Total Variation Regularization and Related Topics

GV08 Optimization and Inverse Problems in Imaging

Felix Lucka

University College London

f.lucka@ucl.ac.uk

Centre for Medical Image Computing

UCL CENTRE FOR INVERSE PROBLEMS

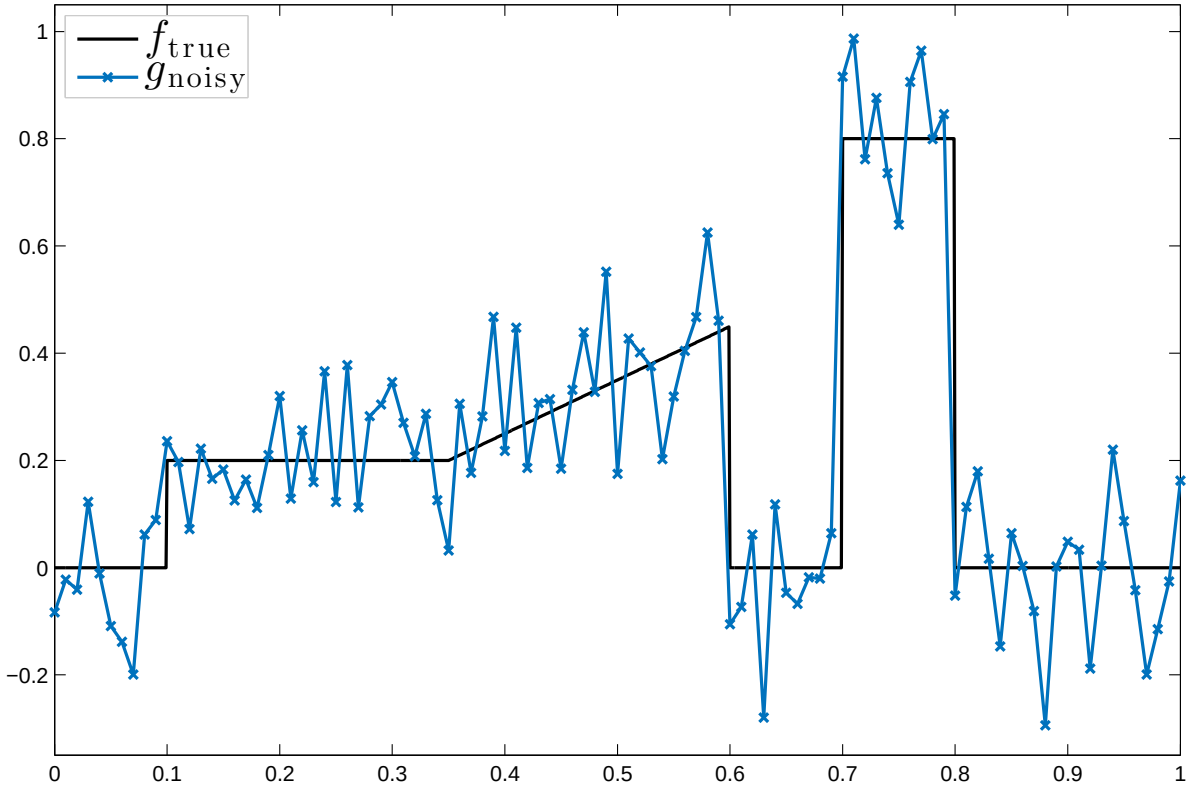
March 2015

Outline

Introduction: Edge-preserving image denoising in 1D

Denoising problem:

$$\tilde{g} = P\, f_{true} + \nu, \qquad \nu \sim \mathcal{N}(0, \sigma^2 I_n)$$



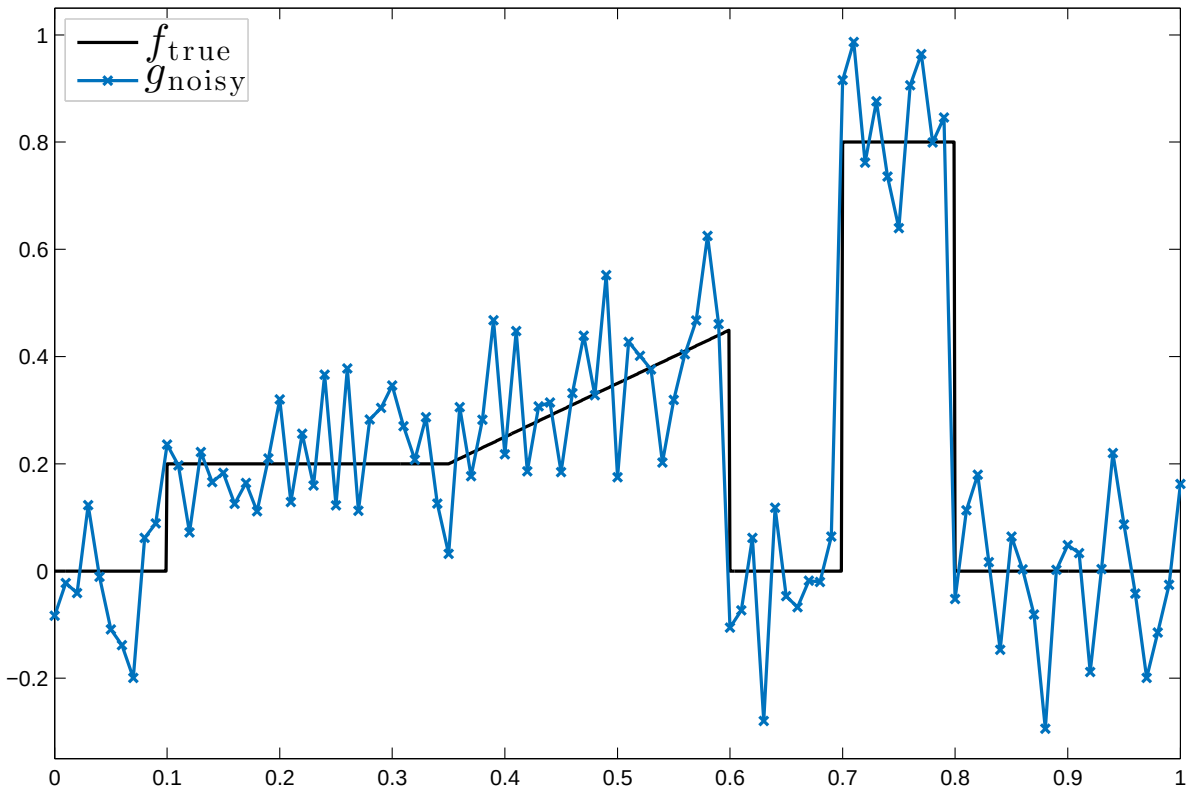
Solution by variational regularization:

$$f_\alpha^\dagger := \underset{f \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \Phi(f) = \|\tilde{g} - f\|_2^2 + \alpha \Psi(f) \right\}$$

Denoising by regularization of the increments

Idea: Remove noise-induced oscillations by regularizing the increments,

$$\Psi(f) = \Psi(Df), \qquad (Df)_i = f_{i+1} - f_i,\ i = 1, \ldots, n-1.$$

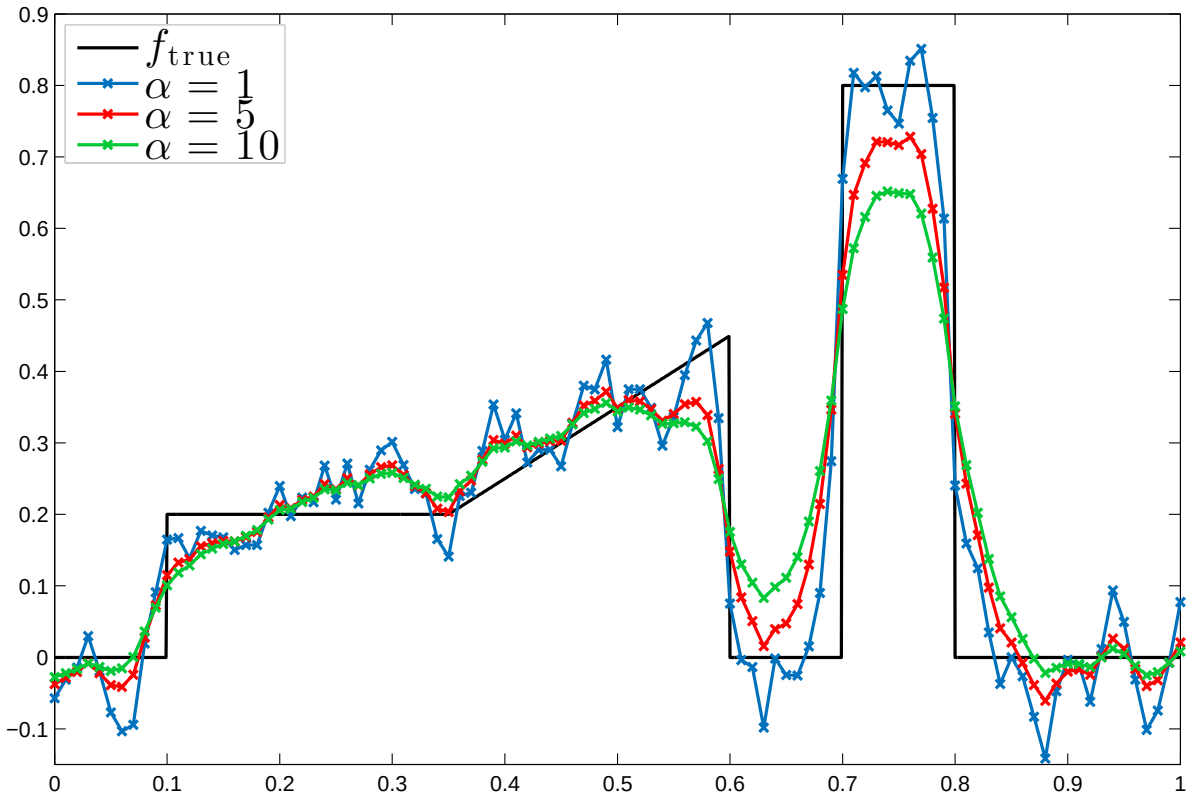


First try, $\Psi(f) = \|Df\|_2^2$.

Denoising by regularization of the increments

Idea: Remove noise-induced oscillations by regularizing the increments,

$$\Psi(f) = \Psi(Df), \qquad (Df)_i = f_{i+1} - f_i,\ i = 1, \ldots, n-1.$$



First try, $\Psi(f) = \|Df\|_2^2 \Longrightarrow$ Result is either noisy or smooth!

Why is the result smooth?

The solution of

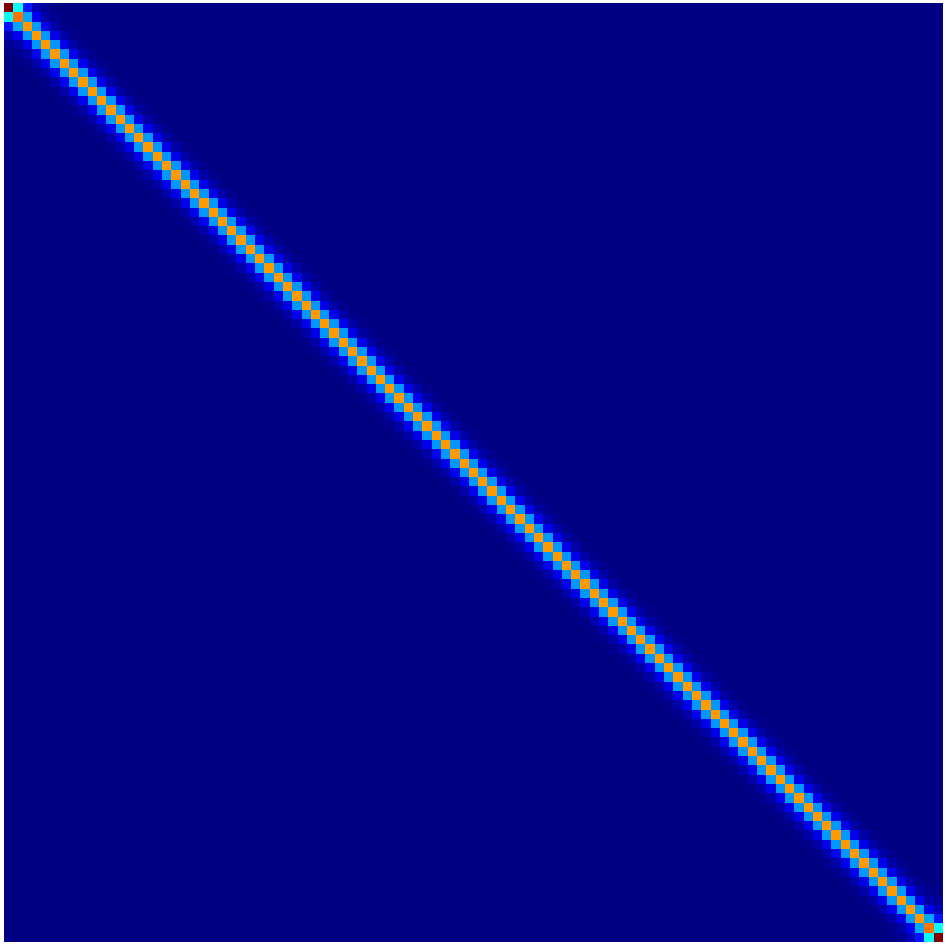
$$f_\alpha^\dagger := \underset{f\in\mathbb{R}^n}{\operatorname{argmin}} \left\{ \|\tilde{g} - f\|_2^2 + \alpha \|Df\|_2^2 \right\} = \underset{f\in\mathbb{R}^n}{\operatorname{argmin}} \left\| \begin{bmatrix} I_n \\ \sqrt{\alpha} D \end{bmatrix} f - \begin{bmatrix} \tilde{g} \\ 0 \end{bmatrix} \right\|_2^2$$

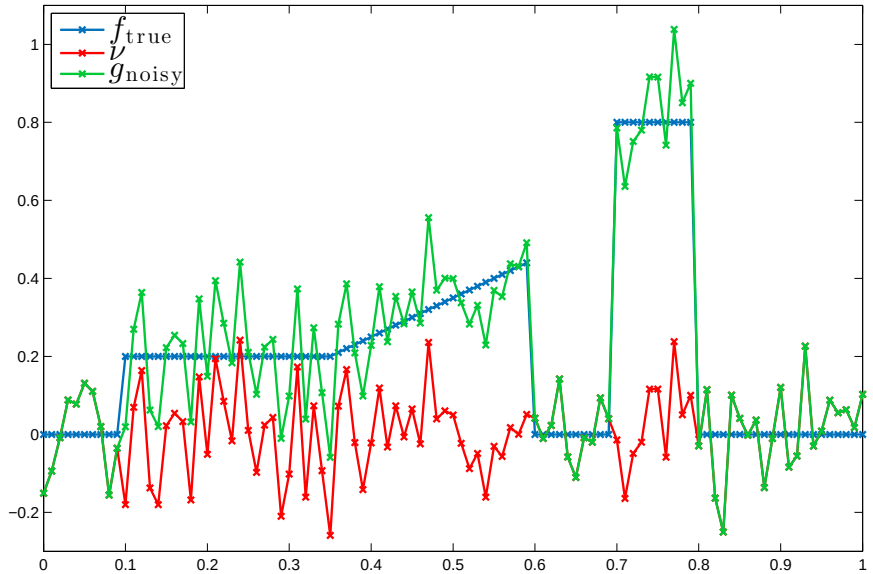
is given by the normal equations

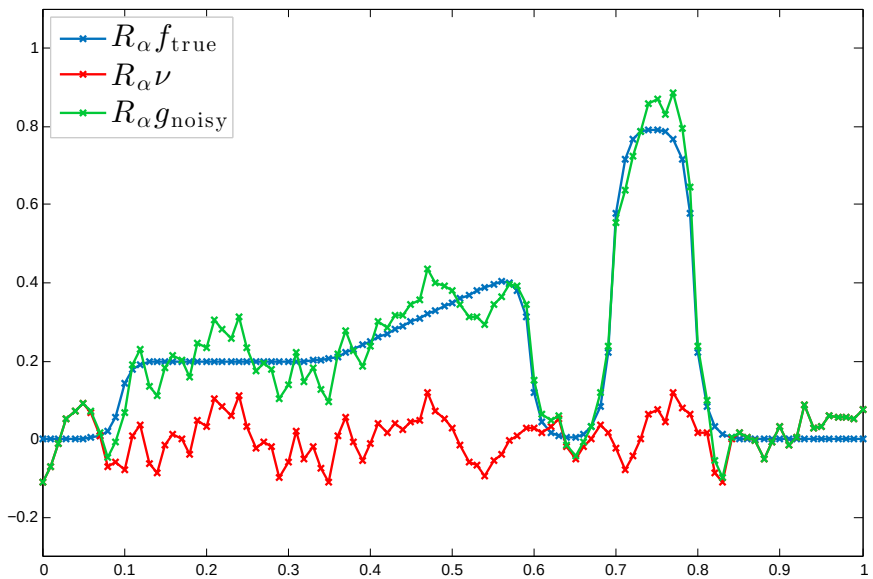
$$\begin{aligned} & \begin{bmatrix} I_n \\ \sqrt{\alpha} D \end{bmatrix}^T \begin{bmatrix} I_n \\ \sqrt{\alpha} D \end{bmatrix} f_\alpha^\dagger = \begin{bmatrix} I_n \\ \sqrt{\alpha} D \end{bmatrix}^T \begin{bmatrix} \tilde{g} \\ 0 \end{bmatrix} \\ \iff \quad & (I_n + \alpha D^T D) f_\alpha^\dagger = \tilde{g} \\ \iff \quad & f_\alpha^\dagger = (I_n + \alpha D^T D)^{-1} \tilde{g} \end{aligned}$$

How does $R_\alpha = (I_n + \alpha D^T D)^{-1}$ look like?

