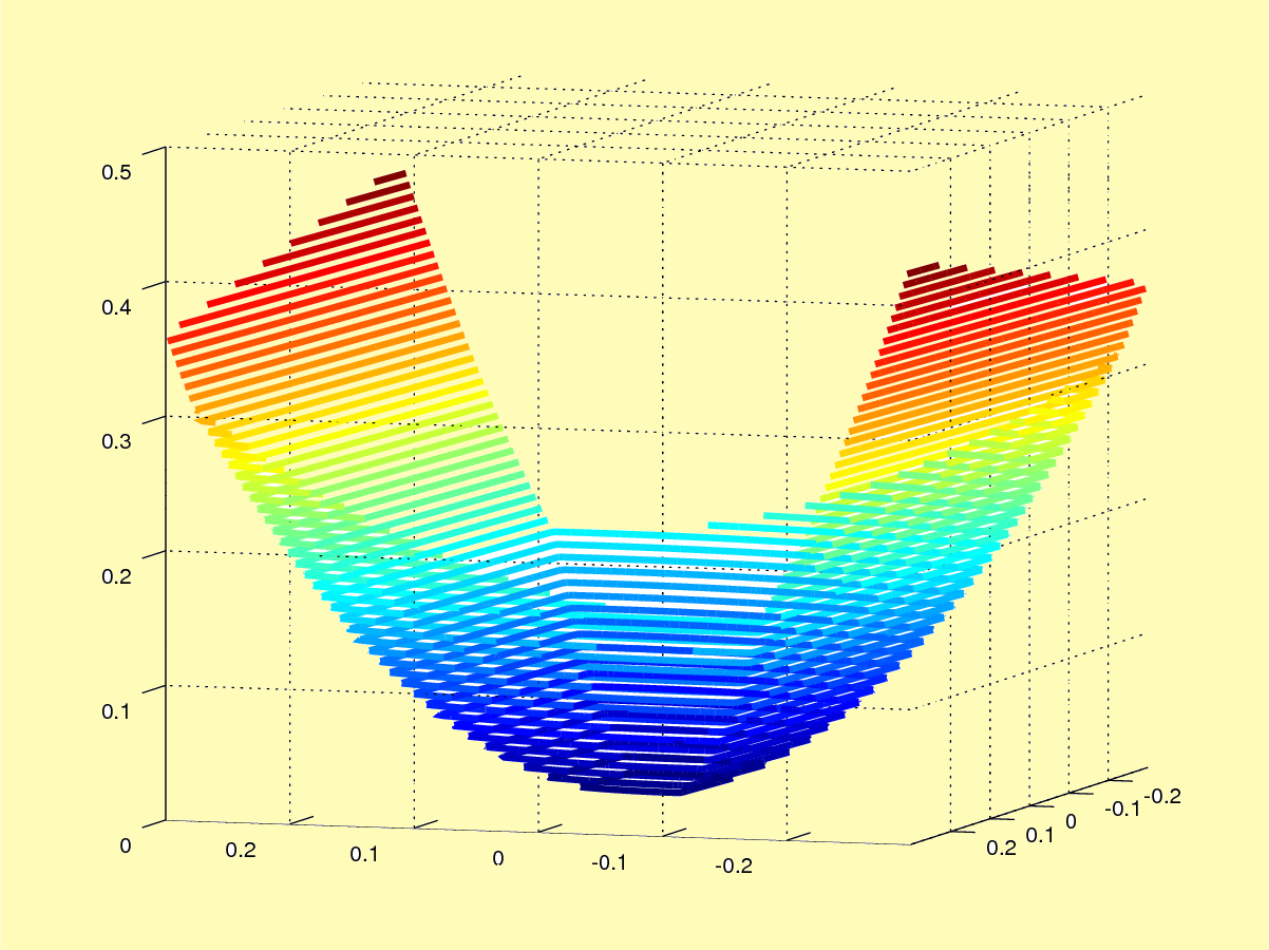
$\mathcal{VU}$s of the function $f(x) = \max\left(\left|x^\top Ax\right|, x^\top Bx\right)$



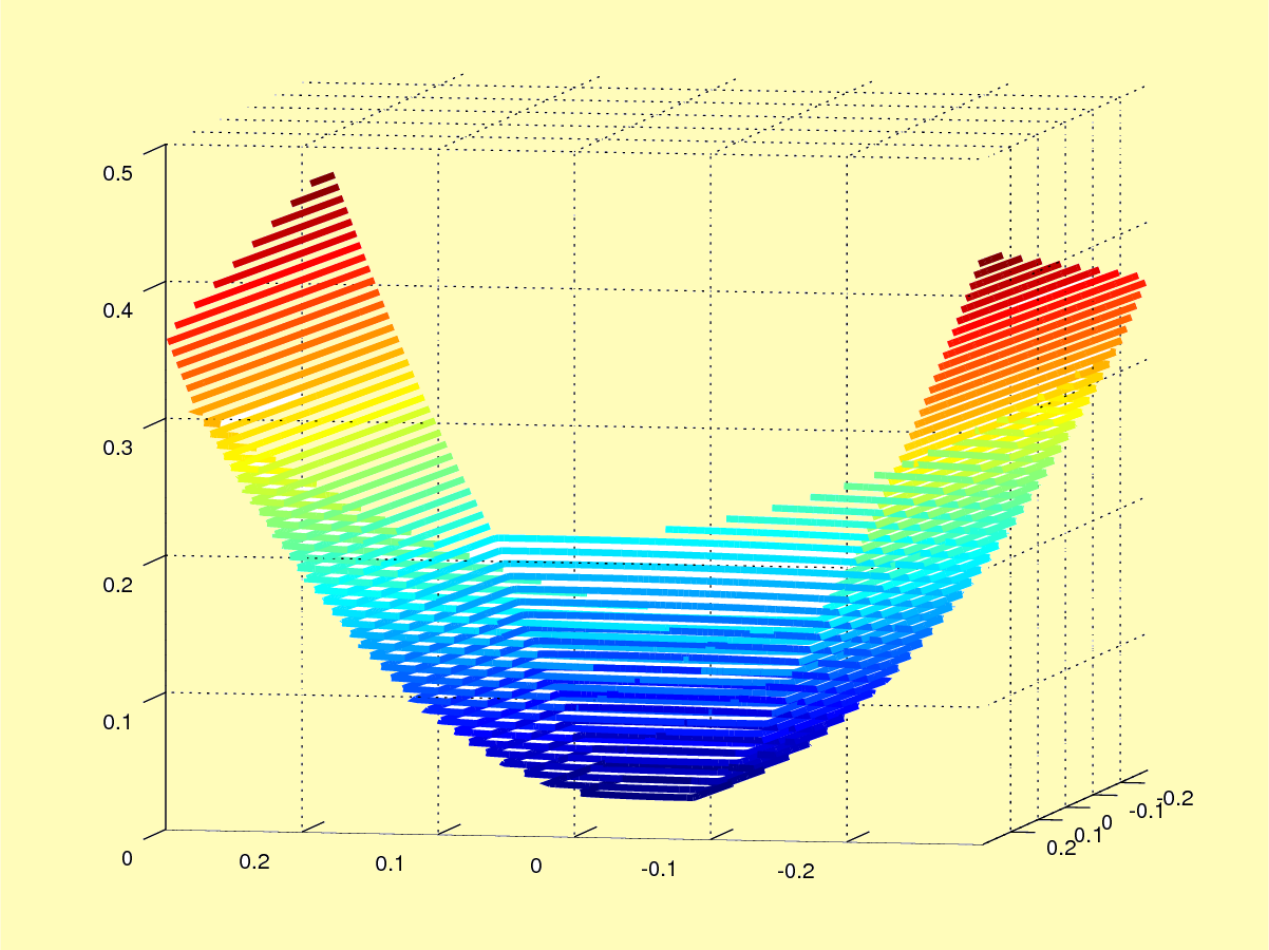
$\mathcal{VU}$s of the function $f(x) = \max\left(\left|x^\top Ax\right|, x^\top Bx\right)$



