*Convex Relaxation in Variational Image Analysis*
*Short Course, HCI*, WS 2011

**Chapter 3**

# Optimization

## Saddle point problems and an optimal first order primal-dual algorithm

Bastian Goldlücke
Computer Vision Group
Technical University of Munich

# Overview

# Overview

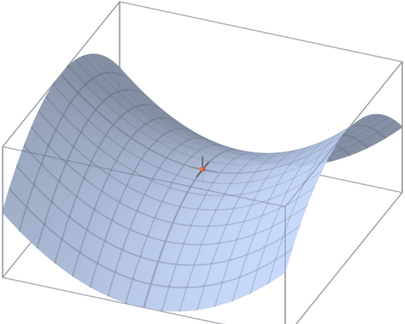
# Saddle points

## Definition

Let $E : \mathcal{V} \times \mathcal{W} \to \mathbb{R} \cup \{\infty\}$ be a functional on $\mathcal{V} \times \mathcal{W}$, where $\mathcal{V}$ and $\mathcal{W}$ are vector spaces. A point $(\hat{x}, \hat{y}) \in \mathcal{V} \times \mathcal{W}$ is called a saddle point of $E$ if

$$\hat{x} \in \operatorname*{argmin}_{x \in \mathcal{V}} E(x, \hat{y})$$

$$\text{and } \hat{y} \in \operatorname*{argmax}_{y \in \mathcal{W}} E(\hat{x}, y).$$



**Example:**

$(0, 0)$ is a saddle point of

$$f(x, y) = x^2 - y^2 \text{ on } \mathbb{R} \times \mathbb{R}.$$

## Min-max problems



The following is a less obvious characterization of a saddle point.

### Proposition

The point $(\hat{x}, \hat{y})$ is a saddle point of $E$ if and only if

$$\min_{x \in \mathcal{V}} \max_{y \in \mathcal{W}} E(x, y) = \max_{y \in \mathcal{W}} \min_{x \in \mathcal{V}} E(x, y) = E(\hat{x}, \hat{y}).$$

Note that in general, min and max do not commute (find a counterexample as an exercise). The proof is again elementary, just note that obviously for all $(x_0, y_0)$,

$$\max_{y \in \mathcal{W}} E(x_0, y) \geq E(x_0, y_0) \geq \min_{x \in \mathcal{V}} E(x, y_0).$$

This inequality still holds when we move to min over $x_0$ and max over $y_0$ on the left and right, so together with the saddle point definition it follows that

$$E(\hat{x}, \hat{y}) \geq \min_{x \in \mathcal{V}} \max_{y \in \mathcal{W}} E(x, y) \geq \max_{y \in \mathcal{W}} \min_{x \in \mathcal{V}} E(x, y) \geq E(\hat{x}, \hat{y}).$$

# Saddle point formulation of the archetypical proble

All problems we will consider have the following form, which is why this theorem is central to our examinations. Note that we have already discussed a special case in exercise 2.1.

**Theorem**

Let $F : \mathcal{W} \to \mathbb{R}$ be a closed and convex functional on the reflexive space $\mathcal{W}$, $G$ a closed and convex functional on the reflexive space $\mathcal{V}$ and let $K : \mathcal{V} \to \mathcal{W}$ be a continuous linear operator. Then

$$
\begin{aligned}
&\min_{x\in\mathcal{V}} F(Kx) + G(x) && \text{primal} \\
=&\min_{x\in\mathcal{V}} \max_{\varphi\in\mathcal{W}^*} \langle Kx, \varphi\rangle - F^*(\varphi) + G(x) && \text{primal-dual} \\
=&\max_{\varphi\in\mathcal{W}^*} \min_{x\in\mathcal{V}} \langle x, K^*\varphi\rangle - F^*(\varphi) + G(x) && \text{dual-primal} \\
=&\max_{\varphi\in\mathcal{W}^*} -(F^*(\varphi) + G^*(-K^*\varphi)) && \text{dual},
\end{aligned}
$$

provided the primal-dual problem has a saddle point $(\hat{x}, \hat{\varphi})$. Note that in this case, $\hat{x}$ is the minimizer of the primal problem and $\hat{\varphi}$ is the maximizer of the dual problem, respectively.