Why is the result smooth?

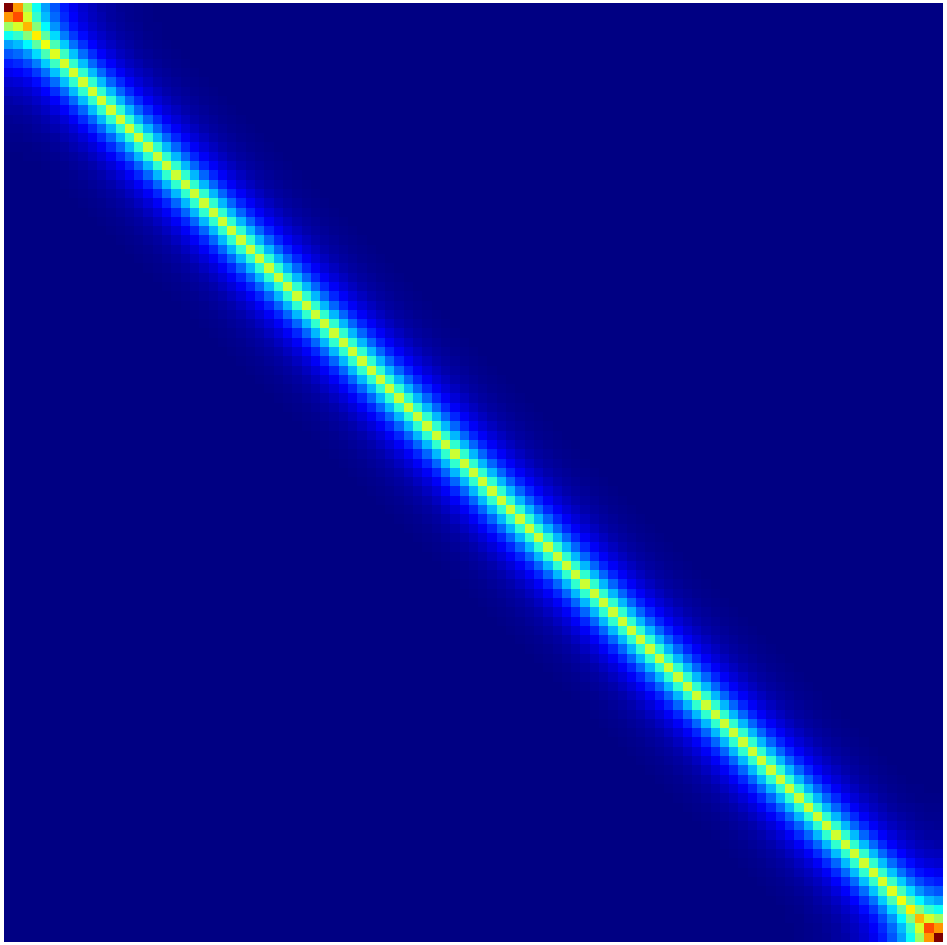


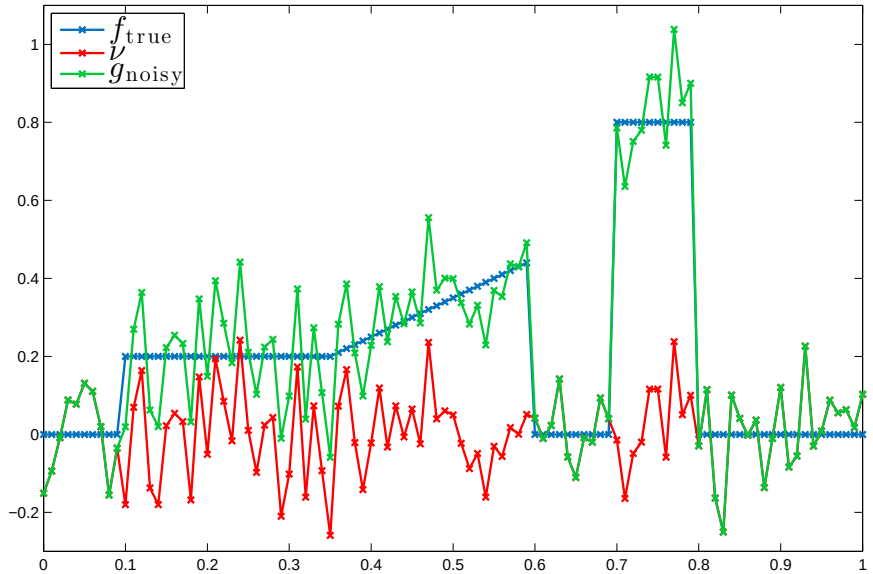


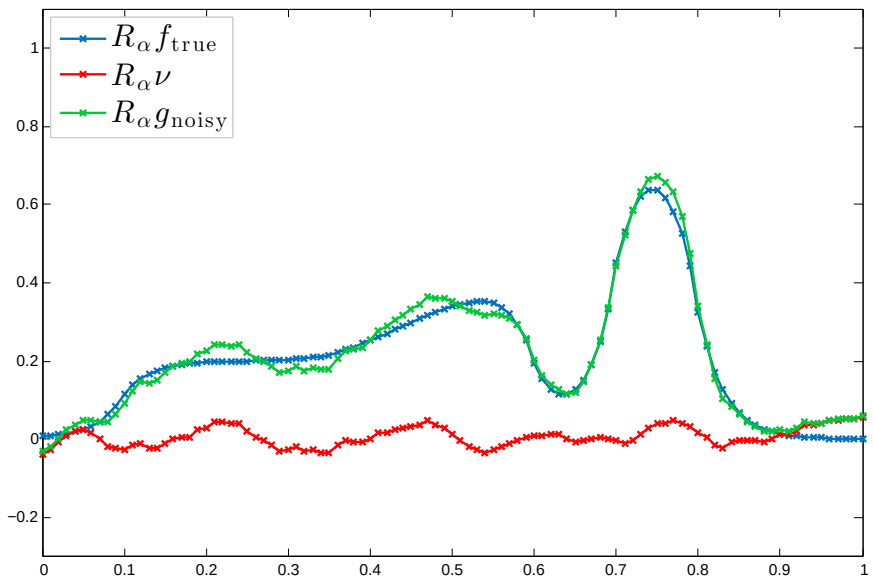


$$R_\alpha = (I_n + \alpha D^T D)^{-1},\ \alpha = 1$$

Why is the result smooth?

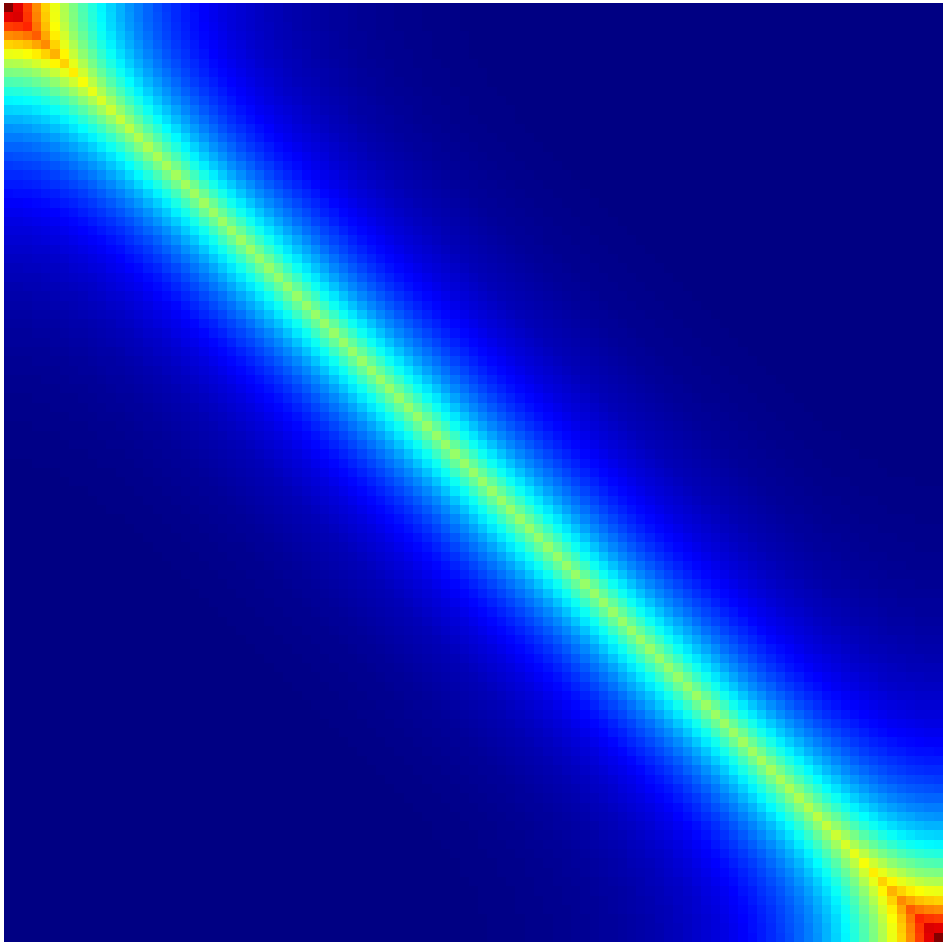


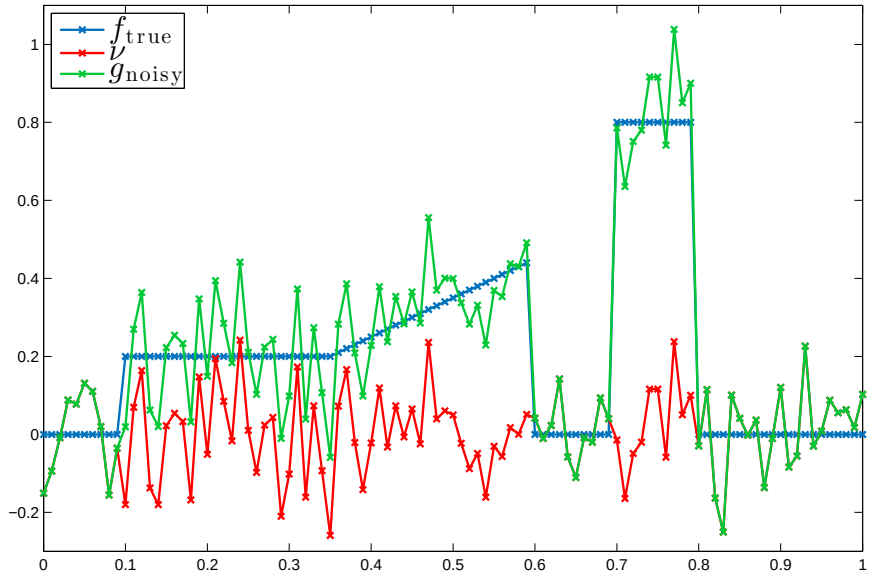


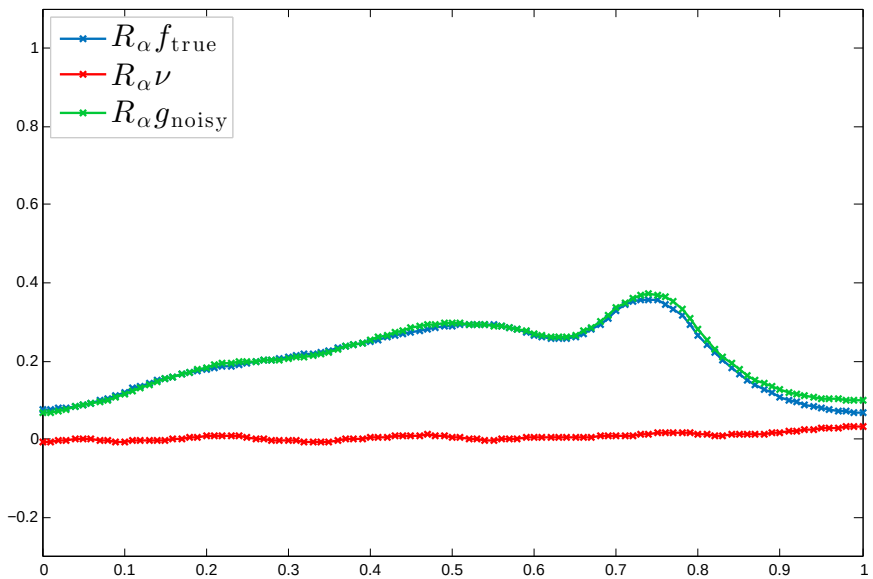


$$R_\alpha = (I_n + \alpha D^T D)^{-1},\ \alpha = 10$$

Why is the result smooth?







