*i*Piano: Inertial Proximal Algorithm for Non-convex Optimization

Thomas Pock

Institute for Computer Graphics and Vision
Graz University of Technology

MOBIS Workshop, University of Graz, July 5th, 2014

Graz University of Technology

Joint work with:
P. Ochs, T. Brox (University of Freiburg)
Y. Chen (Graz University of Technology)

Energy minimization methods

- Typical variational approaches to solve inverse problems consist of a regularization term and a data term

$$\min_{u} \left\{ E(u|f) = \mathcal{R}(u) + \mathcal{D}(u, f) \right\} ,$$

where f is the input data and u is the unknown solution

Energy minimization methods

- Typical variational approaches to solve inverse problems consist of a regularization term and a data term

$$\min_{u} \{E(u|f) = \mathcal{R}(u) + \mathcal{D}(u, f)\} \ ,$$

  where f is the input data and u is the unknown solution

- Low-energy states reflect the physical properties of the problem

Energy minimization methods

- Typical variational approaches to solve inverse problems consist of a regularization term and a data term

$$\min_{u} \{E(u|f) = \mathcal{R}(u) + \mathcal{D}(u, f)\} \ ,$$

  where f is the input data and u is the unknown solution
- Low-energy states reflect the physical properties of the problem
- Minimizer provides the best (in the sense of the model) solution to the problem

Optimization problems are unsolvable

Consider the following general mathematical optimization problem:

$$\begin{aligned} &\min f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0 \,, \quad i = 1 \ldots m \\ & x \in X \,, \end{aligned}$$

where $f_0(x)...f_m(x)$ are real-valued functions, $x = (x_1, ...x_n)^T \in \mathbb{R}^n$ is a n-dimensional real-valued vector, and X is a subset of $\mathbb{R}^n$

How to solve this problem?

Optimization problems are unsolvable

Consider the following general mathematical optimization problem:

$$\begin{aligned} &\min f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0 \ , \quad i = 1 \dots m \\ & x \in X \ , \end{aligned}$$

where $f_0(x)...f_m(x)$ are real-valued functions, $x = (x_1, ...x_n)^T \in \mathbb{R}^n$ is a n-dimensional real-valued vector, and X is a subset of $\mathbb{R}^n$

How to solve this problem?

- Naive: "Download a commercial package ..."

Optimization problems are unsolvable

Consider the following general mathematical optimization problem:

$$\begin{aligned} &\min f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0 \ , \quad i = 1 \dots m \\ & x \in X \ , \end{aligned}$$

where $f_0(x) ... f_m(x)$ are real-valued functions, $x = (x_1, ... x_n)^T \in \mathbb{R}^n$ is a n-dimensional real-valued vector, and X is a subset of $\mathbb{R}^n$

How to solve this problem?

- Naive: "Download a commercial package ..."
- Reality: "Finding a solution is far from being trivial!"

Optimization problems are unsolvable

Consider the following general mathematical optimization problem:

$$\begin{aligned} & \min f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0 \ , \quad i = 1 \ldots m \\ & x \in X \ , \end{aligned}$$

where $f_0(x) ... f_m(x)$ are real-valued functions, $x = (x_1, ... x_n)^T \in \mathbb{R}^n$ is a n-dimensional real-valued vector, and X is a subset of $\mathbb{R}^n$

How to solve this problem?

- Naive: "Download a commercial package ..."
- Reality: "Finding a solution is far from being trivial!"

- Efficiently finding solutions to the whole class of Lipschitz continuous problems is a hopeless case [Nesterov '04]
- Can take several million years for small problems with only 10 unknowns

Optimization problems are unsolvable

Consider the following general mathematical optimization problem:

$$\begin{aligned} &\min f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0\ , \quad i = 1 \dots m \\ & x \in X\ , \end{aligned}$$

where $f_0(x) ... f_m(x)$ are real-valued functions, $x = (x_1, ... x_n)^T \in \mathbb{R}^n$ is a n-dimensional real-valued vector, and X is a subset of $\mathbb{R}^n$

How to solve this problem?

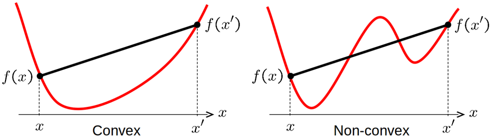
- Naive: "Download a commercial package ..."
- Reality: "Finding a solution is far from being trivial!"

- Efficiently finding solutions to the whole class of Lipschitz continuous problems is a hopeless case [Nesterov '04]
- Can take several million years for small problems with only 10 unknowns

- "Optimization problems are unsolvable"

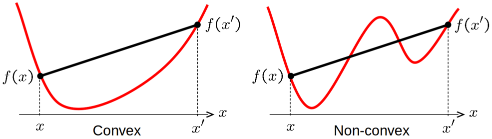
[Nesterov '04]

Convex versus non-convex



"The great watershed in optimization is not between linearity and non-linearity, but convexity and non-convexity." *R. Rockafellar, 1993*

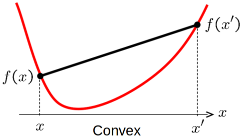
Convex versus non-convex



"The great watershed in optimization is not between linearity and non-linearity, but convexity and non-convexity." *R. Rockafellar, 1993*

- Convex problems
  - Any local minimizer is a global minimizer
  - Result is independent of the initialization
  - Convex models often inferior