# TV minimization as a saddle point problem



In the remainder of the chapter, we will consider the discrete scenario in the case $n = 2$ for simplicity of notation.

- An image $u$ is a matrix in $\mathbb{R}^{N \times M}$.
- A vector field $\boldsymbol{\xi}$ consists of two matrices $\xi^1, \xi^2 \in \mathbb{R}^{N \times M}$.
- The discrete gradient is the map

$$\nabla : \mathbb{R}^{N \times M} \to (\mathbb{R}^{N \times M})^2$$

  implemented with forward differences and Neumann boundary conditions.
- The discrete divergence is the map

$$\text{div} : (\mathbb{R}^{N \times M})^2 \to \mathbb{R}^{N \times M}$$

  implemented with backward differences and Dirichlet boundary conditions.
- The operators satisfy $(\nabla u, \boldsymbol{\xi}) = (u, -\text{div}(\boldsymbol{\xi}))$ for all $u, \boldsymbol{\xi}$, in particular $\nabla^* = -\text{div}$.

# Discrete total variation and conjugate



Remember that in the continuous case, the total variation can be written as $\int_\Omega |\nabla u|_2 \, dx$ for differentiable $u$. This corresponds to the discrete

$$J(u) = \sum_{i=1}^{N} \sum_{j=1}^{M} \left|(\nabla u)_{i,j}\right|_2 = F(Ku),$$

with the maps $K : \mathcal{V} \to \mathcal{W}$, $F : \mathcal{W} \to \mathbb{R}$ defined on the spaces $\mathcal{V} = \mathbb{R}^{N \times M}$, $\mathcal{W} = (\mathbb{R}^{N \times M})^2$ by

$$Ku := \nabla u$$

$$\text{and } F(\xi) := \sum_{i=1}^{N} \sum_{j=1}^{M} \left|\xi_{i,j}\right|_2 .$$

# The primal-dual of the TV-$\mathcal{L}^2$ model

We can now see that the discrete TV-$\mathcal{L}^2$ model is given by

$$\min_{u \in \mathcal{V}} \left\{ F(Ku) + \frac{1}{2\lambda} |u - f|_2^2 \right\},$$

with $K = \nabla$ and $F(\xi) = \sum_{i,j} \left|\xi_{i,j}\right|_2$.

Recall that the convex conjugate of the Euclidean norm on $\mathbb{R}^n$ is given by $|\cdot|_2^* = \delta_B$, where $B$ is the unit ball in $\mathbb{R}^n$. Thus,

$$F^*(\xi) = \delta_{\mathcal{B}}(\xi) \text{ with } \mathcal{B} := B^{N \times M} \subset \mathcal{W},$$

and the primal-dual formulation of TV-$\mathcal{L}^2$ is given by

$$\begin{aligned} &\min_{u \in \mathcal{V}} \max_{\xi \in \mathcal{W}} \left\{ \langle Ku, \xi \rangle + \frac{1}{2\lambda} |u - f|_2^2 - \delta_{\mathcal{B}}(\xi) \right\} \\ = &\min_{u \in \mathcal{V}} \max_{\xi \in \mathcal{B}} \left\{ \langle \nabla u, \xi \rangle + \frac{1}{2\lambda} |u - f|_2^2 \right\} \end{aligned}$$

## An advantage of the saddle formulation

If you solve the saddle point problem as opposed to the primal or dual problem, there is one immediate advantage: at each point during the solving process, you have an estimate how good the current solution is.

Suppose $(x, \varphi)$ is the current estimate for the solution, then the energy of the saddle point $(\hat{x}, \hat{\varphi})$ is somewhere between the primal and dual energies:

$$F(Kx) + G(x) \geq E(\hat{x}, \hat{\varphi}) \geq -F^*(\varphi^*) - G^*(-K^*\varphi).$$

The difference

$$\mathcal{G}(x, \varphi) := F(Kx) + G(x) + F^*(\varphi^*) + G^*(-K^*\varphi)$$

is called the primal-dual gap and tells you how far away the energy of the current estimate is from the energy of the actual solution.

# Overview

# Saddle point algorithm

The goal is to understand and be able to implement an efficient algorithm for saddle point problems described in the following paper:

A.Chambolle and T.Pock

*A first-order primal-dual algorithm for convex problems with applications to imaging*

Preprint, 2010, available online.