$$R_\alpha = (I_n + \alpha D^T D)^{-1},\ \alpha = 100$$

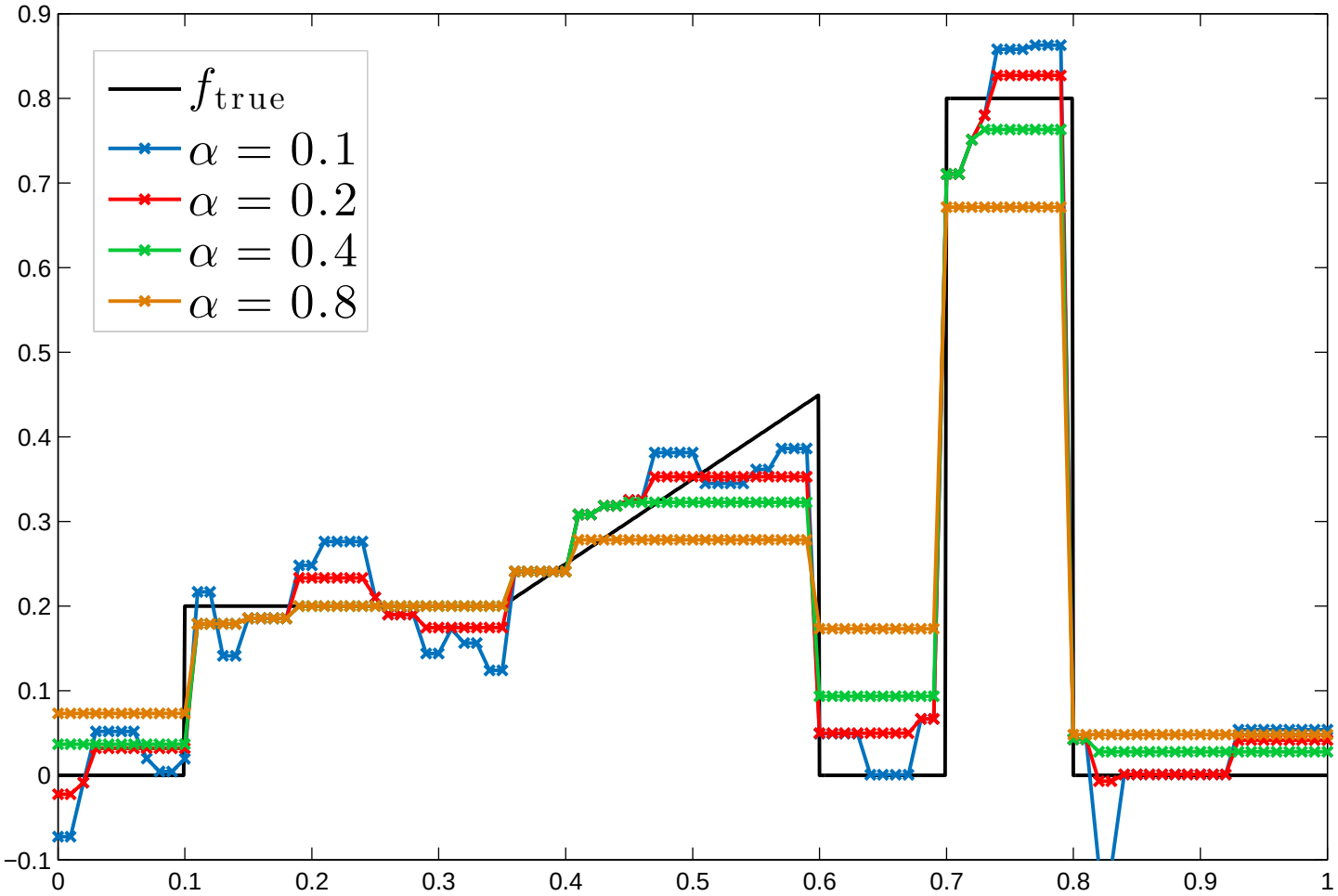
A non-smooth regularizer

For any smooth, convex $\Psi(f)$, the same problem occurs...

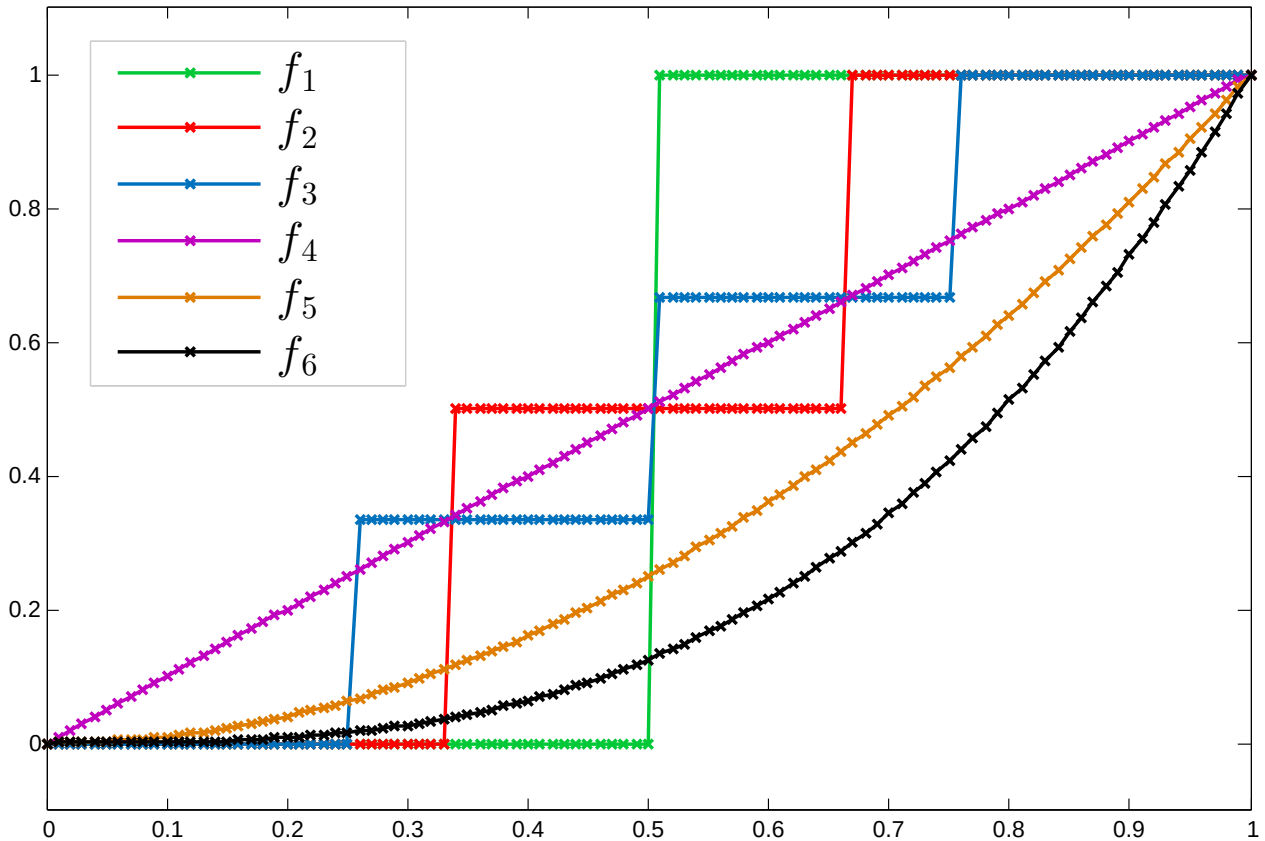
A non-smooth regularizer

For any smooth, convex $\Psi(f)$, the same problem occurs...

Lesson from previous lectures: A non-smooth approach like $\Psi(f) = \|Df\|_1$ might be advantageous as it induces sparsity of the increments (= jumps!).



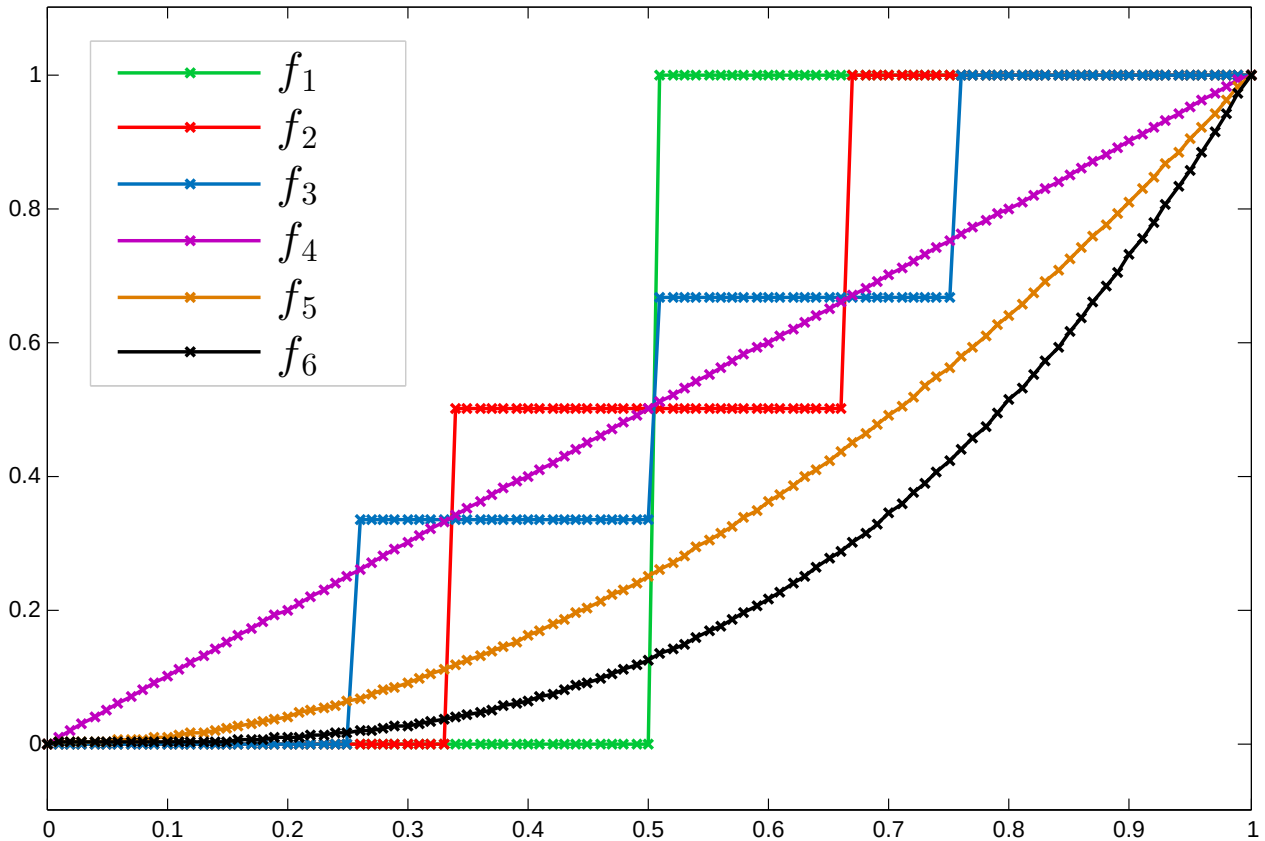
Why is an ℓ_1-norm advantageous?



$$\|Df_1\|_2 = 1.00$$
$$\|Df_2\|_2 = 0.71$$
$$\|Df_3\|_2 = 0.58$$
$$\|Df_4\|_2 = 0.10$$
$$\|Df_5\|_2 = 0.12$$
$$\|Df_6\|_2 = 0.13$$

- $\ell_{p>1}$: Many small jumps are "cheaper" than a large one.
- $\ell_{p=1}$: Splitting a large into smaller steps is not advantageous.
- $\ell_{p<1}$: A large jump is "cheaper" than many small ones.

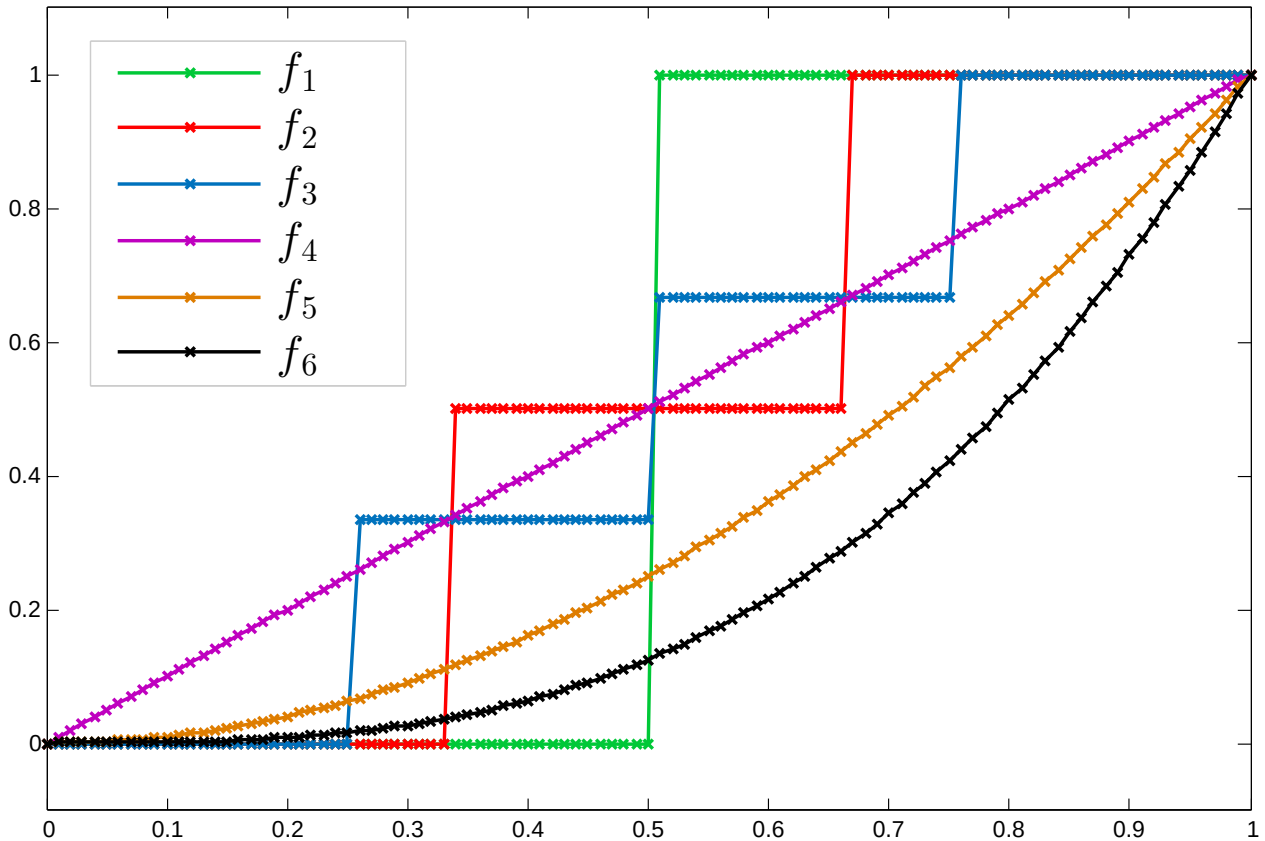
Why is an ℓ_1-norm advantageous?



$$\|Df_1\|_{1.5} = 1.00$$
$$\|Df_2\|_{1.5} = 0.79$$
$$\|Df_3\|_{1.5} = 0.69$$
$$\|Df_4\|_{1.5} = 0.22$$
$$\|Df_5\|_{1.5} = 0.23$$
$$\|Df_6\|_{1.5} = 0.26$$

- $\ell_{p>1}$: Many small jumps are "cheaper" than a large one.
- $\ell_{p=1}$: Splitting a large into smaller steps is not advantageous.
- $\ell_{p<1}$: A large jump is "cheaper" than many small ones.