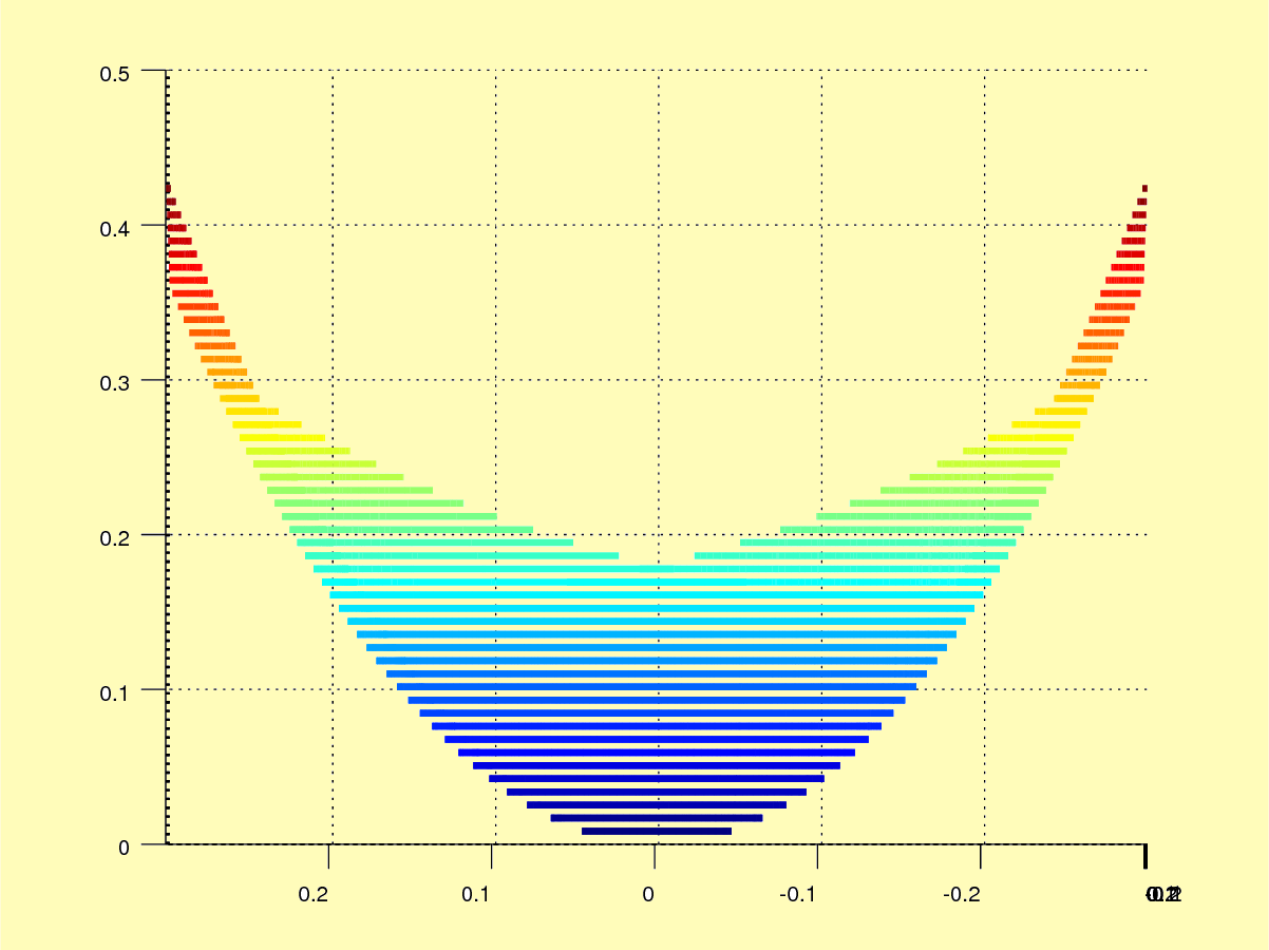
$\mathcal{VU}$s of the function $f(x) = \max\left(\left|x^\top Ax\right|, x^\top Bx\right)$



$\mathcal{VU}$s of the function $f(x) = \max\left(\left|x^\top Ax\right|, x^\top Bx\right)$