A first-order primal-dual algorithm for convex problems with applications to imaging

Antonin Chambolle[*] and Thomas Pock[†]

May 14, 2010

**Abstract**

In this paper we study a first-order primal-dual algorithm for convex optimization problems with known saddle-point structure. We prove convergence to a saddle-point with rate $O(1/N)$ in finite dimensions, which is optimal for the complete class of non-smooth problems we are considering in this paper. We further show accelerations of the proposed algorithm to yield optimal rates on easier problems. In particular we show that we can achieve $O(1/N^2)$ convergence on problems, where the primal or the dual objective is uniformly convex, and we can show linear convergence, i.e. $O(1/e^N)$ on problems where both are uniformly convex. The wide applicability of the proposed algorithm is demonstrated on several imaging problems such as image denoising, image deconvolution, image inpainting, motion estimation and image segmentation.

## 1 Introduction

Variational methods have proven to be particularly useful to solve a number of ill-posed inverse imaging problems. They can be divided into two fundamentally different classes: Convex and non-convex problems. The advantage of convex problems over non-convex problems is that a global optimum can be computed, in general with a good precision and in a reasonable time, independent of the initialization. Hence, the quality of the solution solely depends on the accuracy of the model. On the other hand, non-convex problems have often the ability to model more precisely the process behind an image acquisition, but have the drawback that the quality of the solution is more sensitive to the initialization and the optimization algorithm.

Total variation minimization plays an important role in convex variational methods for imaging. The major advantage of the total variation is that it

[*] CMAP, Ecole Polytechnique, CNRS, 91128, Palaiseau, France.
e-mail: antonin.chambolle@polytechnique.fr

[†] Institute for Computer Graphics and Vision, Graz University of Technology, 8010 Graz, Austria.
e-mail pock@icg.tugraz.at

1

# Reminder: problem formulation

## Theorem

Let $F : \mathcal{W} \to \mathbb{R}$ be a closed and convex functional on the reflexive space $\mathcal{W}$, $G$ a closed and convex functional on the reflexive space $\mathcal{V}$ and let $K : \mathcal{V} \to \mathcal{W}$ be a continuous linear operator. Then

$$\begin{aligned}
&\min_{x \in \mathcal{V}} F(Kx) + G(x) && \text{primal} \\
=&\min_{x \in \mathcal{V}} \max_{\varphi \in \mathcal{W}^*} \langle Kx, \varphi \rangle - F^*(\varphi) + G(x) && \text{primal-dual} \\
=&\max_{\varphi \in \mathcal{W}^*} \min_{x \in \mathcal{V}} \langle x, K^*\varphi \rangle - F^*(\varphi) + G(x) && \text{dual-primal} \\
=&\max_{\varphi \in \mathcal{W}^*} -\left(F^*(\varphi) + G^*(-K^*\varphi)\right) && \text{dual},
\end{aligned}$$

provided the primal-dual problem has a saddle point $(\hat{x}, \hat{\varphi})$. Note that in this case, $\hat{x}$ is the minimizer of the primal problem and $\hat{\varphi}$ is the maximizer of the dual problem, respectively.

# Algorithm 1

Algorithm 1 is designed to find the saddle point of the primal-dual problem

$$\min_{x \in \mathcal{V}} \max_{\varphi \in \mathcal{W}^*} \langle Kx, \varphi \rangle - F^*(\varphi) + G(x).$$

## Algorithm 1 (in the paper)

- Initialization:

  Initial values $x_0, \varphi_0$ arbitrary and $\bar{x}_0 = x_0$

  Step sizes $\sigma, \tau > 0$ and $\theta \in [0, 1]$.

- Iteration:

$$\begin{aligned} \varphi_{n+1} &= \operatorname{prox}_{\sigma F^*}(\varphi_n + \sigma K \bar{x}_n) \\ x_{n+1} &= \operatorname{prox}_{\tau G}(x_n - \tau K^* \varphi_{n+1}) \\ \bar{x}_{n+1} &= x_{n+1} + \theta(x_{n+1} - x_n). \end{aligned}$$

The algorithm has guaranteed convergence under certain conditions which we will see later.