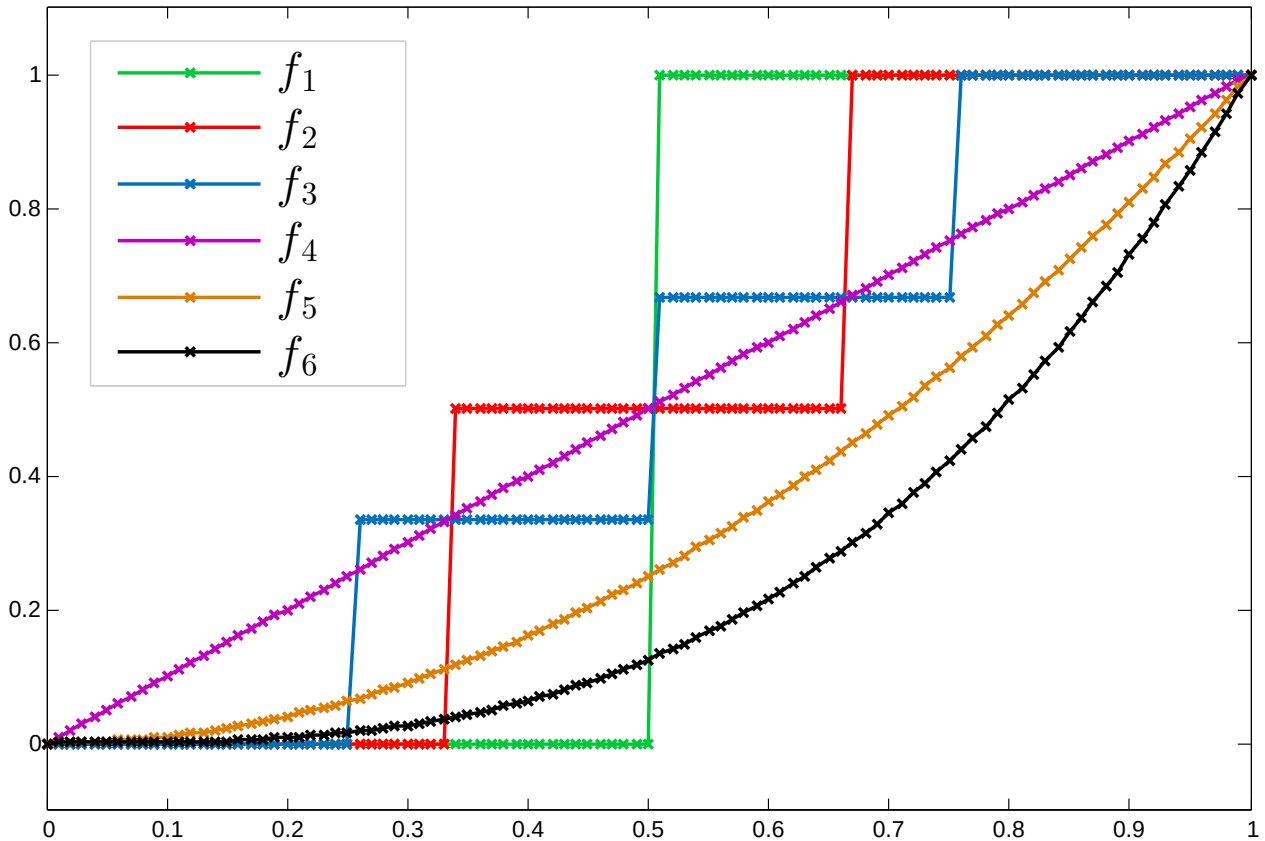
Why is an ℓ_1-norm advantageous?



$$\|Df_1\|_{1.1} = 1.00$$
$$\|Df_2\|_{1.1} = 0.94$$
$$\|Df_3\|_{1.1} = 0.91$$
$$\|Df_4\|_{1.1} = 0.66$$
$$\|Df_5\|_{1.1} = 0.67$$
$$\|Df_6\|_{1.1} = 0.68$$

- $\ell_{p>1}$: Many small jumps are "cheaper" than a large one.
- $\ell_{p=1}$: Splitting a large into smaller steps is not advantageous.
- $\ell_{p<1}$: A large jump is "cheaper" than many small ones.

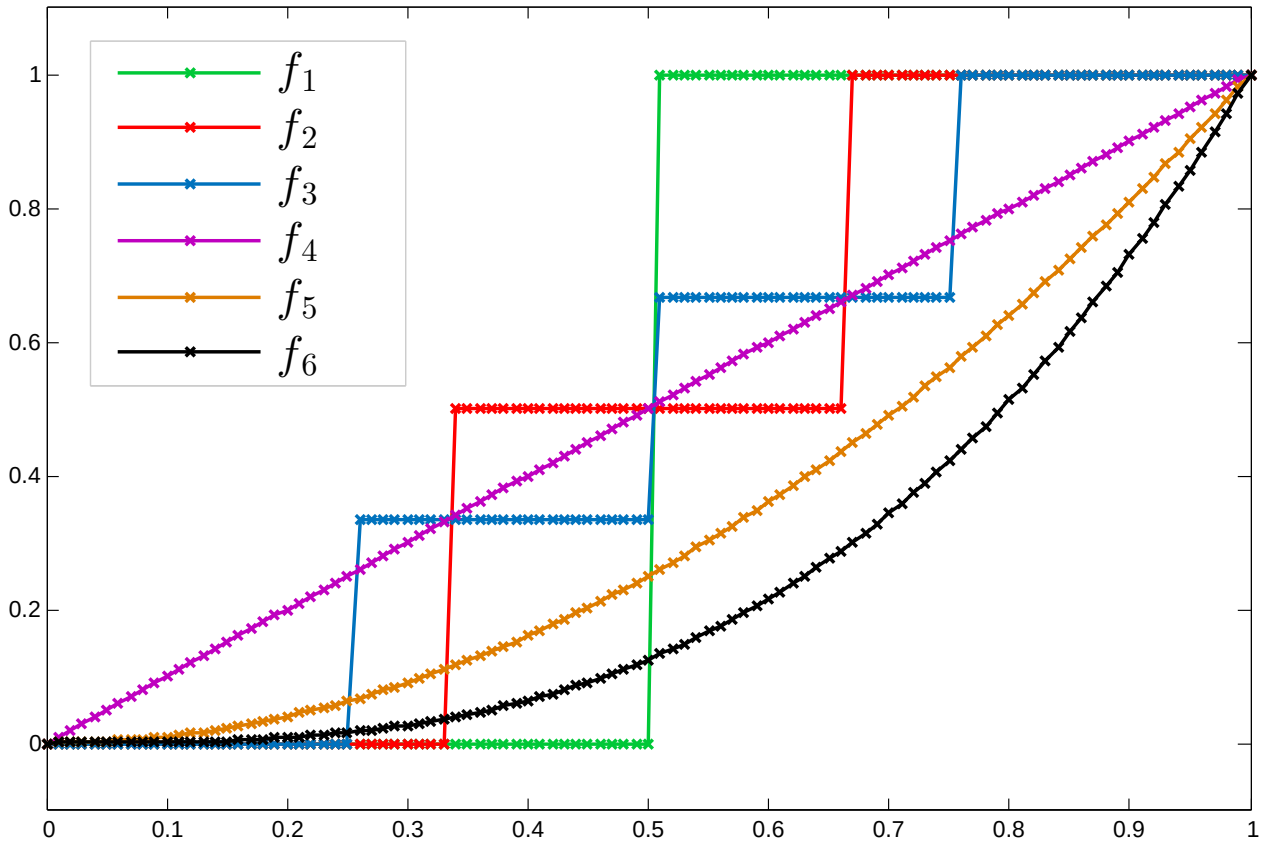
Why is an ℓ_1-norm advantageous?



$$\|Df_1\|_1 = 1.00$$
$$\|Df_2\|_1 = 1.00$$
$$\|Df_3\|_1 = 1.00$$
$$\|Df_4\|_1 = 1.00$$
$$\|Df_5\|_1 = 1.00$$
$$\|Df_6\|_1 = 1.00$$

- $\ell_{p>1}$: Many small jumps are "cheaper" than a large one.
- $\ell_{p=1}$: Splitting a large into smaller steps is not advantageous.
- $\ell_{p<1}$: A large jump is "cheaper" than many small ones.

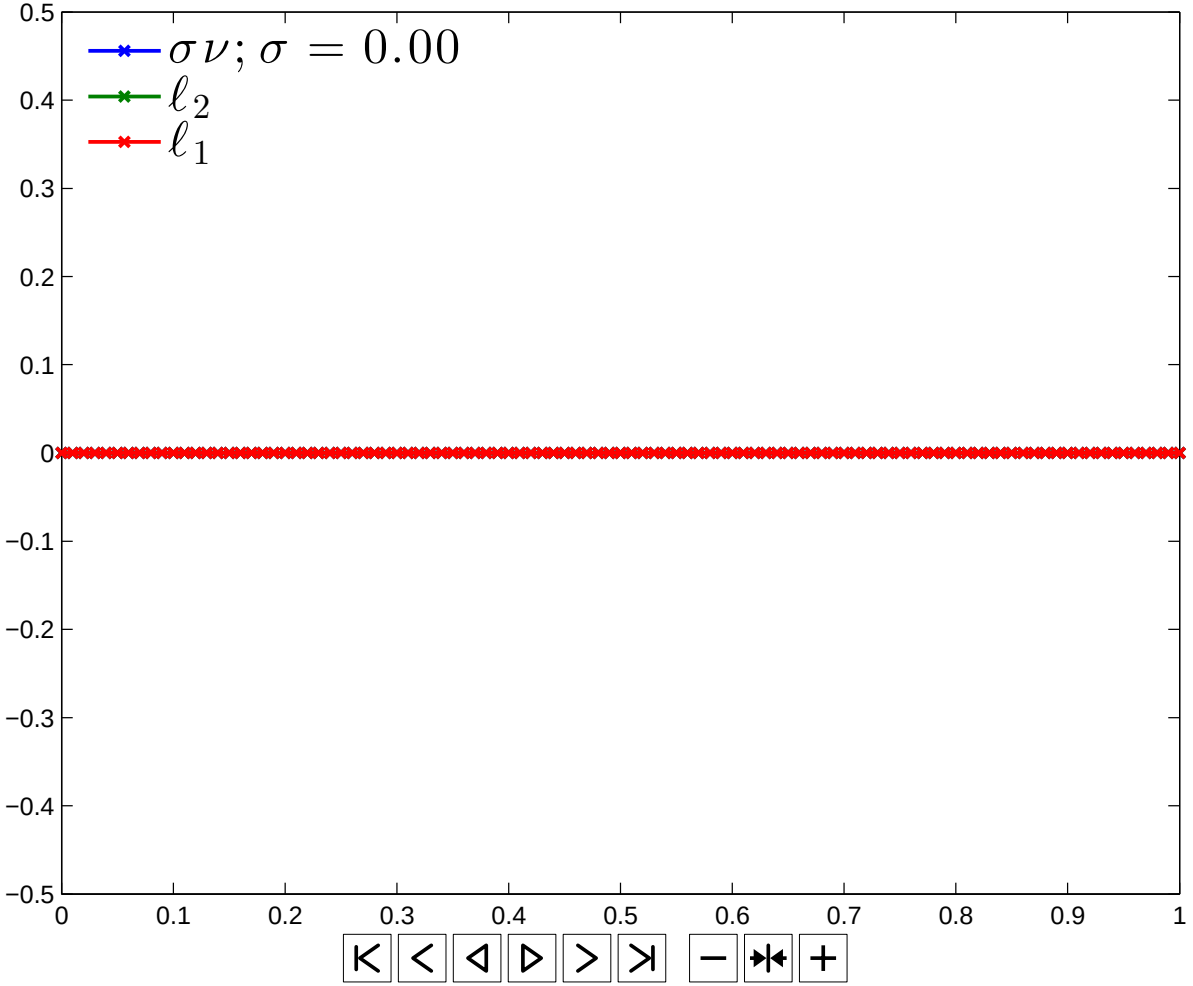
Why is an ℓ_1-norm advantageous?



$$\|Df_1\|_{0.8} = 1.00$$
$$\|Df_2\|_{0.8} = 1.19$$
$$\|Df_3\|_{0.8} = 1.32$$
$$\|Df_4\|_{0.8} = 3.16$$
$$\|Df_5\|_{0.8} = 3.03$$
$$\|Df_6\|_{0.8} = 2.87$$

- $\ell_{p>1}$: Many small jumps are "cheaper" than a large one.
- $\ell_{p=1}$: Splitting a large into smaller steps is not advantageous.
- $\ell_{p<1}$: A large jump is "cheaper" than many small ones.

Why is an ℓ_1-norm advantageous?



For $\tilde{g} = f_{true} + \sigma\nu,\ \nu \sim \mathcal{N}(0,1)$ with increasing σ, we compare

$$\underset{f \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \|\tilde{g} - f\|_2^2 + \|Df\|_2^2 \right\} \quad \textit{vs.} \quad \underset{f \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \|\tilde{g} - f\|_2^2 + \|Df\|_1 \right\}$$

$\Longrightarrow$ ℓ_2 fits noise right away, ℓ_1 only above a threshold.

The total variation of a function in 1D

Our regularization functional

$$\Psi(f) = \|Df\|_1 = \sum_i^{n-1} |f_{i+1} - f_i|$$

can be seen as a discretization of the *total variation* (TV) of a continuous one-dimensional function:

$$TV(f) := \sup \sum_i |f(x_{i+1}) - f(x_i)|,$$

where the supremum is taken over all partitions $0 = x_1 < x_2 < \ldots < x_{n+1} = 1$.

- If f is piecewise constant, $TV(f)$ is the sum of the magnitude of its jumps.
- If f is (weakly) differentiable,

$$TV(f) = \int_0^1 \left\| \frac{df}{dx} \right\| dx$$

Generalization to higher dimensions

A generalization to $\mathbb{R}^d$ is given by

$$TV(f) = \sup_{v \in V} \int_\Omega f \, \nabla \cdot v \, dx,$$
$$V = \{v \in C_0^\infty(\Omega; R^d) \mid \operatorname{ess\,sup}_x \|v(x)\|_2 \leqslant 1\},$$

- For $S \subset\subset \Omega$ with a smooth boundary ∂S we have that $TV(h\mathbb{1}_S)$ is the surface area of S times h.
- For $f \in W^{1,1}(\Omega)$, we have that

$$TV(f) = \int_\Omega \|\nabla f\|_2 \, dx.$$

The (Banach) space of functions of *bounded variation* is defined as

$$BV(\Omega) := \{f \in L^1(\Omega) \mid TV(f) < \infty\}$$

The Rudin-Osher-Fatemi (ROF) model for image denoising

In 1992, Rudin, Osher and Fatemi proposed this denoising technique:

$$TV(f) \to \min_f \quad \text{such that} \quad \int_\Omega (\tilde{g} - f)^2 \, dx \leqslant \varepsilon^2,$$

where ε^2 is a bound on the variance of the noise.

By introducing a Lagrange multiplier, this is equivalent to

$$\min_f \frac{\lambda}{2} \int_\Omega (\tilde{g} - f)^2 \, dx + TV(f),$$

which can be recast to the familiar, discrete denoising model

$$\min_f \|\tilde{g} - f\|_2^2 + \alpha TV(f).$$

An extension to more general inverse problems is given by

$$\min_f \|\tilde{g} - A(f)\|_2^2 + \alpha TV(f).$$

The mathematical analysis of TV regularization is a rich and interesting theory, but needs a lot of preliminaries in functional analysis and convex optimization.

For this course, we stick to the illustrative, hand-waving style and refer to

M. Burger and S. Osher, 2013. *A Guide to the TV Zoo*
in: *Level Set and PDE Based Reconstruction Methods in Imaging*, Lecture Notes in Mathematic. Springer International Publishing.

for more details.