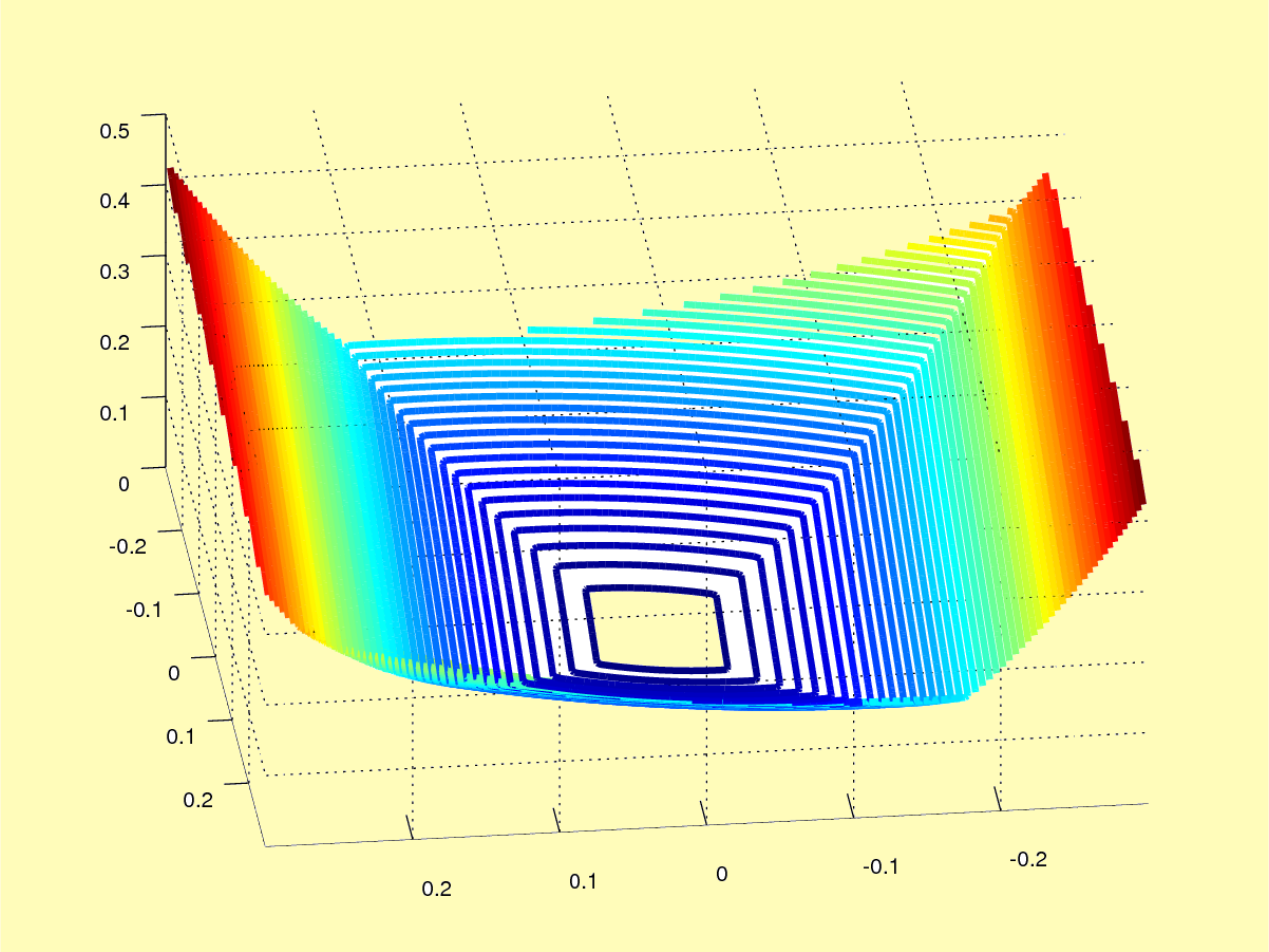


$\mathcal{VU}$s of the function $f(x) = \max\left(\left|x^\top Ax\right|, x^\top Bx\right)$



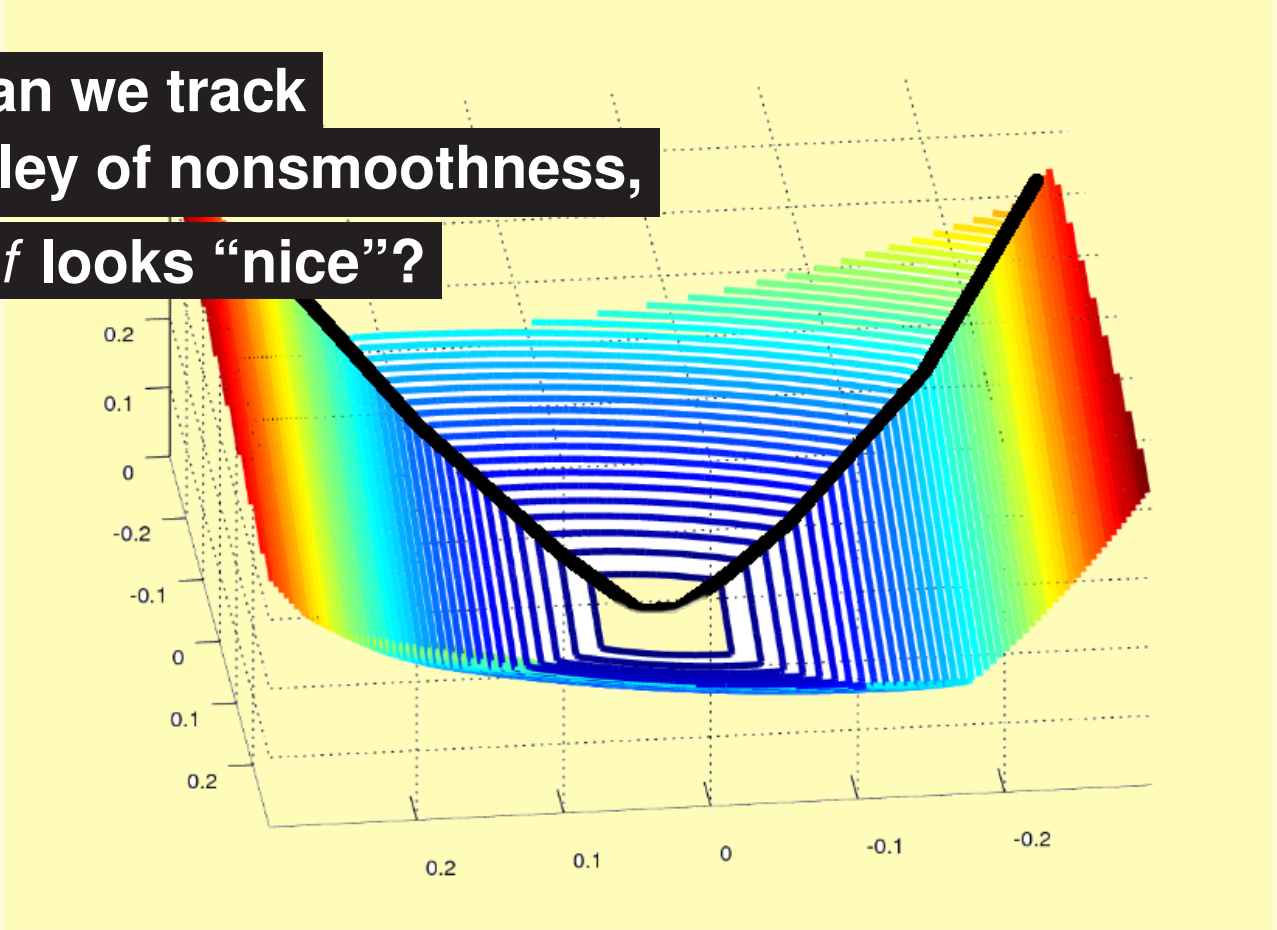
v_k of the function $f(x) = \max\left(\left|x^\top Ax\right|, x^\top Bx\right)$

How can we track the valley of nonsmoothness, where f looks "nice"?