Convex versus non-convex



"The great watershed in optimization is not between linearity and non-linearity, but convexity and non-convexity." R. Rockafellar, 1993

- Convex problems
  - Any local minimizer is a global minimizer
  - Result is independent of the initialization
  - Convex models often inferior
- Non-convex problems
  - In general no chance to find the global minimizer
  - Result strongly depends on the initialization
  - Often give more accurate models

Non-convex optimization problems

- Smooth non-convex problems can be solved via generic nonlinear numerical optimization algorithms (SD, CG, BFGS, ...)
- Hard to generalize to constraints, or non-differentiable functions
- Line-search procedure can be time intensive

Non-convex optimization problems

- Smooth non-convex problems can be solved via generic nonlinear numerical optimization algorithms (SD, CG, BFGS, ...)
- Hard to generalize to constraints, or non-differentiable functions
- Line-search procedure can be time intensive

- A reasonable idea is to develop algorithms for special classes of structured non-convex problems
- A promising class of problems that has a moderate degree of non-convexity is given by the sum of a smooth non-convex function and a non-smooth convex function [Sra '12], [Chouzenoux, Pesquet, Repetti '13]

Problem definition

- We consider the problem of minimizing a function $h\colon X \to \mathbb{R} \cup \{+\infty\}$

$$\min_{x \in X} h(x) = f(x) + g(x)\,,$$

where X is a finite dimensional real vector space.

- We assume that h is coercive, i.e. $\|x\|_2 \to +\infty \quad \Rightarrow \quad h(x) \to +\infty$ and bounded from below by some value $\underline{h} > -\infty$

Problem definition

- We consider the problem of minimizing a function $h\colon X \to \mathbb{R} \cup \{+\infty\}$

$$\min_{x \in X} \; h(x) = f(x) + g(x)\,,$$

where X is a finite dimensional real vector space.

- We assume that h is coercive, i.e. $\|x\|_2 \to +\infty \quad \Rightarrow \quad h(x) \to +\infty$ and bounded from below by some value $\underline{h} > -\infty$

- The function f is possibly non-convex but has a Lipschitz continuous gradient, i.e.

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$$

Problem definition

- We consider the problem of minimizing a function $h \colon X \to \mathbb{R} \cup \{+\infty\}$

$$\min_{x \in X} \; h(x) = f(x) + g(x) \,,$$

where X is a finite dimensional real vector space.

- We assume that h is coercive, i.e. $\|x\|_2 \to +\infty \quad \Rightarrow \quad h(x) \to +\infty$ and bounded from below by some value $\underline{h} > -\infty$

- The function f is possibly non-convex but has a Lipschitz continuous gradient, i.e.

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L \|x - y\|_2$$

- The function g is a proper lower semi-continuous convex function with an efficient to compute proximal map

$$(I + \alpha \partial g)^{-1}(\hat{x}) := \arg\min_{x \in X} \frac{\|x - \hat{x}\|_2^2}{2} + \alpha g(x) \,,$$

where $\alpha > 0$.

Forward-backward splitting

- We aim at seeking a critical point x^*, i.e. a point satisfying $0 \in \partial h(x^*)$ which in our case becomes

$$-\nabla f(x^*) \in \partial g(x^*) \,.$$

- A critical point can also be characterized via the *proximal residual*

$$r(x) := x - (I + \partial g)^{-1}(x - \nabla f(x)) \,,$$

  where I is the identity map.
- Clearly $r(x^*) = 0$ implies that x^* is a critical point.
- The norm of the proximal residual can be used as a (bad) measure of optimality

Forward-backward splitting

- We aim at seeking a critical point x^*, i.e. a point satisfying $0 \in \partial h(x^*)$ which in our case becomes
$$-\nabla f(x^*) \in \partial g(x^*)\,.$$
- A critical point can also be characterized via the *proximal residual*
$$r(x) := x - (I + \partial g)^{-1}(x - \nabla f(x))\,,$$
where I is the identity map.
- Clearly $r(x^*) = 0$ implies that x^* is a critical point.
- The norm of the proximal residual can be used as a (bad) measure of optimality
- The proximal residual already suggests an iterative method of the form
$$x^{n+1} = (I + \alpha\partial g)^{-1}(x^n - \alpha\nabla f(x^n))$$
- For f convex, this algorithm is well studied [Lions, Mercier '79], [Tseng '91], [Daubechie et al. '04], [Combettes, Wajs '05], [Raguet, Fadili, Peyré '13]

Inertial/accelerated methods

- Inertial: Introduced by Polyak in [Polyak '64] as a special case of multi-step algorithms for minimizing a μ-strongly convex function:

$$x^{n+1} = x^n - \alpha \nabla f(x^n) + \beta(x^n - x^{n-1})$$

- Can be seen as an explicit finite differences discretization of the heavy-ball with friction dynamical system

$$\ddot{x}(t) + \gamma \dot{x}(t) + \nabla f(x(t)) = 0\,.$$

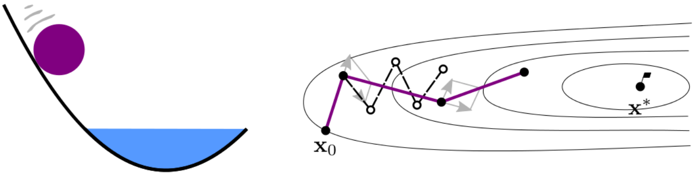
Inertial/accelerated methods

- Inertial: Introduced by Polyak in [Polyak '64] as a special case of multi-step algorithms for minimizing a μ-strongly convex function:

$$x^{n+1} = x^n - \alpha \nabla f(x^n) + \beta(x^n - x^{n-1})$$

- Can be seen as an explicit finite differences discretization of the heavy-ball with friction dynamical system

$$\ddot{x}(t) + \gamma \dot{x}(t) + \nabla f(x(t)) = 0\,.$$



Source: Stich et al.