Discrete generalization to higher dimensions

If we use (i, j) to index the pixel in the i^{th} row and j^{th} column of the discrete, $n_x \times n_y$ sized image f, we can define the *isotropic* and *anisotropic TV* of f as:

$$TV_{\text{iso}}(f) = \sum_{(i,j)} \sqrt{\left(f_{(i+1,j)} - f_{(i,j)}\right)^2 + \left(f_{(i,j+1)} - f_{(i,j)}\right)^2}$$

$$TV_{\text{aniso}}(f) = \sum_{(i,j)} |f_{(i+1,j)} - f_{(i,j)}| + |f_{(i,j+1)} - f_{(i,j)}|$$

$$\text{with} \quad f_{(N_x+1,j)} := f_{(N_x,j)}, \; f_{(i,N_y+1)} := f_{(i,N_y)},$$

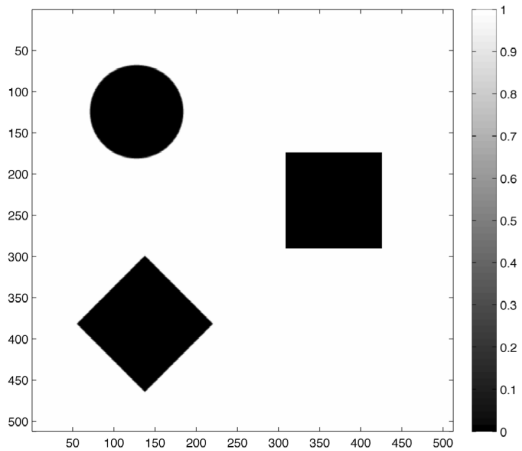
which can be derived from different definitions of the TV functional:

$$TV(f) = \sup_{v \in V} \int_\Omega f \; \nabla \cdot v \, dx,$$

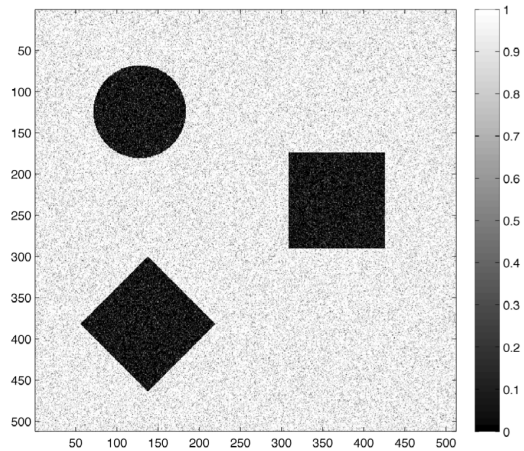
$$V_{\text{iso}} = \{v \in \mathcal{C}_0^\infty(\Omega; R^d) \mid \operatorname*{ess\,sup}_x \|v(x)\|_2 \leqslant 1\}$$

$$V_{\text{aniso}} = \{v \in \mathcal{C}_0^\infty(\Omega; R^d) \mid \operatorname*{ess\,sup}_x \|v(x)\|_1 \leqslant 1\}$$

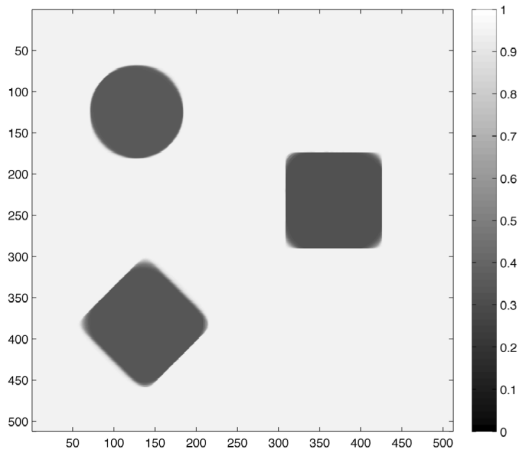
Discrete generalization to higher dimensions



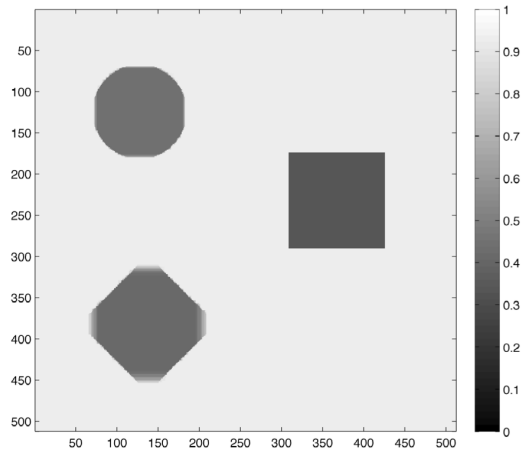
(a) f_{true}



(b) $\tilde{g}$



(c) isotropic TV recon.



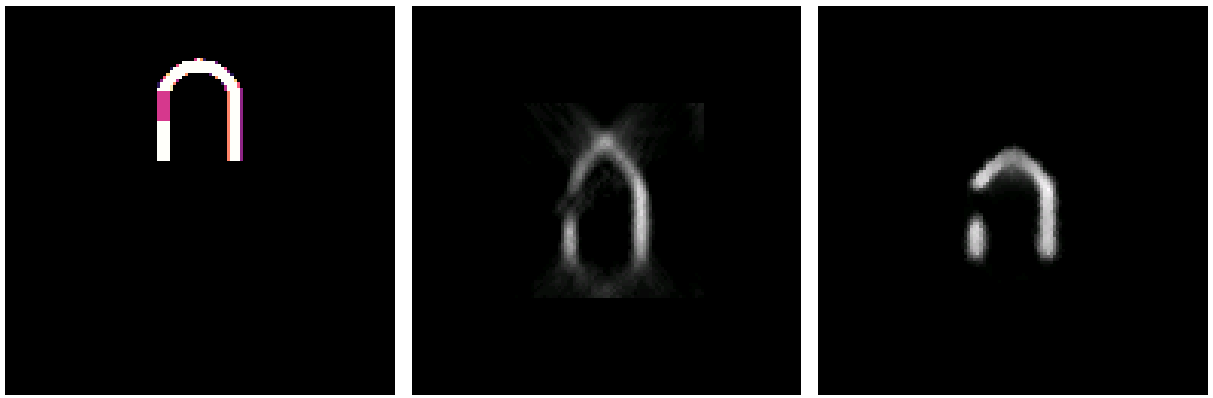
(d) anisotropic TV recon

from: Jahn Müller, 2013. "Advanced Image Reconstruction and Denoising - Bregmanized (Higher Order) Total Variation and Application in PET", *PhD thesis*.

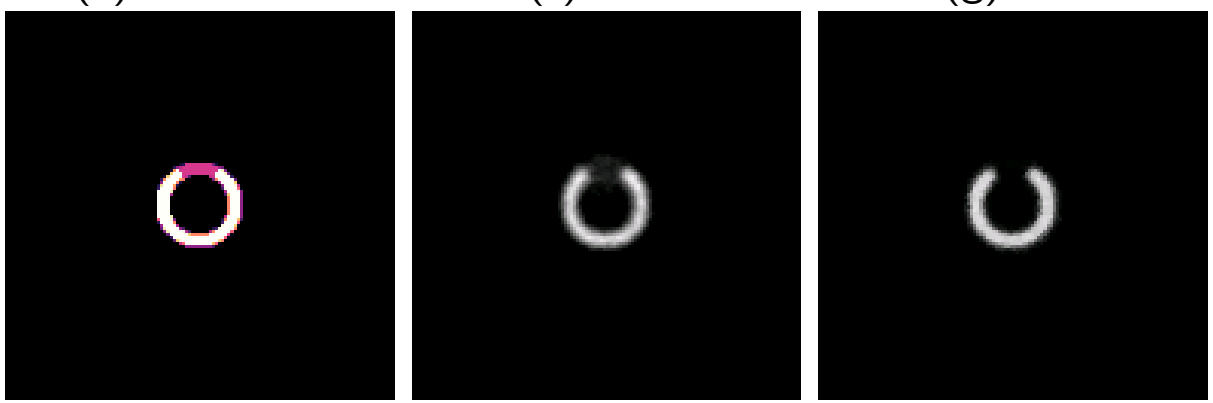
Outline

Limited Angle Computed Tomography (CT)

Traditionally, CT reconstruction is mildly ill-posed. New, limited or sparse angle setups with high noise levels (fast acquisition) change the situation and regularization becomes necessary.



(e) Phantom (f) FBP (g) TV

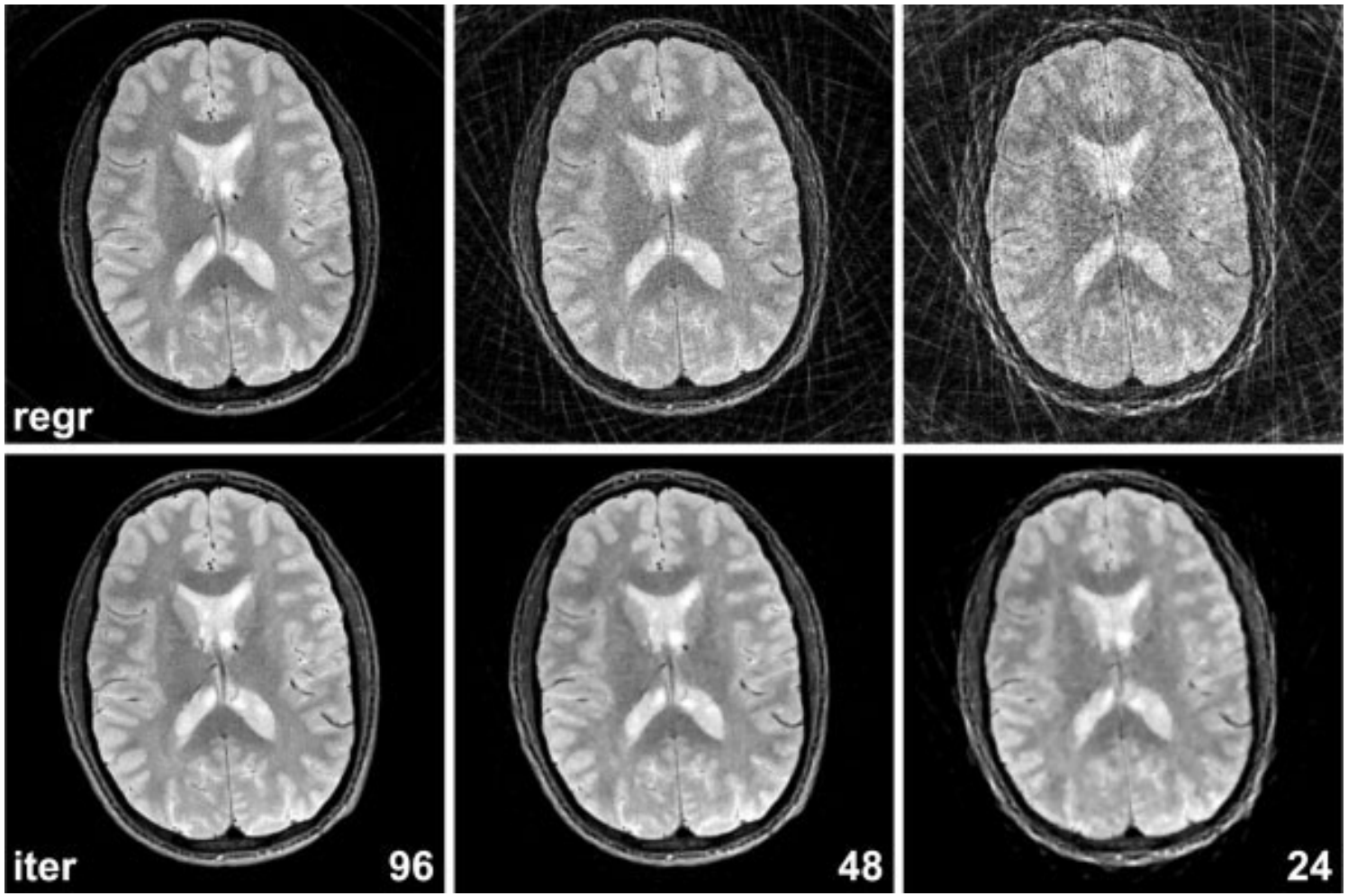


(h) Phantom (i) FBP (j) TV

from: M. Persson, D. Bone and H. Elmqvist, 2001. "Total variation norm for three-dimensional iterative reconstruction in limited view angle tomography", *Phys. Med. Biol.*, 46.

Undersampled magnetic resonance imaging (MRI)

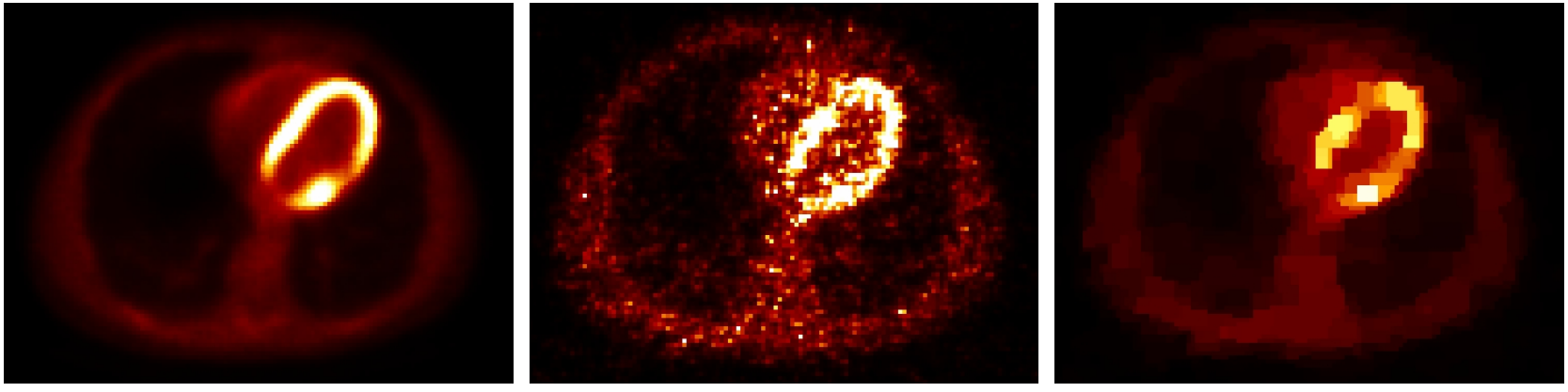
Image reconstruction is mainly achieved by inverting a Fourier transform. New applications heavily undersample in Fourier (k-)space.



from: K.T. Block, M. Uecker, J. Frahm, 2007. "Undersampled Radial MRI with Multiple Coils. Iterative Image Reconstruction Using a Total Variation Constraint", *Magn. Reson. Med.*, 57.

Fast positron emission tomography (PET)

Standart reconstruction techniques require high photon count rates (= long acquisition time / high tracer doses) to produce usable images.