$\mathcal{VU}$s of the function $f(x) = \max\left(\left|x^\top Ax\right|, x^\top Bx\right)$

How can we track the valley of nonsmoothness, where f looks "nice"?



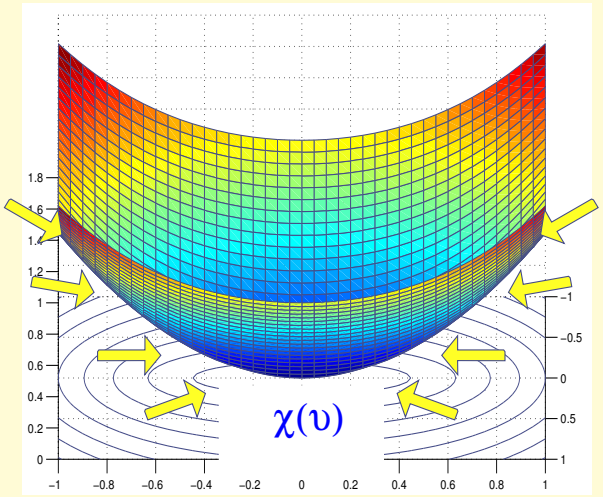
$\mathcal{VU}$-theory & primal-dual tracks

U -Lagrangian:

$$L_U(u,\bar{g}) := \inf_{v\in V}\{f(\bar{x}+u\oplus v)-\langle \bar{g},v\rangle\}$$
$$= f(\bar{x}+u\oplus v(u))-\langle \bar{g},v(u)\rangle$$

$\rightarrow L_U(0,\bar{g}) = f(\bar{x}),\quad \rightarrow L_U(u,\bar{g}) \in C^1(U)$

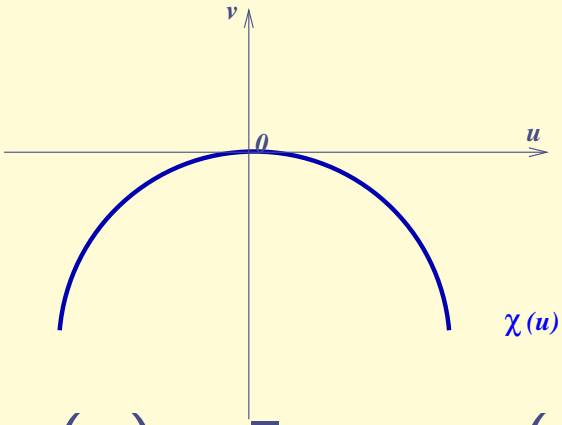
$\rightarrow$ minimizer $v = v(u)$ generates trajectories *smooth tangent to U*

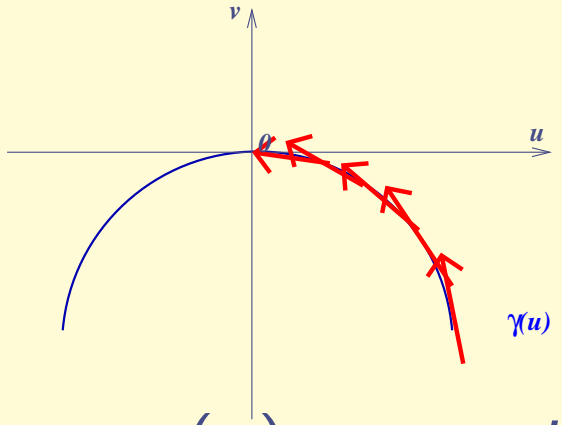


$$\chi(u) := \bar{x}+u\oplus v(u)$$

$\forall \bar{g} \in ri\,\partial f(\bar{x})$ same $v(u)$ and $L_U \in C^2$ i.e., has a U -Hessian

Dual tracks





$(arg = 0)$

$$\chi(u) = \bar{x} + u \oplus v(u) \qquad \gamma(u) = argmin\left\{|g|^2 : g \in \partial f(\chi(u)\right.$$

$$(\chi(u), \gamma(u)) \to (\bar{x}, 0) \text{ as } u \to 0$$

Good primal-dual track $\leftrightarrow$ $L_U \in C^2$ +

$0 \in ri\,\partial f(\bar{x})$ allows for Newton-like method to minimize f:

corrector	—	predictor method
proximal point	—	Newton method
V	—	U

Approximating primal-dual tracks

Fundamental theoretical result:

Proximal Points are on the primal track If

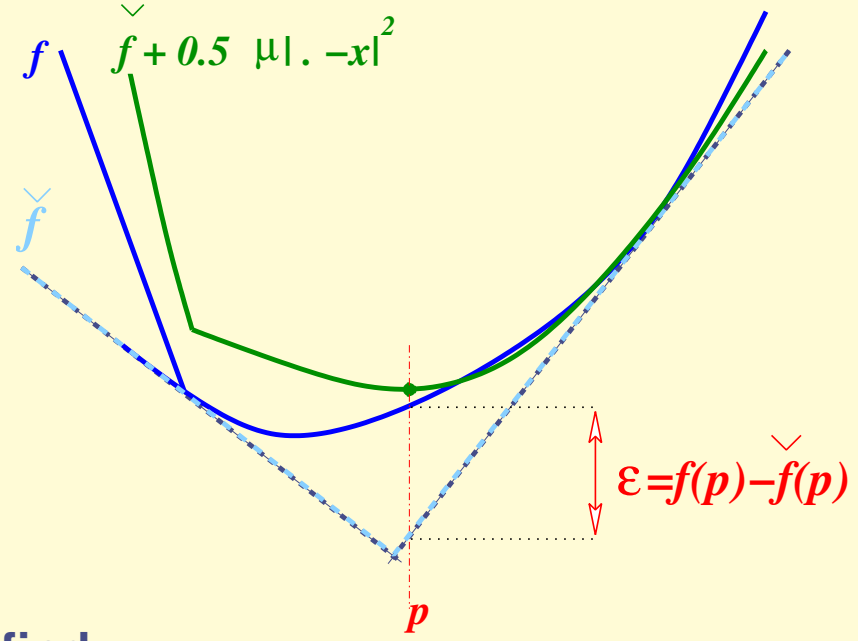
$arg = 0 \in ri\partial f(\bar{x})$, then $\exists u(x)$:

$$p(x) := \operatorname{argmin}\left\{ f(y) + \frac{1}{2}\mu|y-x|^2 \right\} = \chi(u(x))$$

for all $x \approx \bar{x}$ with $\mu = \mu(x) : \mu|x-\bar{x}| \to 0$ as $x \to \bar{x}$