# Interpretation of algorithm I

We want to solve the primal dual problem

$$\min_{x \in \mathcal{V}} \max_{\varphi \in \mathcal{W}^*} \langle Kx, \varphi \rangle - F^*(\varphi) + G(x).$$

Suppose we have an iterate $(x_n, \varphi_n)$ - how can we get closer to the solution?

# Interpretation of algorithm I

We want to solve the primal dual problem

$$\min_{x \in \mathcal{V}} \max_{\varphi \in \mathcal{W}^*} \langle Kx, \varphi \rangle - F^*(\varphi) + G(x).$$

Suppose we have an iterate $(x_n, \varphi_n)$ - how can we get closer to the solution?

## Ascent in $\varphi$

Gradient ascent in $\varphi$ for the differentiable part yields

$$\tilde{\varphi}_{n+1} = \varphi_n + \sigma K x_n.$$

Combined with subgradient descent in the non-differentiable $F^*$ this yields the iteration

$$\varphi_{n+1} = \operatorname{prox}_{\sigma F^*}(\tilde{\varphi}_{n+1}) = \operatorname{prox}_{\sigma F^*}(\varphi_n + \sigma K x_n).$$

# Interpretation of algorithm I

We want to solve the primal dual problem

$$\min_{x \in \mathcal{V}} \max_{\varphi \in \mathcal{W}^*} \langle Kx, \varphi \rangle - F^*(\varphi) + G(x).$$

Suppose we have an iterate $(x_n, \varphi_n)$ - how can we get closer to the solution?

## Descent in $x$

Gradient descent in $x$ for the differentiable part yields

$$\tilde{x}_{n+1} = x_n - \tau K^* \varphi_{n+1}.$$

Combined with subgradient descent in the non-differentiable $G$ this yields the iteration

$$x_{n+1} = \mathrm{prox}_{\tau G}(\tilde{x}_{n+1}) = \mathrm{prox}_{\tau G}(x_n + \tau K^* \varphi_{n+1}).$$

# Interpretation of algorithm I

We want to solve the primal dual problem

$$\min_{x \in \mathcal{V}} \max_{\varphi \in \mathcal{W}^*} \langle Kx, \varphi \rangle - F^*(\varphi) + G(x).$$

Suppose we have an iterate $(x_n, \varphi_n)$ - how can we get closer to the solution?

**Algorithm 0 (Arrow-Hurwicz method)**

Start with arbitrary $(x_0, \varphi_0)$, choose "small enough" step sizes $\sigma, \tau > 0$. Iterate

$$\begin{aligned} \varphi_{n+1} &= \operatorname{prox}_{\sigma F^*}(\varphi_n + \sigma K x_n) \\ x_{n+1} &= \operatorname{prox}_{\tau G}(x_n - \tau K^* \varphi_{n+1}). \end{aligned}$$

This is algorithm 1 for the special case $\theta = 0$. This special case is the classical "Arrow-Hurwicz method", dating back as far as 1958.

# Algorithm I for the ROF (TV-$\mathcal{L}^2$) model

Reminder: The primal-dual formulation of ROF is

$$\min_u \max_{\boldsymbol{\xi}} \langle \nabla u, \boldsymbol{\xi} \rangle - \underbrace{\delta_{|\cdot|_2 \leq 1}}_{F^*(\boldsymbol{\xi})} + \underbrace{\frac{1}{2\lambda} \|u - f\|^2}_{G(u)}.$$

# Algorithm I for the ROF (TV-$\mathcal{L}^2$) model

Reminder: The primal-dual formulation of ROF is

$$\min_u \max_{\boldsymbol{\xi}} \langle \nabla u, \boldsymbol{\xi} \rangle - \underbrace{\delta_{|\cdot|_2 \leq 1}}_{F^*(\boldsymbol{\xi})} + \underbrace{\frac{1}{2\lambda} \|u - f\|^2}_{G(u)}.$$