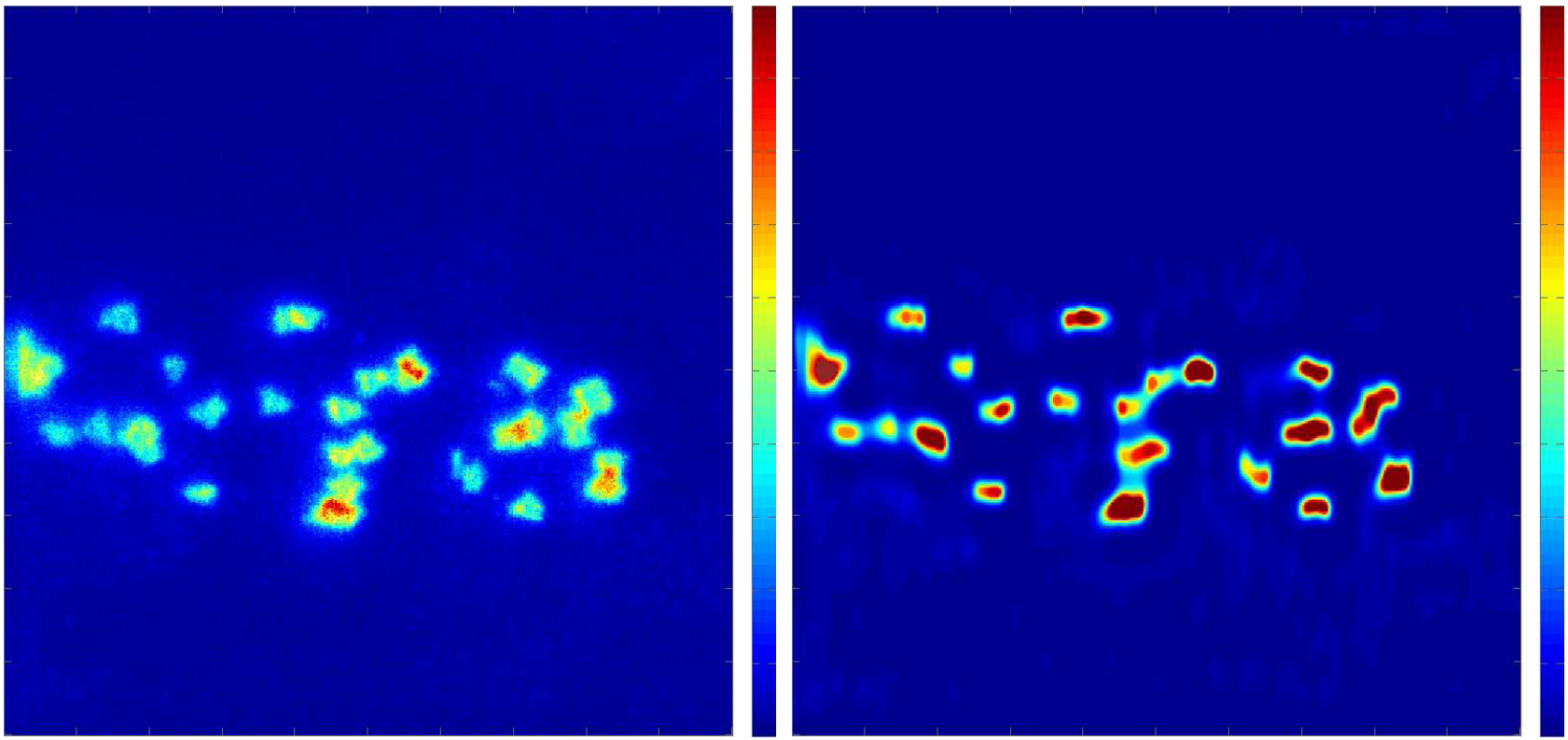


(k) EM, 20 minutes (l) EM, 5 sec (m) EM-TV, 5 sec

from: Jahn Müller, 2013. "Advanced Image Reconstruction and Denoising - Bregmanized (Higher Order) Total Variation and Application in PET", *PhD thesis*.

Optical nanoscopy

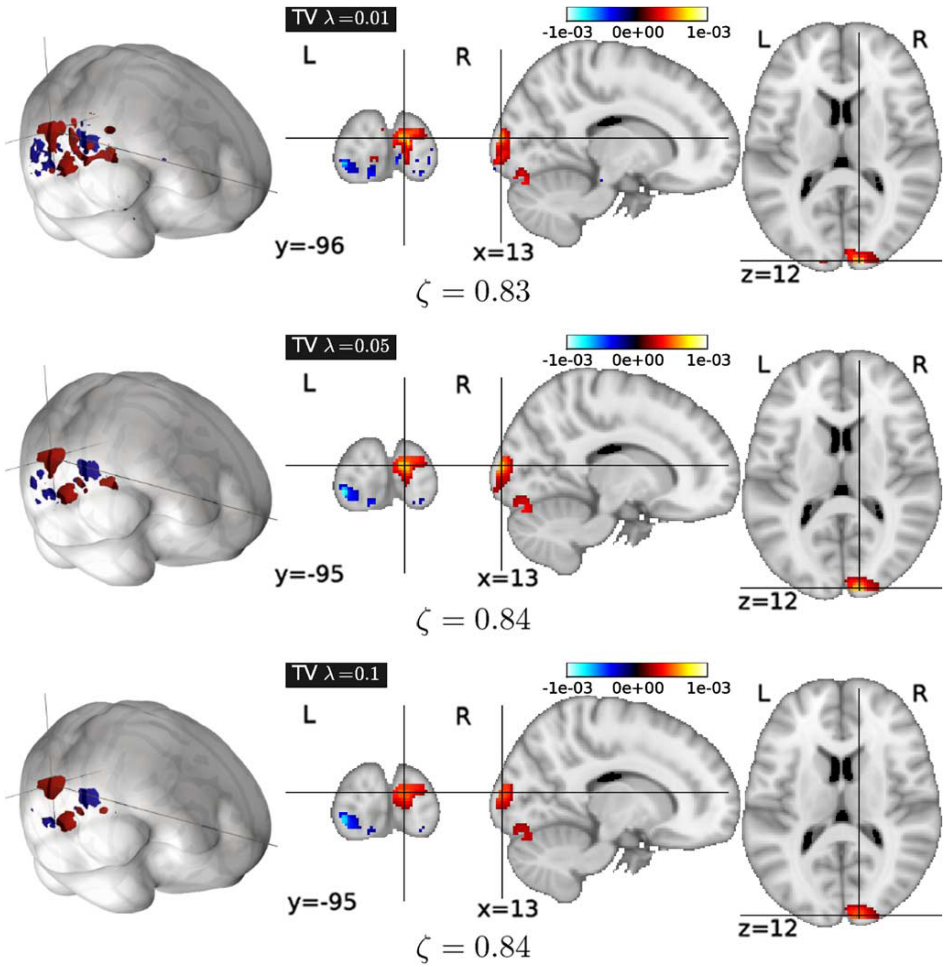
Modern microscopy offering live imaging at nanoscopic scales suffers from low photon counts.



(a) Data (b) EM-TV

Protein Bruchpilot in active zones of neuromuscular synapses in larval Drosophila.

From: C. Brune, A. Sawatzky M. Burger, 2011. "Primal and Dual Bregman Methods with Application to Optical Nanoscopy", *Int. J. Comput. Vis.*, 92.



TV regularization for extracting information form brain images, both for regression and classification tasks.

From V. Michel, A. Gramfort, G. Varoquaux, E. Eger, B. Thirion, 2011. "Total Variation Regularization for fMRI-Based Prediction of Behavior", *IEEE Trans Med Imag*, 30(7).

Reconstruction of ancient frescoes

From M. Fornasier, G. Teschke, R. Ramlau, 2009. "A comparison of joint sparsity and total variation minimization algorithms in a real-life art restoration problem", *Adv. Comput. Math.* 31

Applications: Summary

- TV regularization is particularly successful for reconstructing boundaries of piecewise constant objects from limited, high-noise data.
- Links to compressed sensing (cf. Bangti Jin's lecture)
- Applications beyond biomedical imaging include astronomy, hyperspectral imaging in geoscience, tracking of sharp fronts in weather forecasts, the reconstruction of ancient frescoes and many more...see [Burger and Osher, 2013].
- TV denoising is used as a post-processing step in a lot of applications (not shown here).

Outline

Minimize convex, but non-smooth functional

$$\min_f \quad \|\tilde{g} - Af\|_2^2 + \alpha\|\nabla f\|_1.$$

- Unfortunately, *iterative soft thresholding* or other *proximal gradient schemes* do not work.
- *Steepest decent* or *(quasi-)Newton-type* schemes applied to smooth approximations like

$$|u| \approx \sqrt{u^2 + \varepsilon^2}, \quad \text{or} \quad |u| \approx \begin{cases} |u| - \frac{\varepsilon}{2}, & \text{if } |u| > \varepsilon \\ \frac{u^2}{2\varepsilon} & \text{otherwise} \end{cases}$$

  small step sizes, dependence on ε.
- Anisotropic TV can be recast into *quadratic programming* problem.
- Convex optimization techniques like *dual projected gradient methods* or *primal-dual schemes*.

A general variable splitting approach

We split an unconstrained but coupled problem like

$$\min_f \quad \|\tilde{g} - Af\|_2^2 + \alpha\|Df\|_1.$$

into a constrained but uncoupled problem

$$\min_{f,v} \quad \|\tilde{g} - Af\|_2^2 + \alpha\|v\|_1, \quad \text{such that} \quad Df = v.$$

by introducing an auxiliary variable v.

More general, split such that

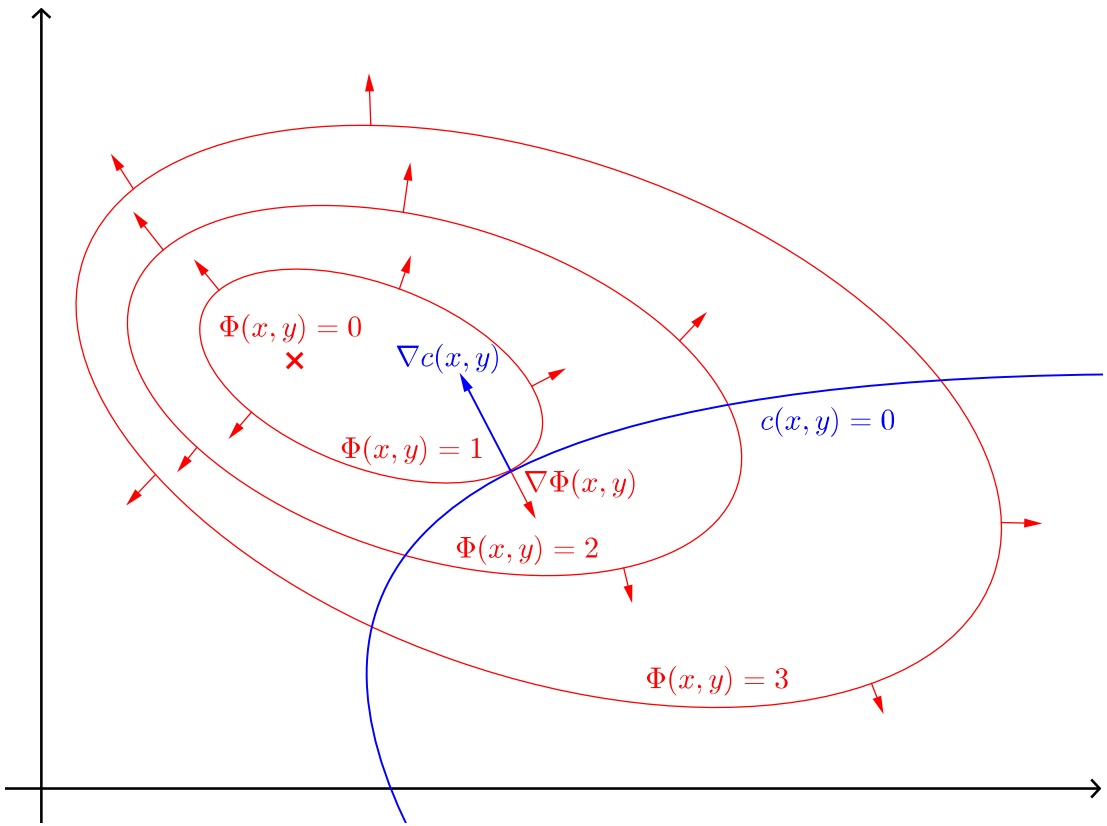
$$\min_f \quad \mathcal{D}(f) + \Psi(f) \;\Leftrightarrow\; \min_{f,v} \quad \tilde{\mathcal{D}}(f) + \tilde{\Psi}(v), \text{ such that } Ef + Hv = c.$$

How do we solve equality-constrained convex optimization problems?

Excursus on equality-constrained convex optimization

Consider an equality-constrained optimization problem in 2D:

$$\begin{aligned} \text{minimize} \quad & \Phi(x, y) \\ \text{subject to} \quad & c(x, y) = 0 \end{aligned}$$

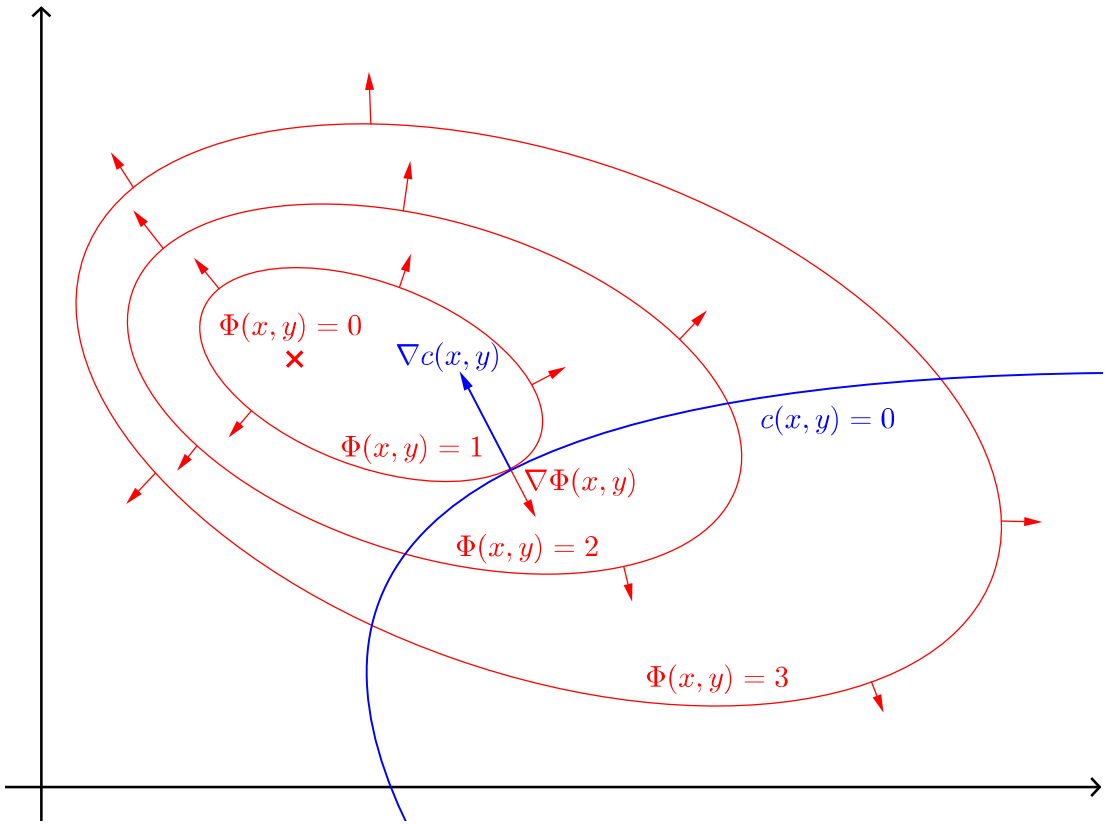


Parameterizing $c(x, y) = 0$ explicitly and solving a 1D problem is not always possible or advantageous.

Walk the line?

Intuition: At an optimal point, $\Phi(x, y)$ cannot be increasing into a direction where $c(x, y) = 0$.

- Walk along $c(x, y) = 0$ until $\Phi(x, y)$ does not change!
- Happens if we walk along a level line of $\Phi(x, y)$ or reach a flat part of $\Phi(x, y)$.