$\Rightarrow$ **use a bundle subroutine**
to approximate the prox
and estimate the pair
$(\chi(u), \gamma(u))$

Bundle approximation



With bundle (x^i, f^i, g^i)
build M, a model for f **near** $\hat{x}$**, and find**

χ-QP solution: $p := argmin\left\{\mathbf{M}(y) + \frac{1}{2}\mu|y-\hat{x}|^2\right\} \approx \chi(u(x))$

γ-QP solution: $s := argmin\left\{|g|^2 : g \in \partial\right.$

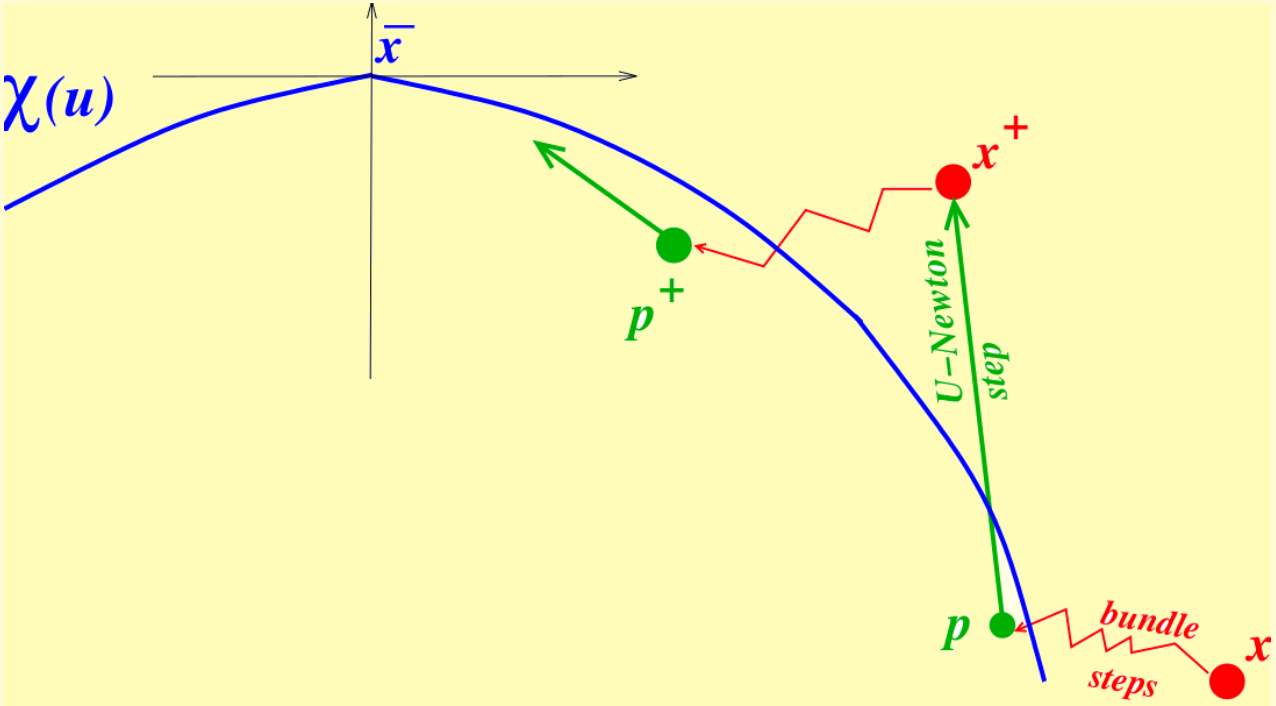
$hatfch(p) \approx \gamma(u(x))$

UNTIL **"good enough":** $\varepsilon \leq \sigma/\mu|s|^2$

By-product: local V U -decomposition, VU

Newton-like corrector-predictor V U algorithm

Given x and a bundle:



- **Corrector step:** Solve $(\chi-$ and γ-QP$)$'s

  $\Rightarrow$ new p, s, VU, and determine H (U-Hessian)
- **Predictor step:** Solve $H\Delta u = -\mathrm{U}^\top s \Rightarrow x^+ = p + \mathrm{U}\Delta u$

Convergence properties

- If infinite bundle steps, the inner sequence converges to a minimizer of f
- Otherwise, either the outer sequence $\{p\}$ is finite with $s = 0$ and last p minimizes f
- or, $\{f(p)\}$ is infinite and decreasing $\Rightarrow$
  - either f unbounded below,
  - or $s \to 0$ and any $acc(\{p\})$ minimizes f

    when $\{\sigma/\mu\} \to 0$.

If the U -Hessian at $\bar{x}$ is positive definite, $0 \in ri\,\partial f(\bar{x})$, and

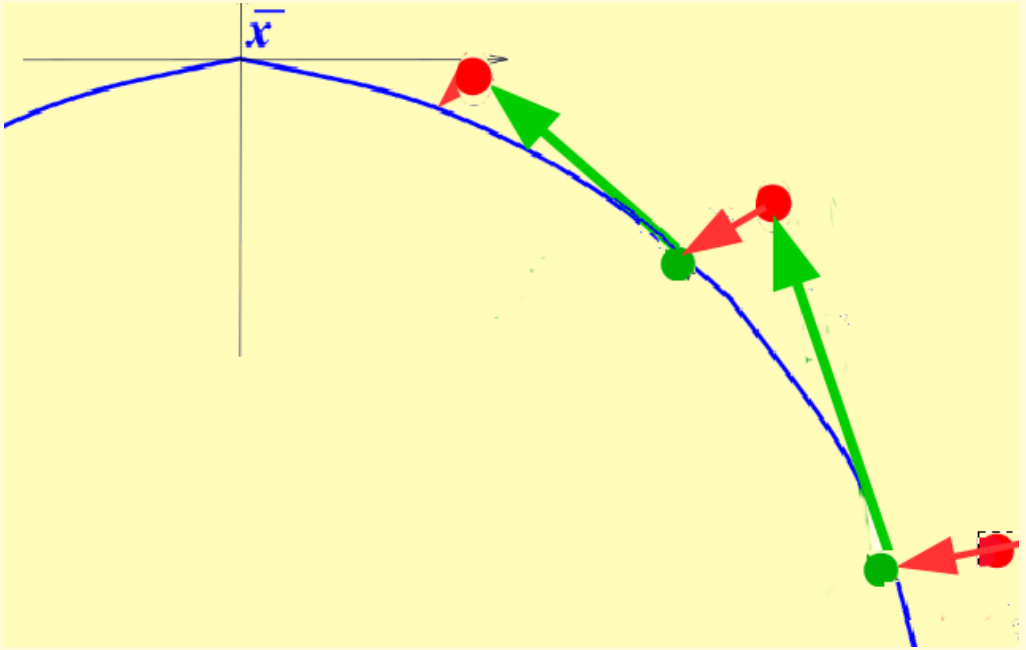
- $\frac{\sigma}{\mu^2} = O(|s^-|^2)$, bounded $\{H^{-1}\}$, $\mathrm{U} \to U$,
- Dennis-Moré-like condition for $\{H\}$
- s approximates γ superlinearly