## Proximation for $F^*$

$$\operatorname{prox}_{\sigma F^*}(\boldsymbol{\eta}) = \operatorname*{argmin}_{\boldsymbol{\xi}} \left\{ \frac{\|\boldsymbol{\xi} - \boldsymbol{\eta}\|^2}{2\sigma} + \delta_{|\cdot|_2 \leq 1}(\boldsymbol{\xi}) \right\} = \operatorname*{argmin}_{|\boldsymbol{\xi}|_2 \leq 1} \|\boldsymbol{\xi} - \boldsymbol{\eta}\|$$

is just the projection onto the unit ball $\{\boldsymbol{\xi} \ : \ |\boldsymbol{\xi}|_2 \leq 1\}$. It can be computed point-wise via

$$\operatorname{prox}_{\sigma F^*}(\boldsymbol{\eta}) = \frac{\boldsymbol{\eta}}{\max(1, |\boldsymbol{\eta}|_2)}.$$

# Algorithm I for the ROF (TV-$\mathcal{L}^2$) model

Reminder: The primal-dual formulation of ROF is

$$\min_u \max_{\xi} \langle \nabla u, \xi \rangle - \underbrace{\delta_{|\cdot|_2 \leq 1}}_{F^*(\xi)} + \underbrace{\frac{1}{2\lambda} \|u - f\|^2}_{G(u)}.$$

## Proximation for $G$

$$\mathrm{prox}_{\tau G}(v) = \operatorname*{argmin}_u \left\{ \frac{\|u - v\|^2}{2\tau} + \frac{1}{2\lambda} \|u - f\|^2 \right\}$$

can be computed point-wise by setting the derivative to zero and solving for $u$. One gets

$$\mathrm{prox}_{\tau G^*}(v) = \frac{\lambda v + \tau f}{\tau + \lambda}.$$

# Algorithm I for the ROF (TV-$\mathcal{L}^2$) model

Reminder: The primal-dual formulation of ROF is

$$\min_u \max_\xi \langle \nabla u, \xi \rangle - \underbrace{\delta_{|\cdot|_2 \leq 1}}_{F^*(\xi)} + \underbrace{\frac{1}{2\lambda} \|u - f\|^2}_{G(u)}.$$

## Algorithm I for TV-$\mathcal{L}^2$

Set $u_0 = 0, \xi_0 = 0, \sigma = \tau = \frac{1}{\sqrt{8}}, \bar{u}_0 = u_0$. Then iterate

$$\begin{aligned}
\tilde{\xi}_{n+1} &= \xi_n + \sigma \nabla \bar{u}_n \\
\xi_{n+1} &= \tilde{\xi}_{n+1} / \max(1, |\tilde{\xi}_{n+1}|_2) \\
u_{n+1} &= \frac{\lambda u_n + \lambda \tau \operatorname{div}(\xi_{n+1}) + \tau f}{\lambda + \tau} \\
\bar{u}_{n+1} &= 2u_{n+1} - u_n.
\end{aligned}$$

# Algorithm I for general inverse problems



Reminder: The primal-dual formulation of a general inverse problem is given by

$$\min_u \max_\xi \langle \nabla u, \xi \rangle - \delta_{|\cdot|_2 \leq 1} + \frac{1}{2\lambda} \left\| Au - f \right\|^2,$$

with an additional operator $A$ on $\mathcal{L}^2(\Omega)$.

A convenient reformulation can be obtained by “dualizing” the norm in the data term, i.e. replacing $\frac{1}{2\lambda} \left\| Au - f \right\|^2$ with its second conjugate. This introduces a second dual variable $q : \Omega \to \mathbb{R}$, and one gets

$$\min_u \max_{\xi, q} \langle \nabla u, \xi \rangle + \langle Au - f, q \rangle - \underbrace{\delta_{|\cdot|_2 \leq 1} + \frac{\lambda}{2} |q|_2^2}_{F^*(\xi)}.$$

# Algorithm I for general inverse problems

The primal-dual formulation of the general inverse problem is

$$\min_{u} \max_{\xi, q} \langle \nabla u, \xi \rangle + \langle Au - f, q \rangle - \underbrace{\delta_{|\cdot|_2 \leq 1} + \frac{\lambda}{2} |q|_2^2}_{F^*(\xi)} .$$

# Algorithm I for general inverse problems



The primal-dual formulation of the general inverse problem is

$$\min_u \max_{\xi,q} \langle \nabla u, \xi \rangle + \langle Au - f, q \rangle - \underbrace{\delta_{|\cdot|_2 \le 1} + \frac{\lambda}{2} |q|_2^2}_{F^*(\xi)} .$$