A note on the convex case

If f is l - strongly convex and ∇f is L - Lipschitz than by setting

- $\alpha = \frac{4}{(\sqrt{l}+\sqrt{L})^2}$
- $\beta = \left(\frac{\sqrt{l}-\sqrt{L}}{\sqrt{l}+\sqrt{L}}\right)^2$

yields an "optimal" linear convergence rate of

$$\|x^n - x^*\|_2 \leq \left(\frac{\sqrt{L}-\sqrt{l}}{\sqrt{L}+\sqrt{l}}\right)^n \|x^0 - x^*\|_2$$

A note on the convex case

If f is l - strongly convex and ∇f is L - Lipschitz than by setting

- $\alpha = \frac{4}{(\sqrt{l}+\sqrt{L})^2}$
- $\beta = \left(\frac{\sqrt{l}-\sqrt{L}}{\sqrt{l}+\sqrt{L}}\right)^2$

yields an "optimal" linear convergence rate of

$$\|x^n - x^*\|_2 \leq \left(\frac{\sqrt{L}-\sqrt{l}}{\sqrt{L}+\sqrt{l}}\right)^n \|x^0 - x^*\|_2$$

- No first-order method can be faster!
- Same performance as CG, but we need to know l, L
- CG only makes sense for quadratic functions
- Heavy-ball can be used together with constraints, non-smooth functions [Ochs, P. et al, '14]

iPiano

inertial **P**roxi**m**al **a**lgorithm for **n**on-convex **o**ptimization

- Initialization: Choose $x^0 \in \mathrm{dom}\, h$ and set $x^{-1} = x^0$.
- Iterations $(n \geq 0)$: Update

$$x^{n+1} = (I + \alpha_n \partial g)^{-1}(x^n - \alpha_n \nabla f(x^n) + \beta_n(x^n - x^{n-1}))\,,$$

for some sequences (α_n), (β_n).

Questions:

- When does this algorithm converge (subsequence, whole sequence)?
- How fast does it converge (convergence rate)?
- Applications?

The Kurdyka-Łojasiewicz property

Definition

The function $F\colon \mathbb{R}^N \to \mathbb{R} \cup \{\infty\}$ has the Kurdyka-Łojasiewicz property at $x^* \in \operatorname{dom}\partial F$, if there exist $\eta \in (0,\infty]$, a neighborhood U of x^* and a continuous concave function $\varphi\colon [0,\eta) \to \mathbb{R}_+$ such that $\varphi(0) = 0$, $\varphi \in C^1((0,\eta))$, for all $s \in (0,\eta)$ it is $\varphi'(s) > 0$, and for all $x \in U \cap [F(x^*) < F < F(x^*) + \eta]$ the Kurdyka-Łojasiewicz inequality holds, i.e.,

$$\varphi'(F(x) - F(x^*))\operatorname{dist}(0, \partial F(x)) \geq 1\,.$$

- Intuetively, we can bound the subgradients from below by a re-parametrization of the function values
- The Kurdyka-Łojasiewicz property holds for real, semi-algebraic functions
- Recently, the Kurdyka-Łojasiewicz property attracted a lot of attention for proving convergence of descent methods [Attouch, Bolte et al. '10-'13], [Chouzenoux, Pesquet, Repetti '13], ...

Abstract convergence for two-step algorithms

- We extend the convergence result of [Attouch, Bolte, Svaiter '13] for one-step algorithms to the case of two-step algorithms
- Let $F(z^n)$ be a proper, lower semicontinuous function, $(z^n) = (x^n, x^{n-1})$, $\Delta_n := \|x^n - x^{n-1}\|_2$

Abstract convergence for two-step algorithms

- We extend the convergence result of [Attouch, Bolte, Svaiter '13] for one-step algorithms to the case of two-step algorithms
- Let $F(z^n)$ be a proper, lower semicontinuous function, $(z^n) = (x^n, x^{n-1})$, $\Delta_n := \|x^n - x^{n-1}\|_2$
- We require the following conditions to be satisfied:

(H1) For each $n \in \mathbb{N}$, it holds

$$F(z^{n+1}) + a\Delta_n^2 \leq F(z^n)\,.$$

(H2) For each $n \in \mathbb{N}$, there exists $w^{n+1} \in \partial F(z^{n+1})$ such that

$$\|w^{n+1}\|_2 \leq \frac{b}{2}(\Delta_n + \Delta_{n+1})\,.$$

(H3) There exists a subsequence $(z^{n_j})_{j\in\mathbb{N}}$ such that

$$z^{n_j} \to \tilde{z} \quad \text{and} \quad F(z^{n_j}) \to F(\tilde{z})\,, \qquad \text{as } j \to \infty\,.$$

Convergence of the whole sequence to a critical point

Theorem

Let $F\colon \mathbb{R}^{2N} \to \mathbb{R} \cup \{\infty\}$ be a proper lower semi-continuous function and $(z^n)_{n\in\mathbb{N}} = (x^n, x^{n-1})_{n\in\mathbb{N}}$ a sequence that satisfies H1, H2, and H3. Moreover, let F have the Kurdyka-Łojasiewicz property at the cluster point $\tilde{x}$ specified in H3.
Then, the sequence $(x^n)_{n=0}^{\infty}$ has finite length, i.e., $\sum_{n=1}^{\infty} \Delta_n < \infty$, and converges to $\bar{x} = \tilde{x}$ as $n \to \infty$, where $(\bar{x}, \bar{x})$ is a critical point of F.