$\Rightarrow$ superlinear convergence of $\{p\}$ to $\bar{x}$

$\mathcal{VU}$-Algorithm:

As $u \to 0$:

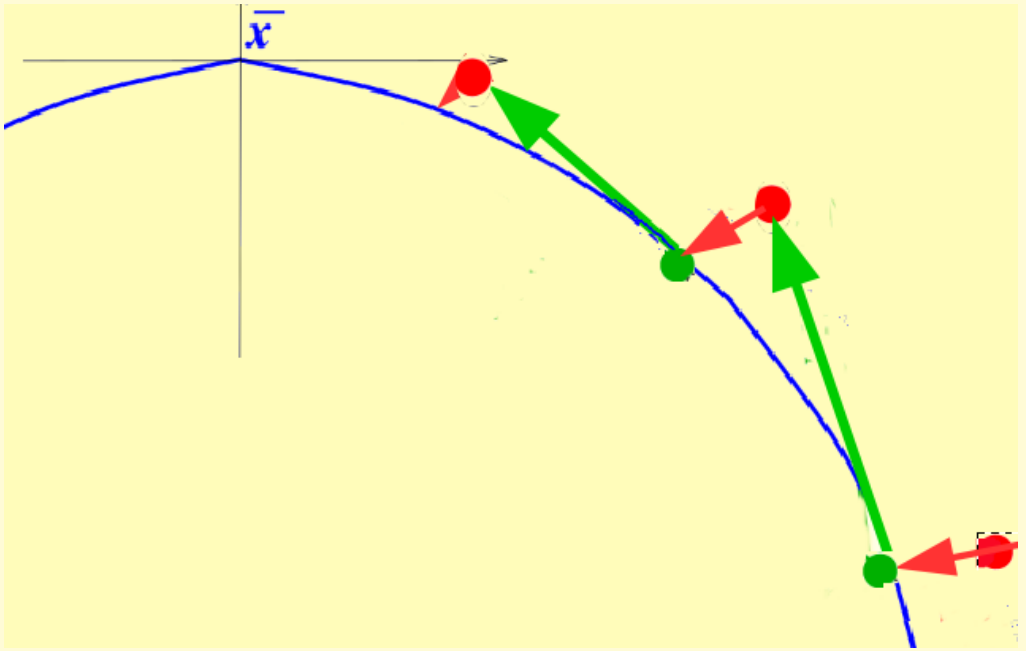
- The ~~fast~~ track $\chi(u) \to x^*$
- The $\mathcal{U}$-gradient, $\nabla L_{\mathcal{U}}(u; g^*_{\mathcal{V}}) \to 0$



$\mathcal{VU}$-Algorithm:

As $u \to 0$:

- **$\mathcal{V}$-step:** The ~~fast~~ track $\chi(u) \to x^*$
- **$\mathcal{U}$-step:** The $\mathcal{U}$-gradient, $\nabla L_{\mathcal{U}}(u; g^*_{\mathcal{V}}) \to 0$



$\mathcal{VU}$-Algorithm:

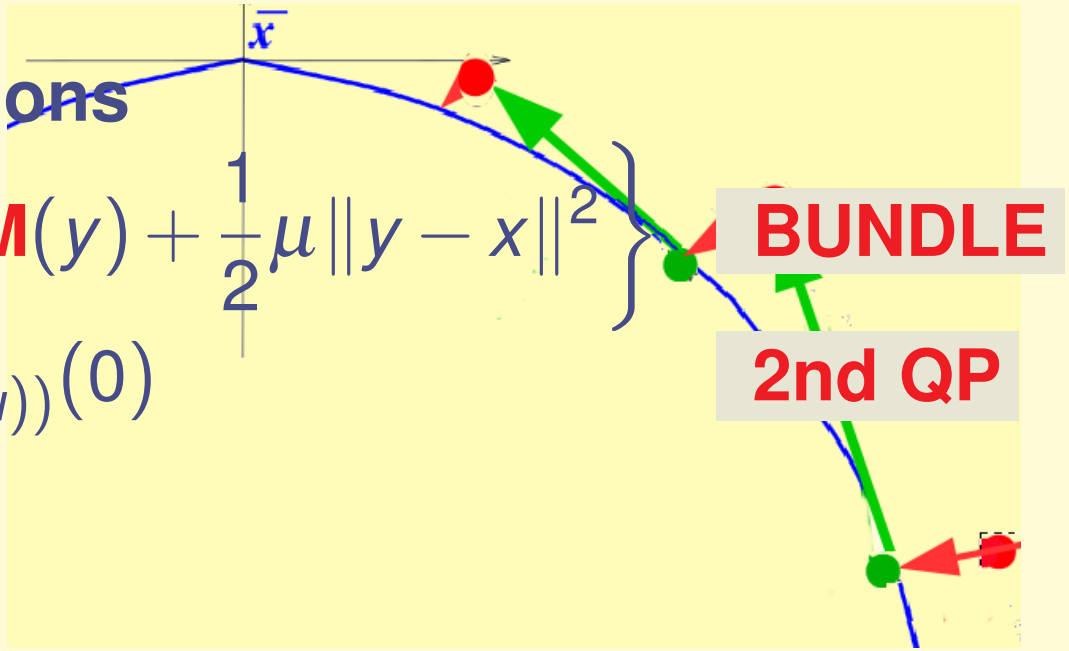
As $u \to 0$:

- **$\mathcal{V}$-step:** The ~~slow~~ ***fast*** track $\chi(u) \to x^*$
- **$\mathcal{U}$-step:** The $\mathcal{U}$-gradient, $\nabla L_{\mathcal{U}}(u; g^*_{\mathcal{V}}) \to 0$

Primal-dual approximations

$$\chi \approx \operatorname{argmin}\left\{ \mathbf{M}(y) + \frac{1}{2}\mu \|y - x\|^2 \right\}$$

$$\nabla L_{\mathcal{U}} \approx \operatorname{proj}_{\partial \mathbf{M}(\chi(u))}(0)$$