## Proximation for $F^*$

The proximation can be computed separately for $\xi$ and $q$. For $\xi$ we get the same result as for ROF. For $q$, we need to compute

$$\operatorname*{argmin}_q \left\{ \frac{\|q - \tilde{q}\|^2}{2\sigma} + \frac{\lambda}{2} \|q\|^2 \right\} .$$

Setting again the derivative to zero, one obtains the location of the minimum as

$$\hat{q} = \frac{\tilde{q}}{1 + \lambda\sigma} .$$

# Algorithm I for general inverse problems



The primal-dual formulation of the general inverse problem is

$$\min_{u} \max_{\xi, q} \langle \nabla u, \xi \rangle + \langle Au - f, q \rangle - \underbrace{\delta_{|\cdot|_2 \leq 1} + \frac{\lambda}{2} |q|_2^2}_{F^*(\xi)}.$$

## Proximation for $G$

Since $G = 0$ and $u$ is unconstrained, the proximation for $G$ is the identity.

# Algorithm I for general inverse problems



The primal-dual formulation of the general inverse problem is

$$\min_u \max_{\xi,q} \langle \nabla u, \xi \rangle + \langle Au - f, q \rangle - \underbrace{\delta_{|\cdot|_2 \leq 1} + \frac{\lambda}{2} |q|_2^2}_{F^*(\xi)} .$$

## Algorithm I for the general inverse problem

Set $u_0 = 0, \xi_0 = 0, q_0 = 0, \sigma = \tau = \frac{1}{\max(\sqrt{8}, \|A\|)}, \bar{u}_0 = u_0$. Then iterate

$$\begin{aligned}
\tilde{\xi}_{n+1} &= \xi_n + \sigma \nabla \bar{u}_n, \\
\tilde{q}_{n+1} &= q_n + \sigma (A\bar{u}_n - f) \\
\xi_{n+1} &= \tilde{\xi}_{n+1} / \max(1, |\tilde{\xi}_{n+1}|_2), \\
q_{n+1} &= \tilde{q}_{n+1} / (1 + \sigma\lambda) \\
u_{n+1} &= u_n + \tau(\operatorname{div}(\xi_{n+1}) - A^* q) \\
\bar{u}_{n+1} &= 2u_{n+1} - u_n.
\end{aligned}$$

# Algorithm I for linear data terms

Reminder: The primal-dual formulation of the linear model is

$$\min_u \max_{\boldsymbol{\xi}} \langle \nabla u, \boldsymbol{\xi} \rangle - \underbrace{\delta_{|\cdot|_2 \leq 1}}_{F^*(\boldsymbol{\xi})} + \underbrace{\langle a, u \rangle}_{G(u)},$$

with a weight function $a \in \mathcal{L}^2 \Omega$.

# Algorithm I for linear data terms

Reminder: The primal-dual formulation of the linear model is

$$\min_u \max_{\boldsymbol{\xi}} \langle \nabla u, \boldsymbol{\xi} \rangle - \underbrace{\delta_{|\cdot|_2 \leq 1}}_{F^*(\boldsymbol{\xi})} + \underbrace{\langle a, u \rangle}_{G(u)},$$

with a weight function $a \in \mathcal{L}^2\Omega$.

## Proximation for $F^*$

The proximation for $F^*$ is the same as for the ROF model.

# Algorithm I for linear data terms

Reminder: The primal-dual formulation of the linear model is

$$\min_u \max_{\xi} \langle \nabla u, \xi \rangle - \underbrace{\delta_{|\cdot|_2 \leq 1}}_{F^*(\xi)} + \underbrace{\langle a, u \rangle}_{G(u)},$$

with a weight function $a \in \mathcal{L}^2\Omega$.

## Proximation for $G$

$$\text{prox}_{\tau G}(v) = \underset{u}{\text{argmin}} \left\{ \frac{\|u - v\|^2}{2\tau} + \langle a, u \rangle \right\}$$

can be computed point-wise by setting the derivative to zero and solving for $u$. One gets

$$\text{prox}_{\tau G^*}(v) = v + \tau a.$$

# Algorithm I for linear data terms

Reminder: The primal-dual formulation of the linear model is

$$\min_u \max_\xi \langle \nabla u, \xi \rangle - \underbrace{\delta_{|\cdot|_2 \leq 1}}_{F^*(\xi)} + \underbrace{\langle a, u \rangle}_{G(u)},$$

with a weight function $a \in \mathcal{L}^2\Omega$.