- Details of the proof see [Ochs, Chen, Brox, P. SIIMS '14]
- In order to apply this result to iPiano, it remains to show that H1-H3 hold

Back to our class of problems: basic inequalities

We can describe our model class $h = f + g$, by the following to inequalities

Lemma

Let ∇f *be* L*-Lipschitz. Then for any* $x, y \in \operatorname{dom} f$ *it holds that*

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2} \|x - y\|_2^2 .$$

Back to our class of problems: basic inequalities

We can describe our model class $h = f + g$, by the following to inequalities

Lemma

Let ∇f be L-Lipschitz. Then for any $x, y \in \operatorname{dom} f$ it holds that

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2} \|x - y\|_2^2 .$$

Lemma

Let g be a proper lower semi-continuous convex function, then it holds for any $x, y \in X$, $s \in \partial g(x)$ that

$$g(y) \geq g(x) + \langle s, y - x \rangle \ .$$

A Lyapunov function

- Let us consider the function $H_\delta(x,y) := h(x) + \delta\|x-y\|_2^2$, $\delta \in \mathbb{R}$, and the distance of two subsequent iterates $\Delta_n := \|x^n - x^{n-1}\|_2$
- The main iterate of the algorithm is given by

$$x^{n+1} = (I + \alpha_n \partial g)^{-1}(x^n - \alpha_n \nabla f(x^n) + \beta_n(x^n - x^{n-1}))$$

- Applying the previous inequalities to the iteration yields the following result:

Lemma

(a) *The sequence* $(H_{\delta_n}(x^n, x^{n-1}))_{n=0}^{\infty}$ *is monotonically decreasing and thus converging. In particular, it holds*

$$H_{\delta_{n+1}}(x^{n+1}, x^n) \leq H_{\delta_n}(x^n, x^{n-1}) - \gamma_n \Delta_n^2 ,$$

where γ_n, δ_n *is some pos. parameter depending on* α_n, β_n.

(b) *It holds* $\sum_{n=0}^{\infty} \Delta_n^2 < \infty$ *and, thus,* $\lim_{n\to\infty} \Delta_n = 0$.

A Lyapunov function

- Let us consider the function $H_\delta(x,y) := h(x) + \delta\|x-y\|_2^2$, $\delta \in \mathbb{R}$, and the distance of two subsequent iterates $\Delta_n := \|x^n - x^{n-1}\|_2$
- The main iterate of the algorithm is given by

$$x^{n+1} = (I + \alpha_n \partial g)^{-1}(x^n - \alpha_n \nabla f(x^n) + \beta_n(x^n - x^{n-1}))$$

- Applying the previous inequalities to the iteration yields the following result:

Lemma

(a) *The sequence* $(H_{\delta_n}(x^n, x^{n-1}))_{n=0}^\infty$ *is monotonically decreasing and thus converging. In particular, it holds*

$$H_{\delta_{n+1}}(x^{n+1}, x^n) \leq H_{\delta_n}(x^n, x^{n-1}) - \gamma_n \Delta_n^2\,,$$

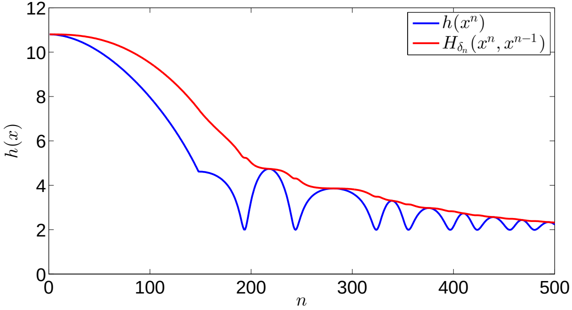
where γ_n, δ_n *is some pos. parameter depending on* α_n, β_n.

(b) *It holds* $\sum_{n=0}^\infty \Delta_n^2 < \infty$ *and, thus,* $\lim_{n\to\infty} \Delta_n = 0$.