$\mathcal{VU}$-Algorithm: approximations

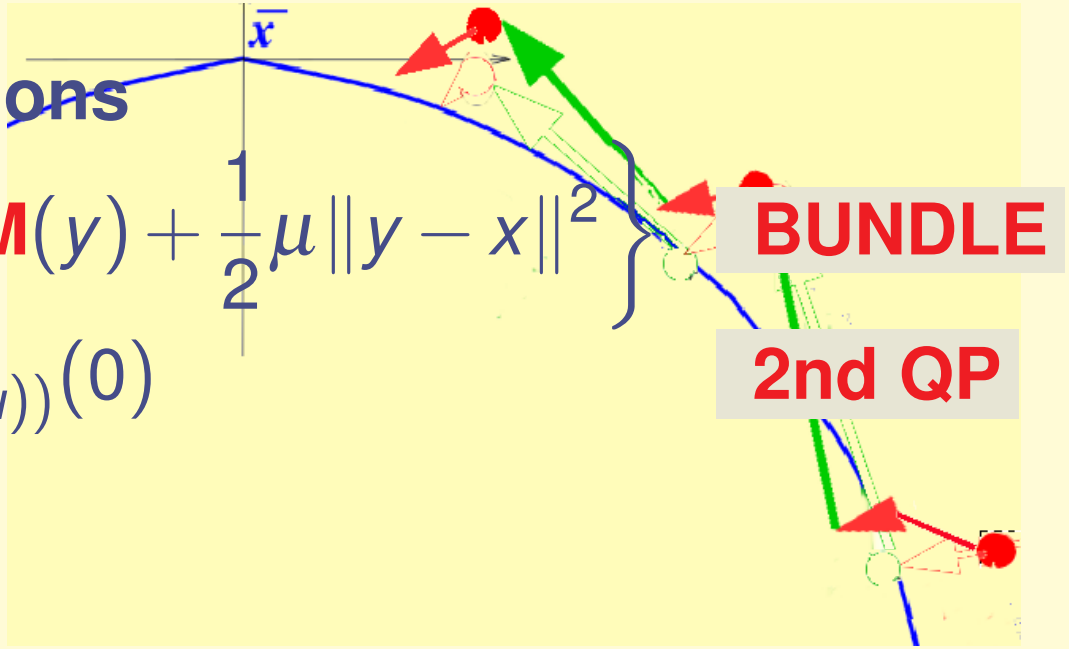
As $u \to 0$:

- **$\mathcal{V}$-step:** The ~~fast~~ track $\chi(u) \to x^*$
- **$\mathcal{U}$-step:** The $\mathcal{U}$-gradient, $\nabla L_{\mathcal{U}}(u; g^*_{\mathcal{V}}) \to 0$

Primal-dual approximations

$$\chi \approx \operatorname{argmin}\left\{ \mathbf{M}(y) + \frac{1}{2}\mu \|y - x\|^2 \right\}$$

$$\nabla L_{\mathcal{U}} \approx \operatorname{proj}_{\partial \mathbf{M}(\chi(u))}(0)$$



$\mathcal{VU}$-Algorithm: approximations

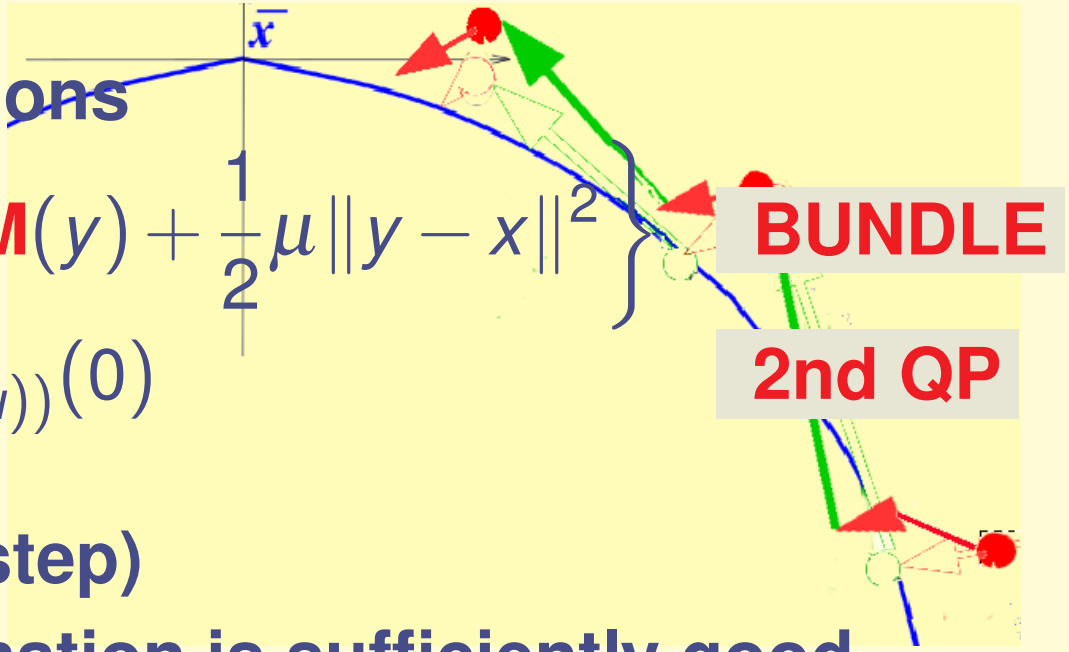
As $u \to 0$:

- **$\mathcal{V}$-step:** The ~~fast~~ track $\chi(u) \to x^*$
- **$\mathcal{U}$-step:** The $\mathcal{U}$-gradient, $\nabla L_{\mathcal{U}}(u; g^*_{\mathcal{V}}) \to 0$

Primal-dual approximations

$$\chi \approx \operatorname{argmin}\left\{ \mathbf{M}(y) + \frac{1}{2}\mu\|y - x\|^2 \right\}$$

$$\nabla L_{\mathcal{U}} \approx \operatorname{proj}_{\partial \mathbf{M}(\chi(u))}(0)$$