## Algorithm I for the linear model

Set $u_0 = 0, \xi_0 = 0, \sigma = \tau = \frac{1}{\sqrt{8}}, \bar{u}_0 = u_0$. Then iterate

$$\begin{aligned}
\tilde{\xi}_{n+1} &= \xi_n + \sigma \nabla \bar{u}_n \\
\xi_{n+1} &= \tilde{\xi}_{n+1} / \max(1, |\tilde{\xi}_{n+1}|_2) \\
u_{n+1} &= u_n + \tau(\operatorname{div}(\xi_{n+1}) + a) \\
\bar{u}_{n+1} &= 2u_{n+1} - u_n.
\end{aligned}$$

# Convergence theorem for Algorithm I

Theorem 1 from the Chambolle/Pock paper proves convergence of algorithm 1.

**Convergence of algorithm I**

Assume that the saddle point problem has at least one solution. Let the sequence $(x_n, y_n)$ be generated by Algorithm I, with $\sigma, \tau > 0$ chosen such that $\sigma\tau < \frac{1}{\|K\|^2}$ and $\theta = 1$. Then there exists a saddle point $(\hat{x}, \hat{y})$ such that

$$x_n \to \hat{x} \text{ and } y_n \to \hat{y}.$$

Note that the norm of the discrete gradient operator is $\|\nabla\| = \sqrt{8}$. Chambolle / Pock also show that convergence of the primal-dual gap to zero is of order $\mathcal{O}(1/n)$.

# Uniform convexity

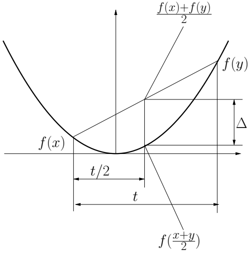
Convergence of algorithm 1 can be accelerated if $G$ is *uniformly convex*.

## Definition (uniform convex)

Let $G$ be convex and define for $t > 0$

$$\gamma_G(t) := \inf_{x,y \in \mathrm{dom}(G), \|x-y\|=t} \left\{ \Delta := \frac{G(x)+G(y)}{2} - G\left(\frac{x+y}{2}\right) \right\}.$$

The function $G$ is called uniform convex with modulus $\gamma > 0$ if $\gamma_G(t) > \gamma$ for all $t > 0$.




**Example**: $\frac{1}{2\lambda}\|\cdot\|^2$ is uniform convex with modulus $1/\lambda$.

# Acceleration

## Algorithm 2 (in the paper)

Algorithm 2 can be applied if $G$ is uniformly convex with modulus $\gamma$.

- Initialization:

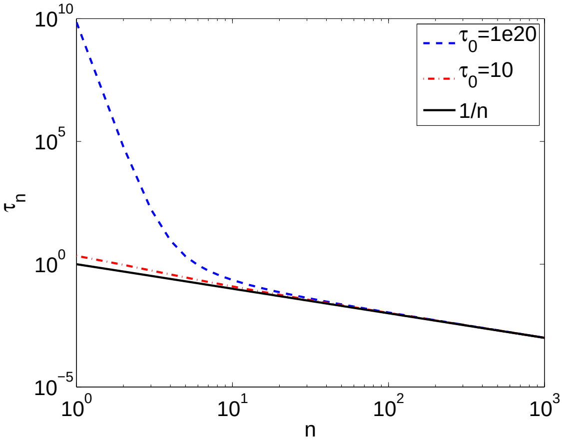
$$\text{Initial values } x_0, \varphi_0 \text{ arbitrary and } \bar{x}_0 = x_0$$

$$\text{Initial step sizes } \tau_0 > 0, \sigma_0 = \frac{1}{\tau_0 \|K\|^2}.$$

- Iteration:

$$\begin{aligned}
\varphi_{n+1} &= \mathrm{prox}_{\sigma_n F^*}(\varphi_n + \sigma_n K \bar{x}_n) \\
x_{n+1} &= \mathrm{prox}_{\tau_n G}(x_n - \tau_n K^* \varphi_{n+1}) \\
\theta_n &= \frac{1}{\sqrt{1 + 2\gamma\tau_n}}, \quad \tau_{n+1} = \theta_n \tau_n, \quad \sigma_{n+1} = \frac{\sigma_n}{\theta_n} \\
\bar{x}_{n+1} &= x_{n+1} + \theta_{n+1}(x_{n+1} - x_n).
\end{aligned}$$