Note that from $\lim_{n\to\infty} \Delta_n = 0 \not\Rightarrow \sum_{n=0}^\infty \Delta_n < \infty$, *e.g choose* $\Delta_n = 1/n$

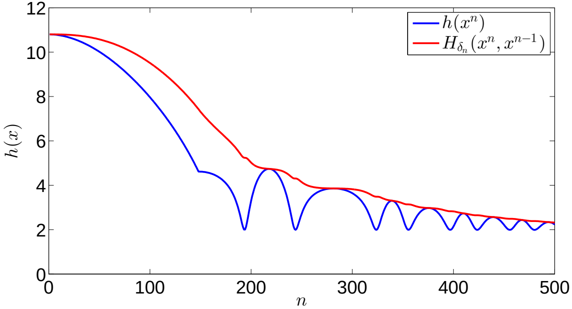
Discussion

- We do not guarantee monotone decrease of the function values $h(x^n)$ but we guarantee monotone decrease of the function $H_\delta(x, y) := h(x) + \delta\|x - y\|_2^2$



Discussion

- We do not guarantee monotone decrease of the function values $h(x^n)$ but we guarantee monotone decrease of the function $H_\delta(x, y) := h(x) + \delta\|x - y\|_2^2$



- To ensure convergence we obtain: $\alpha_n < \frac{2(1-\beta_n)}{L_n}$, which is the same as in [Zavriev, Kostyuk '93] for $g = 0$

Convergence of a subsequence

Based on the previous lemma we can draw our first conclusion about the convergence of the algorithm in the general case (no KL)

Theorem

(a) *The sequence* $(h(x^n))_{n=0}^{\infty}$ *converges.*

(b) *There exists a converging subsequence* $(x^{n_k})_{k=0}^{\infty}$.

(c) *Any limit point* $x^* := \lim_{k\to\infty} x^{n_k}$ *is a critical point of* h.

Convergence of a subsequence

Based on the previous lemma we can draw our first conclusion about the convergence of the algorithm in the general case (no KL)

Theorem

(a) *The sequence* $(h(x^n))_{n=0}^{\infty}$ *converges.*

(b) *There exists a converging subsequence* $(x^{n_k})_{k=0}^{\infty}$.

(c) *Any limit point* $x^* := \lim_{k\to\infty} x^{n_k}$ *is a critical point of* h.

- (a) follows from the fact that we can "sandwich" $h(x^n)$ between $H_{-\delta_n}(x^n, x^{n-1})$ and $H_{\delta_n}(x^n, x^{n-1})$

Convergence of a subsequence

Based on the previous lemma we can draw our first conclusion about the convergence of the algorithm in the general case (no KL)

Theorem

(a) *The sequence* $(h(x^n))_{n=0}^{\infty}$ *converges.*

(b) *There exists a converging subsequence* $(x^{n_k})_{k=0}^{\infty}$.

(c) *Any limit point* $x^* := \lim_{k\to\infty} x^{n_k}$ *is a critical point of* h.

- (a) follows from the fact that we can "sandwich" $h(x^n)$ between $H_{-\delta_n}(x^n, x^{n-1})$ and $H_{\delta_n}(x^n, x^{n-1})$
- (b) follows from the boundedness of the level sets of h and the Bolzano Weierstrass theorem

Convergence of a subsequence

Based on the previous lemma we can draw our first conclusion about the convergence of the algorithm in the general case (no KL)

Theorem

(a) *The sequence* $(h(x^n))_{n=0}^{\infty}$ *converges.*

(b) *There exists a converging subsequence* $(x^{n_k})_{k=0}^{\infty}$.

(c) *Any limit point* $x^* := \lim_{k\to\infty} x^{n_k}$ *is a critical point of* h.

- (a) follows from the fact that we can "sandwich" $h(x^n)$ between $H_{-\delta_n}(x^n, x^{n-1})$ and $H_{\delta_n}(x^n, x^{n-1})$
- (b) follows from the boundedness of the level sets of h and the Bolzano Weierstrass theorem
- (c) follows from the Lipschitz continuity of ∇f and the lower semi-continuity of g

Convergence of the whole sequence

Theorem

Let $(x^n)_{n\in\mathbb{N}}$ *be generated by the iPiano Algorithm, and let* $\delta_n = \delta$ *for all* $n \in \mathbb{N}$*. Then, the sequence* $(x^{n+1}, x^n)_{n\in\mathbb{N}}$ *satisfies H1, H2, and H3 for the function* $H_\delta(x, y)$*. Moreover, if* $H_\delta(x, y)$ *has the Kurdyka-Łojasiewicz property at a cluster point* (x^*, x^*)*, then the sequence* $(x^n)_{n\in\mathbb{N}}$ *has finite length,* $x^n \to x^*$ *as* $n \to \infty$*, and* (x^*, x^*) *is a critical point of* H_δ*, hence* x^* *is a critical point of* h*.*

Convergence of the whole sequence

Theorem

Let $(x^n)_{n\in\mathbb{N}}$ be generated by the iPiano Algorithm, and let $\delta_n = \delta$ for all $n \in \mathbb{N}$. Then, the sequence $(x^{n+1}, x^n)_{n\in\mathbb{N}}$ satisfies H1, H2, and H3 for the function $H_\delta(x, y)$. Moreover, if $H_\delta(x, y)$ has the Kurdyka-Łojasiewicz property at a cluster point (x^*, x^*), then the sequence $(x^n)_{n\in\mathbb{N}}$ has finite length, $x^n \to x^*$ as $n \to \infty$, and (x^*, x^*) is a critical point of H_δ, hence x^* is a critical point of h.

(H1) follows from the monotone decrease of H_δ

$$H_{\delta_{n+1}}(x^{n+1}, x^n) \leq H_{\delta_n}(x^n, x^{n-1}) - \gamma_n \Delta_n^2 .$$

Convergence of the whole sequence

Theorem

Let $(x^n)_{n\in\mathbb{N}}$ be generated by the iPiano Algorithm, and let $\delta_n = \delta$ for all $n \in \mathbb{N}$. Then, the sequence $(x^{n+1}, x^n)_{n\in\mathbb{N}}$ satisfies H1, H2, and H3 for the function $H_\delta(x, y)$. Moreover, if $H_\delta(x, y)$ has the Kurdyka-Łojasiewicz property at a cluster point (x^*, x^*), then the sequence $(x^n)_{n\in\mathbb{N}}$ has finite length, $x^n \to x^*$ as $n \to \infty$, and (x^*, x^*) is a critical point of H_δ, hence x^* is a critical point of h.