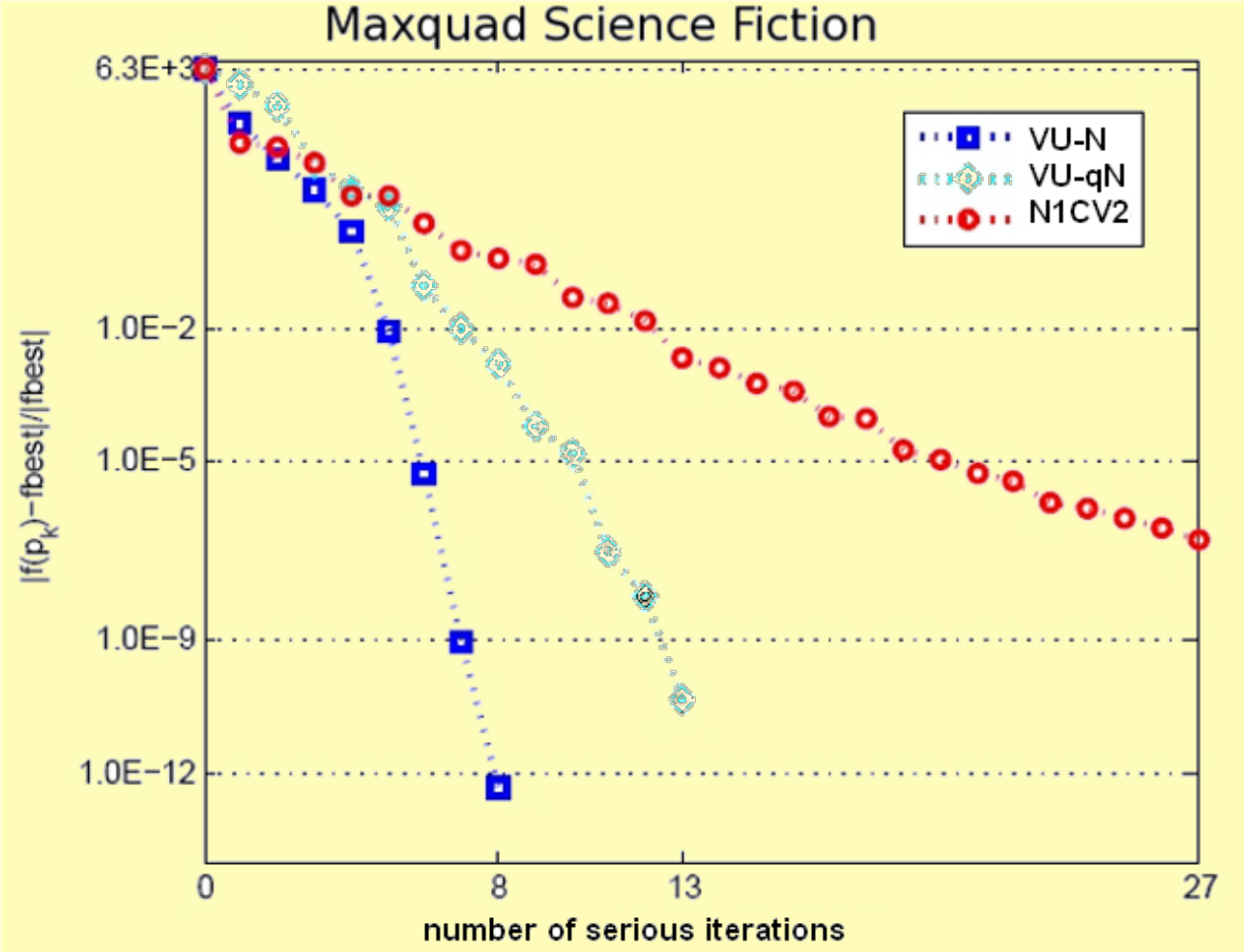


Change x (serious step)

only when approximation is sufficiently good

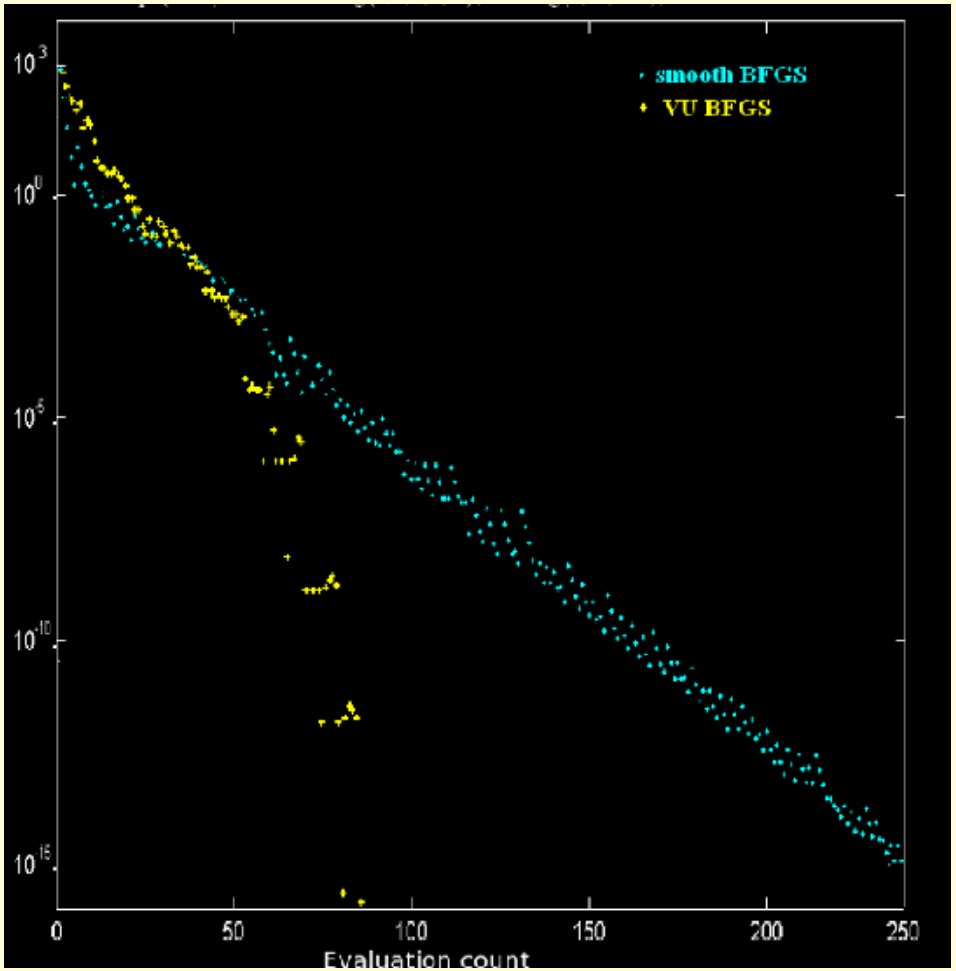
$\mathcal{VU}$-Algorithm is globally and superlinearly convergent

Comparison with bundle method



$\mathcal{VU}$-Algorithm is globally and superlinearly convergent

Comparison with BFGS method



Concluding comments

On-going & Future work

- **O**: Derivative-free variant
- **O**: ε-V U variant
- **F**: Application to 2-stage stochastic programming problems