# Behaviour of step sizes $\tau_n$




The sequence $(\tau_n)_{n\geq1}$ is almost independent from the initial $\tau_0$, and converges asymptotically to $1/n$.

# Convergence theorem for Algorithm 2

Theorem 2 from the Chambolle/Pock paper proves convergence of algorithm 2.

**Convergence of algorithm 2**

Assume that the saddle point problem has at least one solution. Let the sequence $(x_n, y_n)$ be generated by Algorithm 2. Then there exists a saddle point $(\hat{x}, \hat{y})$ such that

$$x_n \to \hat{x} \text{ and } y_n \to \hat{y}.$$

Convergence of the primal-dual gap to zero is shown to be of order $\mathcal{O}(1/n^2)$.

Actually, the theorem proves $\mathcal{O}(1/n)$-convergence of the solution, which implies $\mathcal{O}(1/n^2)$-convergence of the primal-dual gap to zero.

# Comparison (ROF model, according to paper)

Iterations and time to reach accuracy of $\epsilon$
Chambolle and Pock, 2010

| | $\lambda = 1/16$ | | $\lambda = 1/8$ | |
|---|---|---|---|---|
| | $\varepsilon = 10^{-4}$ | $\varepsilon = 10^{-6}$ | $\varepsilon = 10^{-4}$ | $\varepsilon = 10^{-6}$ |
| ALG1 | 214 (3.38s) | 19544 (318.35s) | 309 (5.20s) | 24505 (392.73s) |
| ALG2 | 108 (1.95s) | 937 (14.55s) | 174 (2.76s) | 1479 (23.74s) |
| AHZC | 65 (0.98s) | 634 (9.19s) | 105 (1.65s) | 1001 (14.48s) |
| FISTA | 107 (2.11s) | 999 (20.36s) | 173 (3.84s) | 1540 (29.48s) |
| NEST | 106 (3.32s) | 1213 (38.23s) | 174 (5.54s) | 1963 (58.28s) |
| ADMM | 284 (4.91s) | 25584 (421.75s) | 414 (7.31s) | 33917 (547.35s) |
| PGD | 620 (9.14s) | 58804 (919.64s) | 1621 (23.25s) | – |
| CFP | 1396 (20.65s) | – | 3658 (54.52s) | – |

- Arrow Hurwicz method performs best but can not be shown to converge within $O(1/N^2)$.
- Algorithm 2 performs slightly worse but still better than established $O(1/N^2)$ methods such as FISTA and Nesterov.
- For non-smooth data terms, Algorithm 1 seems to outperform known methods.

# Overview

# Summary

- Saddle point problems are combined minimization/maximization problems of convex-concave functionals.
- The minimization problems in computer vision which are of the form that interest us can be rewritten as saddle point problems - in particular minimization problems with the total variation as a regularizer.
- If the dual energy is available and can easily be computed, the primal-dual gap gives information about how good an estimate for a solution is.
- We have seen how the proximation operator can be computed for the saddle point formulation of standard computer vision applications.
- We have discussed the theorems about optimal choice of step sizes in Algorithm 1, and that it has $\mathcal{O}(1/n)$-convergence.
- We have seen how Algorithm 1 can be accelerated in the case of uniform convex data terms to yield $\mathcal{O}(1/n^2)$-convergence.

# Open source implementation

COCOLIB - Convex Continuous Optimization Library

- Available on SourceForge (GPL3)
- C++ / CUDA, command line tool for image processing
- Currently implemented TV and VTV deblurring / denoising / inpainting / segmentation
- Various algorithms, including FISTA and algorithm 1 and 2 from this lecture
- More coming soon, e.g. total curvature and multilabel methods