(H1) follows from the monotone decrease of H_δ

$$H_{\delta_{n+1}}(x^{n+1}, x^n) \le H_{\delta_n}(x^n, x^{n-1}) - \gamma_n \Delta_n^2 \,.$$

(H2) follows from the subdifferential of H_δ

$$\|w^{n+1}\|_2 \quad \le \quad \tfrac{1}{\alpha_n}(\alpha_n L_n + 1 + 4\alpha_n\delta)\Delta_{n+1} + \tfrac{1}{\alpha_n}\beta_n\Delta_n \,.$$

Convergence of the whole sequence

Theorem

Let $(x^n)_{n\in\mathbb{N}}$ be generated by the iPiano Algorithm, and let $\delta_n = \delta$ for all $n \in \mathbb{N}$. Then, the sequence $(x^{n+1}, x^n)_{n\in\mathbb{N}}$ satisfies H1, H2, and H3 for the function $H_\delta(x, y)$. Moreover, if $H_\delta(x, y)$ has the Kurdyka-Łojasiewicz property at a cluster point (x^*, x^*), then the sequence $(x^n)_{n\in\mathbb{N}}$ has finite length, $x^n \to x^*$ as $n \to \infty$, and (x^*, x^*) is a critical point of H_δ, hence x^* is a critical point of h.

(H1) follows from the monotone decrease of H_δ

$$H_{\delta_{n+1}}(x^{n+1}, x^n) \le H_{\delta_n}(x^n, x^{n-1}) - \gamma_n \Delta_n^2\,.$$

(H2) follows from the subdifferential of H_δ

$$\|w^{n+1}\|_2 \quad \le \quad \tfrac{1}{\alpha_n}(\alpha_n L_n + 1 + 4\alpha_n\delta)\Delta_{n+1} + \tfrac{1}{\alpha_n}\beta_n\Delta_n\,.$$

(H3) follows from the convergence of a subsequence of (x^n) and the fact that $\Delta_n \to 0$ as $n \to \infty$.

Convergence rate in the non-convex case

- Absence of convexity makes live hard

Convergence rate in the non-convex case

- Absence of convexity makes live hard

Theorem

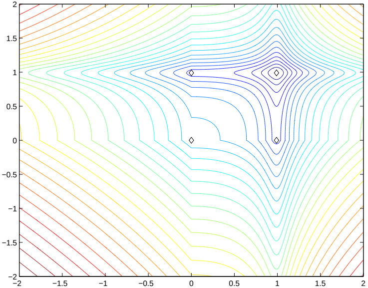
The iPiano algorithm guarantees that for all $N \geq 0$

$$\min_{0 \leq n \leq N} \|r(x^n)\|_2 \leq \frac{2}{c_1 c_2} \sqrt{\frac{h(x^0) - \underline{h}}{N+1}}$$

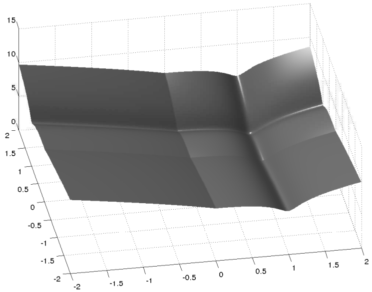
i.e. the smallest proximal residual converges with rate $\mathcal{O}(1/\sqrt{N})$.

- Similar bound for $\beta = 0$ is shown in [Nesterov '12]

Ability to overcome spurious stationary solutions



(a) Contour plot of $h(x)$



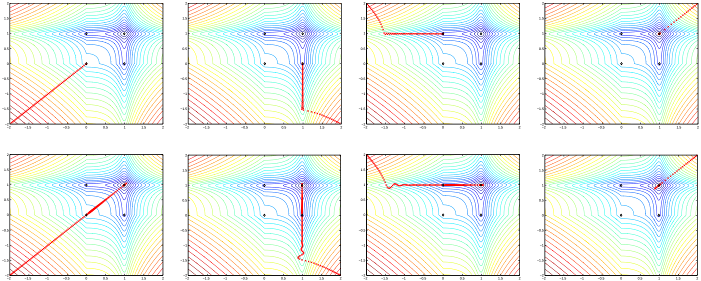
(b) Energy landscape of $h(x)$

$$\min_{x \in \mathbb{R}^N} h(x) := f(x) + g(x)\,, \quad f(x) = \frac{1}{2}\sum_{i=1}^{N} \log(1 + \mu(x_i - u_i^0)^2)\,, \quad g(x) = \lambda \|x\|_1\,,$$

Effect of the inertial force

$\beta = 0.00$:

$\beta = 0.75$:



The inertial force helps to overcome spurious stationary solutions

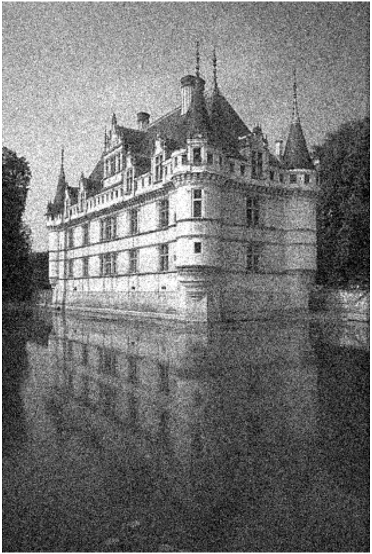
Application to student-t regularized image denoising

- We consider the following class of non-convex image denoising models

$$\min_{u \in \mathbb{R}^N} \sum_{i=1}^{N_f} \vartheta_i \sum_j \varphi((K_i u)_j) + \frac{\lambda}{p} \|u - u^0\|_p^p\,, \quad p \in \{1, 2\}$$

- The potential functions are given by $\varphi(t) = \log(1 + t^2)$
- Obvious splitting into a smooth function plus a convex function with easy to compute proximal map
- The linear operators K_i are given by learned filter kernels k_i
- Gives excellent results for image denoising [Chen et al. '13]
- Comparison based on the error $\mathcal{E}^n = h^n - h^*$
- In this example, h^* appears to be the same for all tested algorithms (which is not true in general).

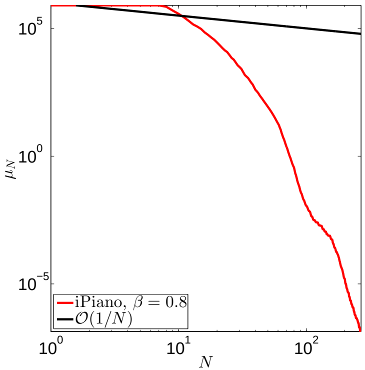
Results for ℓ_2 denoising



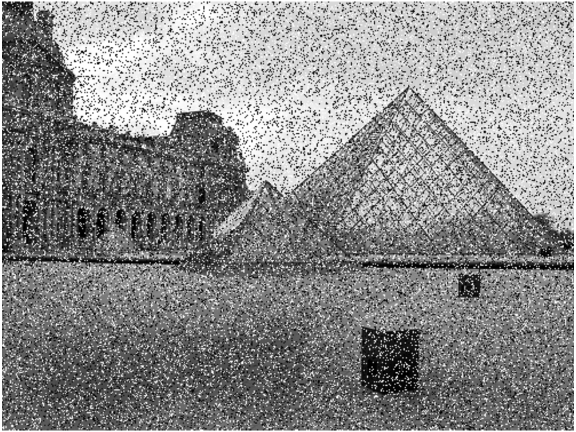
Results for ℓ_2 denoising

Results for ℓ_2 denoising

	iPiano with different β							L-BFGS	
tol	0.00	0.20	0.40	0.60	0.80	0.95	T_1(s)	iter.	T_2(s)
10^1	505	344	222	129	79	299	47.177	66	27.054
10^0	664	451	290	168	98	342	59.133	79	32.143
10^{-1}	857	579	371	216	143	384	85.784	93	36.926
10^{-2}	1086	730	468	271	173	427	103.436	107	41.939
10^{-3}	1347	904	577	338	199	473	119.149	124	48.272



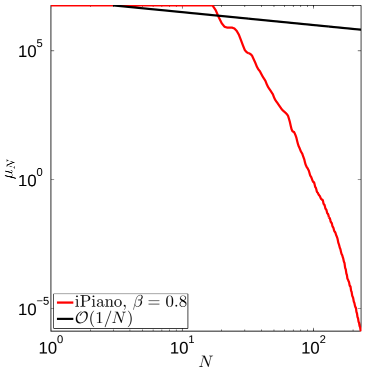
Results for ℓ_1 data term



Results for ℓ_1 data term

Results for ℓ_1 data term

	iPiano with different β							L-BFGS	
tol	0.00	0.20	0.40	0.60	0.80	0.95	T_1(s)	iter.	T_2(s)
10^1	847	538	341	195	96	304	65.679	265	121.303
10^0	1077	682	433	247	120	349	81.761	285	130.846
10^{-1}	1311	835	530	303	143	395	97.060	298	136.326
10^{-2}	1559	997	631	362	164	440	111.579	311	141.876
10^{-3}	1818	1169	741	424	185	485	126.272	327	148.945



Application to image compression based on linear diffusion

- A new image compression methodology introduced in [Galic, Weickert, Welk, Bruhn, Belyaev, Seidel '08]
- The idea is to select a subset of image pixels such that the reconstruction of the whole image via linear diffusion yields the best reconstruction [Hoeltgen, Setzer, Weickert '13]

Application to image compression based on linear diffusion

- Is written as the following bilevel optimization problem

$$\min_{u,c} \frac{1}{2}\|u - u^0\|_2^2 + \lambda\|c\|_1$$
$$\text{s.t. } C(u - u^0) - (I - C)Lu = 0\,,$$

  where $C = \mathrm{diag}(c) \in \mathbb{R}^{N\times N}$ and L is the Laplace or biharmonic operator

- We can transform the problem into an non-convex single-level problem of the form

$$\min_{c} \frac{1}{2}\|A^{-1}Cu^0 - u^0\|_2^2 + \lambda\|c\|_1\,, \quad A = C + (C - I)L$$

- Perfectly fits to the framework of iPiano
- We choose $f = \frac{1}{2}\|A^{-1}Cu^0 - u^0\|_2^2$ and $g = \lambda\|c\|_1$
- The gradient of f is given by

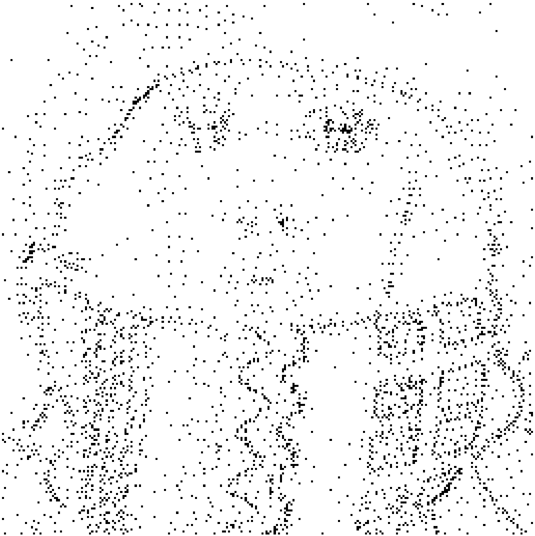
$$\nabla f(c) = \mathrm{diag}(-(I+L)u + u^0)(A^\top)^{-1}(u - u^0)\,, \quad u = A^{-1}Cu^0$$

- Lipschitz, if at least one entry of c is non-zero
- One evaluation of the gradient requires to solve two linear systems
- Proximal map with respect to g is standard

Results for Trui

Input

Results for Trui



5% of the pixels

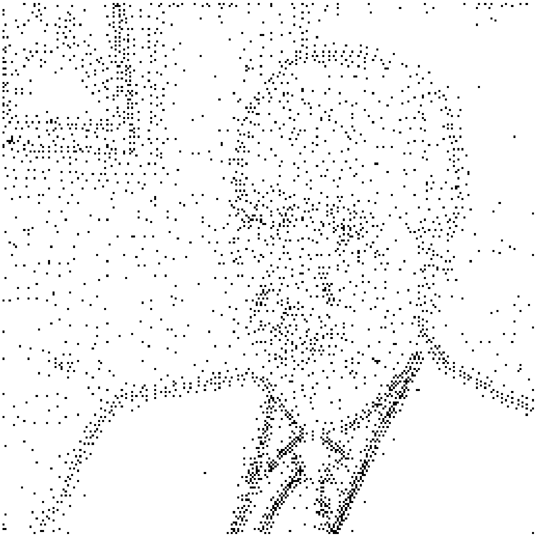
Results for Trui

Reconstruction

Results for Walter

Input

Results for Walter



5% of the pixels

Results for Walter

Reconstruction

Phase field models

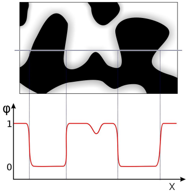
- Mathematical model for solving interfacial problems
- Approximation of the interface length via the Mordica-Mortola phase field energy

$$\int_\Gamma \,\mathrm{d}\gamma \approx \int_\Omega \frac{\varepsilon}{2}|\nabla u|^2 + \frac{1}{\varepsilon}W(u)\,\mathrm{d}x\,,$$

where $W(t) = (t(1-t))^2/2$ is a double-well potential

Phase field models

- Mathematical model for solving interfacial problems
- Approximation of the interface length via the Mordica-Mortola phase field energy
$$\int_\Gamma \mathrm{d}\gamma \approx \int_\Omega \frac{\varepsilon}{2}|\nabla u|^2 + \frac{1}{\varepsilon}W(u)\,\mathrm{d}x\,,$$
where $W(t) = (t(1-t))^2/2$ is a double-well potential
- Non-convex, but smooth energy
- Can be combined with arbirtrary non-smooth but convex energy



(Source: wikipedia)

Videos ...

Curvature

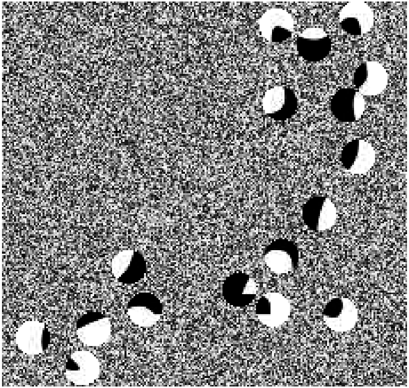
- Phase-fields are close to distance functions around the interface and hence they allow to reliably estimate the curvature of the interface
- Approximation of the Willmore energy

$$\frac{1}{2}\int_\Gamma h^2 \,\mathrm{d}\gamma \approx \frac{1}{2\varepsilon}\int_\Omega (\Delta u - \frac{1}{\varepsilon} W'(u))^2 \,\mathrm{d}x$$

- De Giorgi conjecture: Γ-convergence as $\varepsilon \to 0$
- Length vs. curvature regularization

Videos ...

Image inpainting



Input image

Conclusion

- Proposed an inertial forward-backward algorithm (iPiano) for minimizing the sum of a smooth and a convex function
- Existence of a converging subsequence in the most-general case
- Convergence of the whole sequence in case the Kurdyka-Łojasiewicz property holds
- $\mathcal{O}(1/\sqrt{N})$ convergence of the proximal residual in the general case
- Application to non-convex problems in image processing
- Can be easily parallelized or implemented in mobile hardware

Thank you for listening!