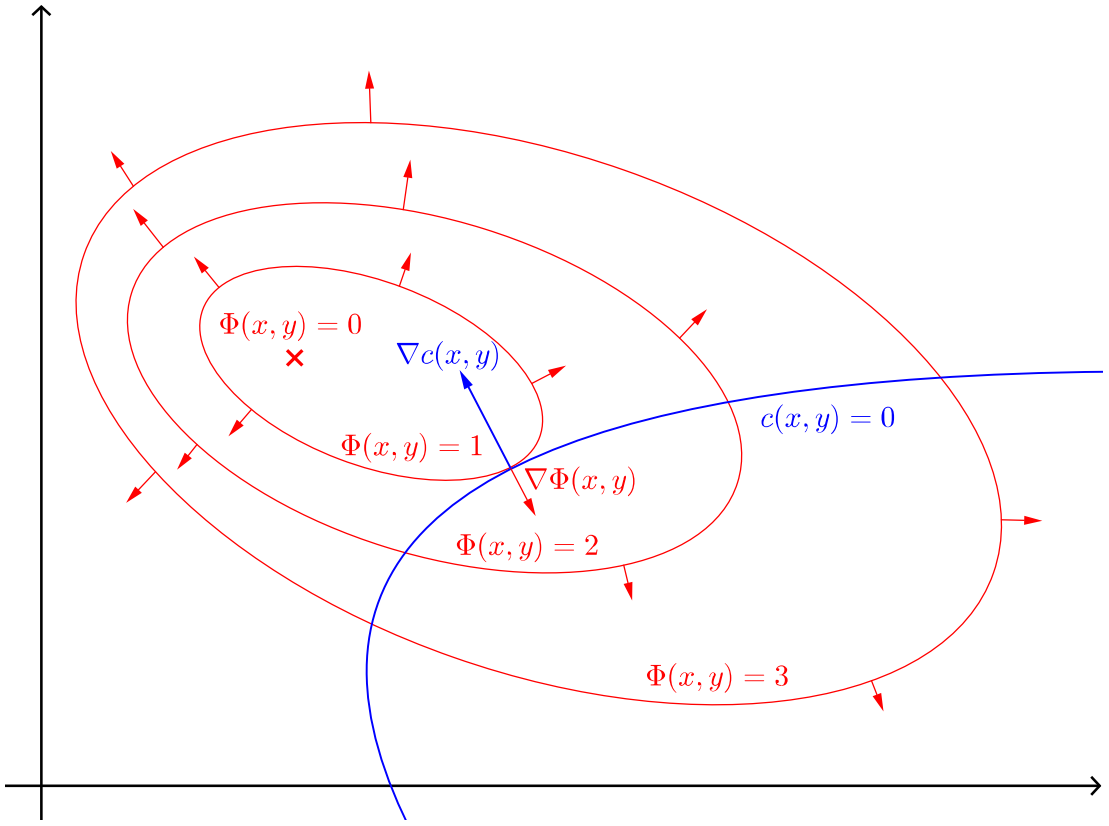
In the first case, the gradient of $\Phi(x, y)$ and $c(x, y)$ are parallel.

Lagrangian multipliers: Where the name comes from

The gradient of $\Phi(x, y)$ and $c(x, y)$ are parallel if

$$\nabla_{x,y}\Phi(x, y) = -\mu\nabla_{x,y}c(x, y)$$

μ is called *Lagrange multiplier*, the "$-$" is convention).
If Φ is flat, then $\nabla_{x,y}\Phi(x, y) = 0$ and $\mu = 0$ fulfills the equation.



The conditions

$$\nabla_{x,y}\Phi(x,y) = -\mu\nabla_{x,y}c(x,y)$$
$$c(x,y) = 0$$

can be combined by introducing the *Lagrange function*

$$\mathcal{L}(x,y,\mu) := \Phi(x,y) + \mu c(x,y)$$

Now,

$$\nabla_{x,y,\mu}\mathcal{L}(x,y,\mu) = 0$$

encodes both conditions!

The constrained extrema of $\Phi(x,y)$ are critical points of $\mathcal{L}(x,y,\mu)$, but they are not local extrema of $\mathcal{L}(x,y,\mu)$!

With m constraints and $x \in \mathbb{R}^n$, the outlined *method of Lagrange multipliers* becomes

$$\mathcal{L}(x, \mu) := \Phi(x) + \sum_i^m \mu_i c_i(x)$$

$$\nabla_x \mathcal{L}(x, \mu) = 0 \;\Leftrightarrow\; \nabla\Phi(x) + \sum_i^m \mu_i c_i(x) = 0$$

$$\nabla_\mu \mathcal{L}(x, \mu) = 0 \;\Leftrightarrow\; c_i(x) = 0,\; i = 1, \ldots, m$$

An extension to inequality constraints leads to the Karush-Kuhn-Tucker (KKT) conditions.

The Lagrange dual function

The *Lagrange dual function* is defined as

$$\mathcal{G}(\mu) := \inf_x \mathcal{L}(x, \mu) = \inf_x \left\{ \Phi(x) + \sum_i^m \mu_i c_i(x) \right\}$$

It is always concave! Now, for the *primal optimization problem*

$$\begin{aligned} \text{minimize} \quad & \Phi(x) \\ \text{subject to} \quad & c(x) = 0 \end{aligned}$$

with optimal value p^*, we can define the *dual optimization problem* as

$$\text{maximize} \quad \mathcal{G}(\mu).$$

We have that $\mathcal{G}(\mu) \leqslant p^*$ for all μ. In particular, $g^* = \sup_\mu \mathcal{G}(\mu)$ is a lower bound for p^*.

How to make practical use of that for convex Φ and linear c?

The dual ascent method

If *strong duality* holds (e.g., if $\Phi(x)$ is strict convex), $g^* = p^*$ and we can recover the optimal x^* from the optimal μ^* by solving:

$$x^* = \underset{x}{\operatorname{argmin}}\, \mathcal{L}(x, \mu^*)$$

Let's assume $c(x) = Ex - b$: $\mathcal{L}(x, \mu) := \Phi(x) + \mu^T(Ex - b)$

The *dual ascent method* solves the dual problem via *gradient ascent*:

- For a given μ, compute $x^+ = \operatorname{argmin}_x \mathcal{L}(x, \mu)$
- Then $\nabla \mathcal{G}(\mu) = Ex^+ - b$, (residual for the equality constraint!)

The dual ascent iteration is given by:

$$x^{k+1} := \underset{x}{\operatorname{argmin}}\, \mathcal{L}(x, \mu^k)$$
$$\mu^{k+1} := \mu^k + \tau^k \left(Ex^{k+1} - b\right)$$

Under strong assumptions and a suitable step size τ^k, we have $\mathcal{G}(\mu^{k+1}) > \mathcal{G}(\mu^k)$ and (x^k, μ^k) converge to optimal points.

Often, dual ascent is not directly applicable or not robust enough!

In *augmented Lagrangian techniques*, $\mathcal{L}(x, \mu)$ is modified to

$$\mathcal{L}_\rho(x, \mu) := \Phi(x) + \mu^T(Ex - b) + (\rho/2)\|Ex - b\|_2^2,$$

which can be viewed as the normal Lagrangian for

$$\begin{aligned} \text{minimize} \quad & \Phi(x) + (\rho/2)\|Ex - b\|_2^2 \\ \text{subject to} \quad & Ex = b \end{aligned}$$

which is equivalent to the original problem!

The dual ascent for the augmented Lagrangian (*method of multipliers*)

$$\begin{aligned} x^{k+1} &:= \operatorname*{argmin}_x \mathcal{L}_p(x, \mu^k) \\ \mu^{k+1} &:= \mu^k + \rho\left(Ex^{k+1} - b\right) \end{aligned}$$

is robust and converges under way more general assumptions.

We have

- split our original problem into a decoupled but constrained problem:

$$\min_{f,v} \quad \tilde{\mathcal{D}}(f) + \tilde{\Psi}(v), \quad \text{such that} \quad Ef + Hv = b.$$

- the method of multipliers to solve constrained problems

Let's bring them together!

Augmented Lagrangian:

$$\mathcal{L}_p(f, v, \mu) = \tilde{\mathcal{D}}(f) + \tilde{\Psi}(v) + \mu^T(Ef + Hv - b) + \rho/2\|Ef + Hv - b\|_2^2$$

Method or multipliers:

$$(f^{k+1}, v^{k+1}) := \operatorname*{argmin}_{(f,v)} \mathcal{L}_p(f, v, \mu^k)$$

$$\mu^{k+1} := \mu^k + \rho\left(Ef^{k+1} + Hv^{k+1} - b\right)$$

The penalty term destroys the decoupling of $\tilde{\mathcal{D}}(f)$ and $\tilde{\Psi}(v)$!

Alternating direction method of multipliers (ADMM)

Replace the joint minimization in the first step

$$(f^{k+1}, v^{k+1}) := \underset{(f,v)}{\operatorname{argmin}}\, \mathcal{L}_p(f, v, \mu^k)$$

by a single (or more) alternation over f and v (Gauss-Seidel type):

$$\begin{aligned} f^{k+1} &:= \underset{f}{\operatorname{argmin}}\, \mathcal{L}_p(f, v^k, \mu^k) \\ v^{k+1} &:= \underset{v}{\operatorname{argmin}}\, \mathcal{L}_p(f^{k+1}, v, \mu^k) \\ \mu^{k+1} &:= \mu^k + \rho\left(Ef^{k+1} + Hv^{k+1} - b\right) \end{aligned}$$

That is the *alternating direction method of multipliers (ADMM)*.

Scaled, explicit form of ADMM

We introduce $w = \mu/\rho$ (*scaled dual variable*), combine the linear and quadratic terms and drop all terms not depending on the variable to minimize to obtain:

$$f^{k+1} := \operatorname*{argmin}_f \left\{ \tilde{\mathcal{D}}(f) + (\rho/2)\|Ef + Hv^k - b + w^k\|_2^2 \right\}$$

$$v^{k+1} := \operatorname*{argmin}_v \left\{ \tilde{\Psi}(v) + (\rho/2)\|Ef^{k+1} + Hv - b + w^k\|_2^2 \right\}$$

$$w^{k+1} := w^k + \left(Ef^{k+1} + Hv^{k+1} - b\right)$$

We define the *primal residual* r^k as

$$r^k = Ef^k + Hv^k - b$$

Thereby:

$$w^k = w^0 + \sum_i r^i$$

Convergence of ADMM

If $\tilde{\mathcal{D}}$ and $\tilde{\Psi}$ are proper, closed and convex and $\mathcal{L}_0(f, v, \mu^k)$ has a saddle point, we have that

- *Residual convergence*: $r^k = Ef^k + Hv^k - b \longrightarrow 0$ as $k \longrightarrow \infty$, i.e., the artificial split is resolved.
- *Objective convergence*: $\tilde{\mathcal{D}}(f^k) + \tilde{\Psi}(v^k) \longrightarrow p^*$, the optimal value of the primal problem.
- *Dual variable convergence*: $\mu \longrightarrow \mu^*$, an optimal dual point.

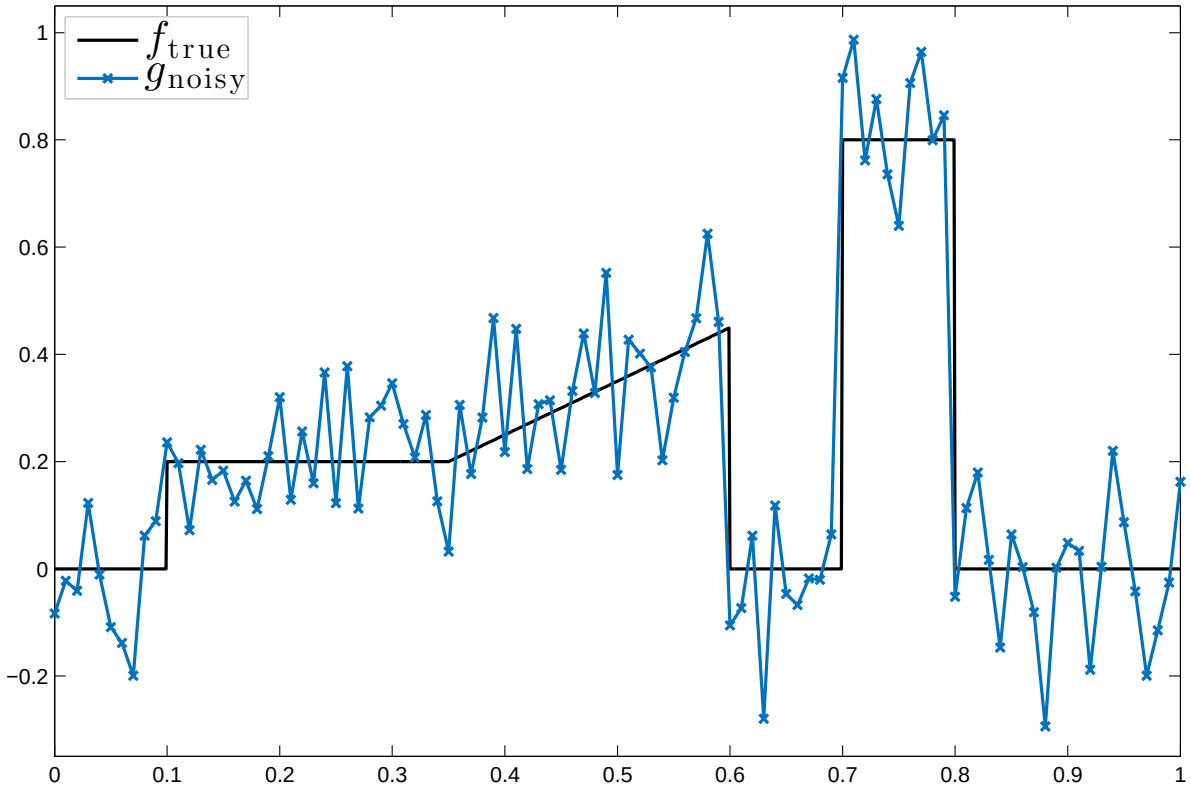
Practically:

- ADMM converges fast (some tens of iterations) to moderate accuracy but can then take long to converge to high accuracy (similar to conjugate gradient methods)
- Different from, e.g. Newton's method, that converges fast in the vicinity of the optimal point.
- Therefore, ADMM is popular in non-smooth, large-scale problems where moderate accuracy is sufficient and Newton's method is not applicable.

ADMM for TV denoising in 1D

Denoising problem:

$$\tilde{g} = P f_{true} + \nu, \qquad \nu \sim \mathcal{N}(0, \sigma^2 I_n)$$



Solution by variational regularization:

$$f_\alpha^\dagger := \underset{f \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \Phi(f) = \tfrac{1}{2}\|\tilde{g} - f\|_2^2 + \alpha \|Df\|_1 \right\}$$

ADMM for TV denoising in 1D

$$\Phi(f) = \tfrac{1}{2}\|\tilde{g} - f\|_2^2 + \alpha\|Df\|_1$$

we split by $v = Df$ which corresponds to $E = D, H = -Id, b = 0$:

$$\begin{aligned}
f^{k+1} &:= \operatorname*{argmin}_f \left\{ \frac{1}{2}\|\tilde{g} - f\|_2^2 + (\rho/2)\|Df - v^k + w^k\|_2^2 \right\} \\
v^{k+1} &:= \operatorname*{argmin}_v \left\{ \alpha\|v\|_1 + (\rho/2)\|Df^{k+1} - v + w^k\|_2^2 \right\} \\
w^{k+1} &:= w^k + \left(Df^{k+1} - v^{k+1}\right)
\end{aligned}$$

- The first step is a first order Tikhonov regularization with an initial guess for the edges $v^k - w^k$. Solution:

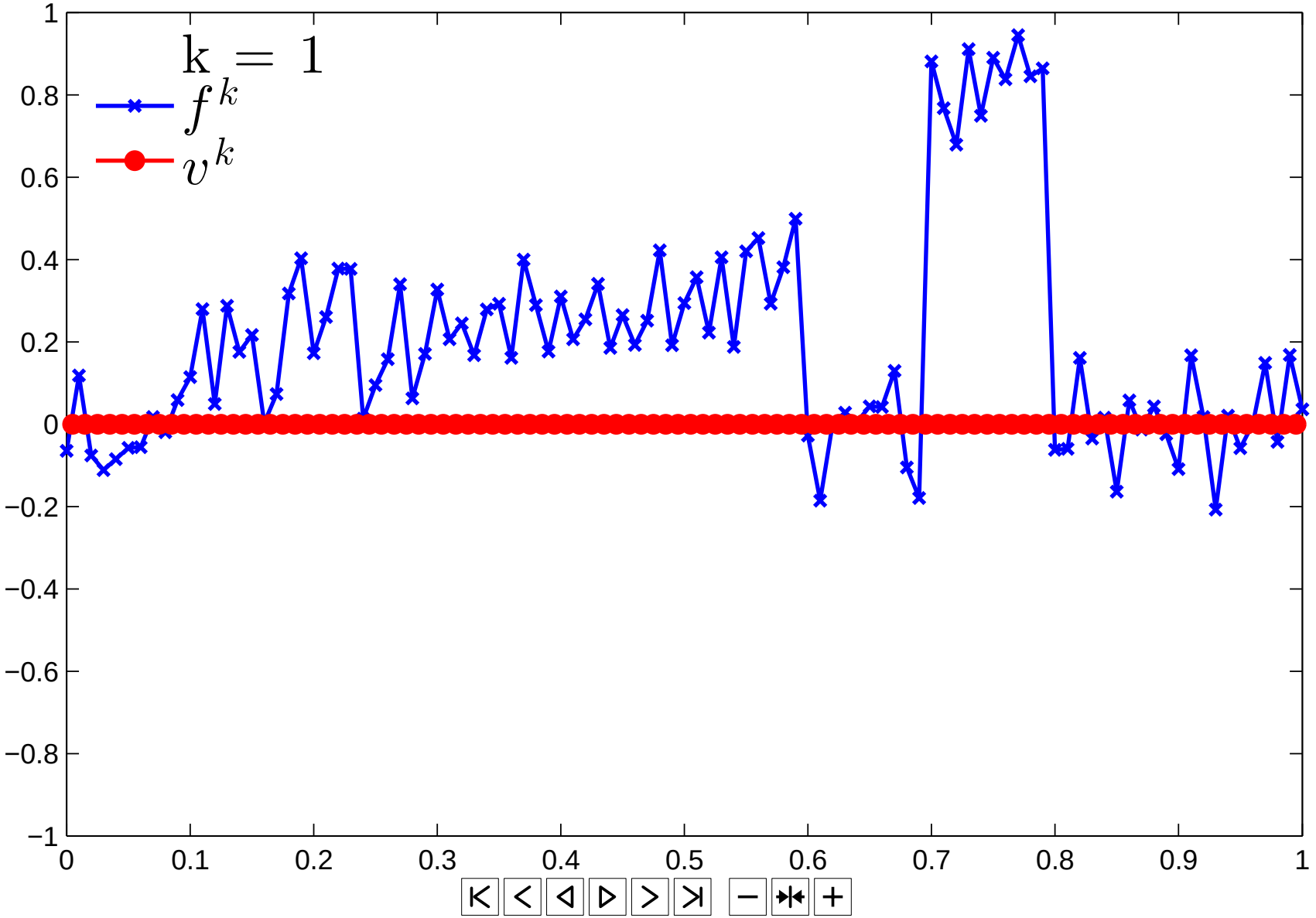
$$f^{k+1} = \left(I_n + \rho D^T D\right)^{-1} \left(\tilde{g} + \rho D^T (v^k - w^k)\right)$$

- The second step decouples into one-dimensional problems:

$$\min_s \ \tfrac{1}{2}(s-t)^2 + \lambda|s|, \quad s = v_i, \ t = \left(Df^{k+1} + w^k\right)_i, \ \lambda = \alpha/\rho$$

The solution is given by the soft thresholding operator $S_\lambda(t)$.
(cf. slide 21 of Bangti Jin's lecture).

ADMM for TV denoising in 1D



ADMM for TV deblurring in 2D

Now we look the deblurring of a $n_x \times n_y$ image convoluted by a Gaussian kernel:

$$\tilde{g} = Af_{true} + \nu, \qquad \nu \sim \mathcal{N}\left(0, \sigma^2 I_{n_x \times n_y}\right)$$

(a) f_{true}

(b) $A\,f_{true}$

(c) $\tilde{g}$

Solution by minimizing $\Phi(f) = \frac{1}{2}\|\tilde{g} - Af\|_2^2 + \alpha TV(f)$, where $TV(f)$ is the isotropic TV functional:

$$TV(f) = \sum_{(i,j)} \sqrt{\left(f_{(i+1,j)} - f_{(i,j)}\right)^2 + \left(f_{(i,j+1)} - f_{(i,j)}\right)^2}$$

$$\text{with} \quad f_{(n_x+1,j)} := f_{(n_x,j)}, \; f_{(i,n_y+1)} := f_{(i,n_y)},$$

ADMM for TV deblurring in 2D

$$\Phi(f) = \tfrac{1}{2}\|\tilde{g} - Af\|_2^2 + \alpha TV(f)$$

We split by

$$v := \begin{bmatrix} v_x \\ v_y \end{bmatrix} = Df := \begin{bmatrix} D_x \\ D_y \end{bmatrix} f,$$

where D_x and D_y are the finite difference operators in x and y direction.

The first step,

$$f^{k+1} := \underset{f}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\tilde{g} - Af\|_2^2 + (\rho/2)\|Df - v^k + w^k\|_2^2 \right\}$$

is again a first order Tikhonov regularization with an initial guess for the edges $v^k - w^k$. Solution:

$$f^{k+1} = \left(A^T A + \rho D_x^T D_x + \rho D_y^T D_y\right)^{-1} h^k$$

$$h^k := \left(A^T \tilde{g} + \rho D_x^T (v_x^k - w_x^k) + \rho D_y^T (v_y^k - w_y^k)\right); \quad w := \begin{bmatrix} w_x \\ w_y \end{bmatrix}$$

An iterative solution of the augmented least-squares problem is preferable!

$$\Phi(f) = \tfrac{1}{2}\|\tilde{g} - Af\|_2^2 + \alpha TV(f), \qquad v := \begin{bmatrix} v_x \\ v_y \end{bmatrix} = Df := \begin{bmatrix} D_x \\ D_y \end{bmatrix} f$$

The split turns $$TV(f) = \sum_{(i,j)} \sqrt{\left(f_{(i+1,j)} - f_{(i,j)}\right)^2 + \left(f_{(i,j+1)} - f_{(i,j)}\right)^2}$$

into $$\sum_{(i,j)} \sqrt{(v_x)_{(i,j)}^2 + (v_y)_{(i,j)}^2} \quad =: \quad \|V\|_{2,1}, \qquad V := [v_x, v_y],$$

i.e., the ℓ_1 norm of the amplitude (ℓ_2-norm) of a vector field v.

The problem

$$v^{k+1} := \underset{v}{\operatorname{argmin}} \left\{ \alpha \|V\|_{2,1} + (\rho/2) \|Df^{k+1} - v + w^k\|_2^2 \right\}$$

decouples into two-dimensional problems:

$$\min_s \ \tfrac{1}{2}(s_x - t_x)^2 + \tfrac{1}{2}(s_x - t_x)^2 + \lambda \sqrt{s_x^2 + s_y^2}, \qquad \text{with} \quad \lambda = \alpha/\rho,$$

$$s_x = (v_x)_{(i,j)}\,, s_y = (v_y)_{(i,j)}\,, t_x = \left(D_x f^{k+1} + w_x^k\right)_{(i,j)}\,, t_y = \left(D_y f^{k+1} + w_y^k\right)_{(i,j)}$$

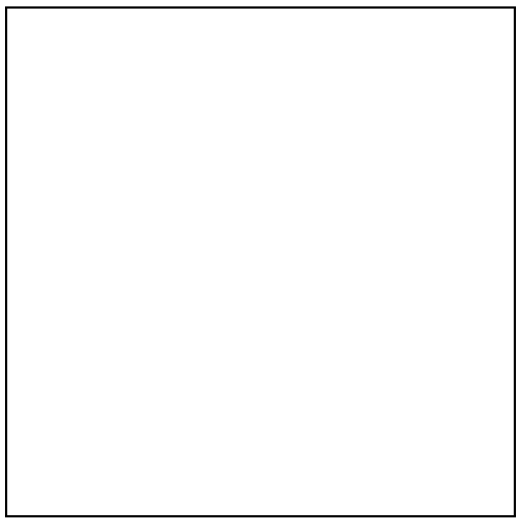
The solution is given by the *vectorial soft thresholding operator* $S_\lambda^{\text{vec}}(t)$:

$$S_\lambda^{\text{vec}}(t) := \begin{cases} \frac{S_\lambda(\|t\|_2)}{\|t\|_2}\ t & \text{if } \|t\|_2 > \lambda \\ 0 & \text{otherwise} \end{cases}$$

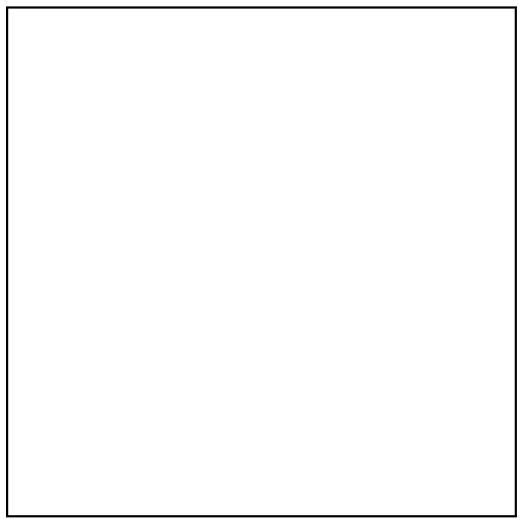
which soft threholds on the amplitude of the vector, only.

Iteration $k = 1$

(d) f^k



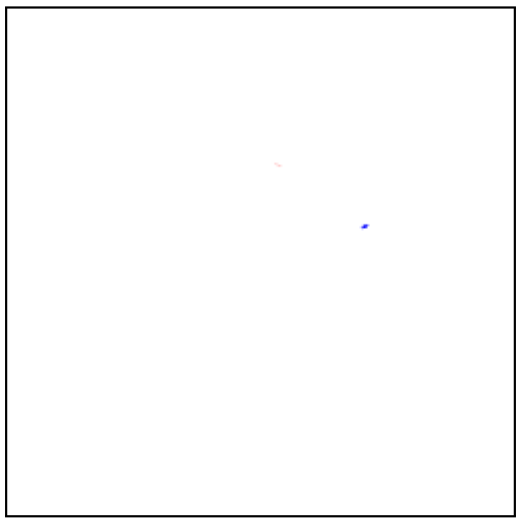
(e) v_x^k



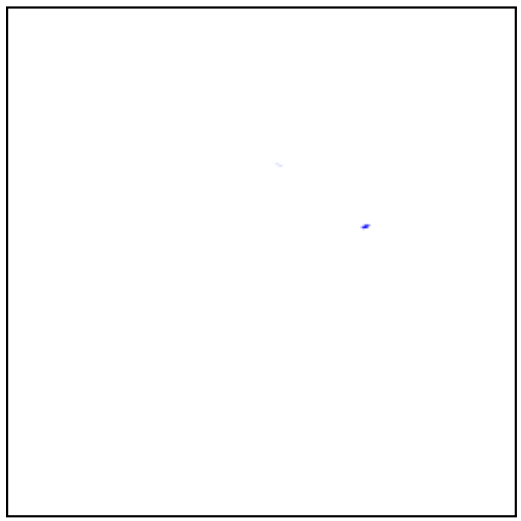
(f) v_y^k

Iteration $k = 2$

(a) f^k



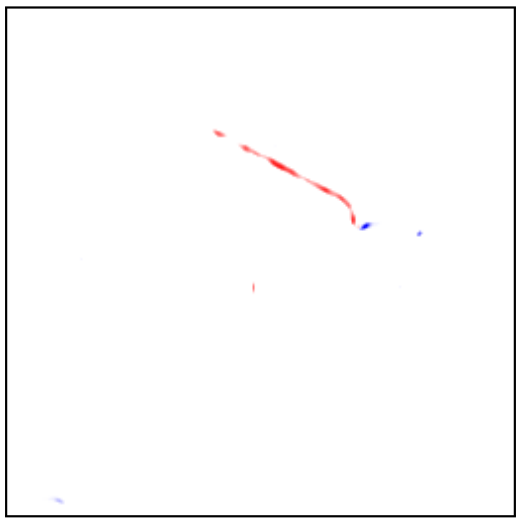
(b) v_x^k



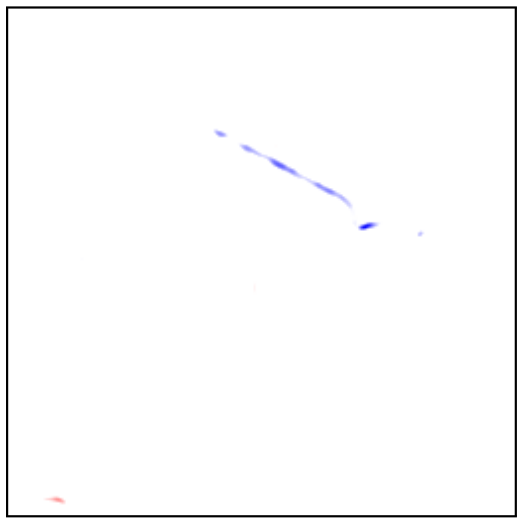
(c) v_y^k

Iteration $k = 3$

(a) f^k



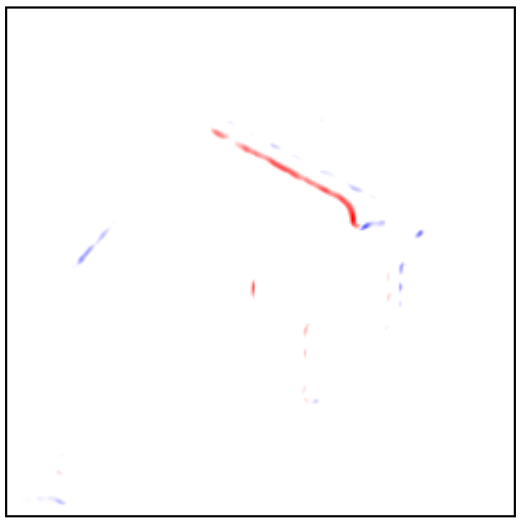
(b) v_x^k



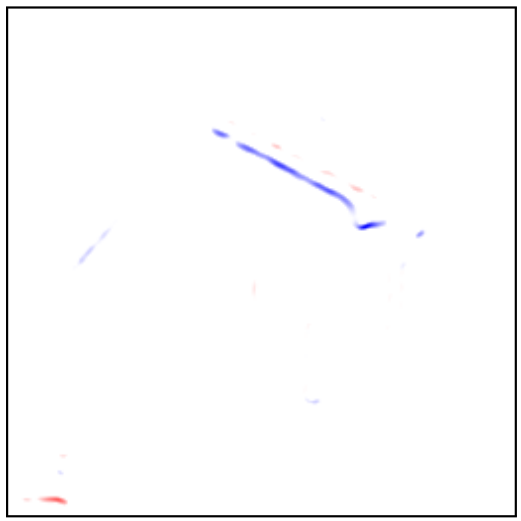
(c) v_y^k

Iteration $k = 4$

(a) f^k



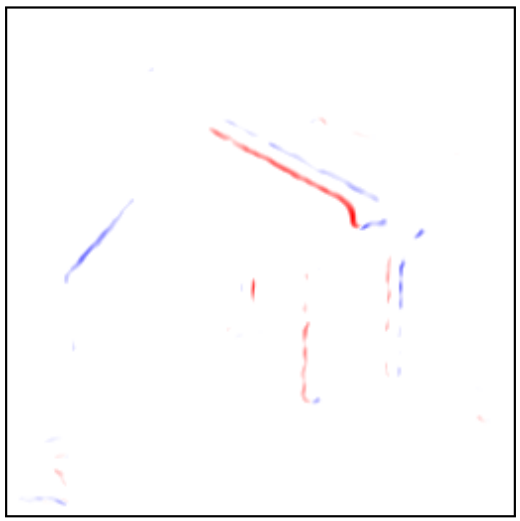
(b) v_x^k



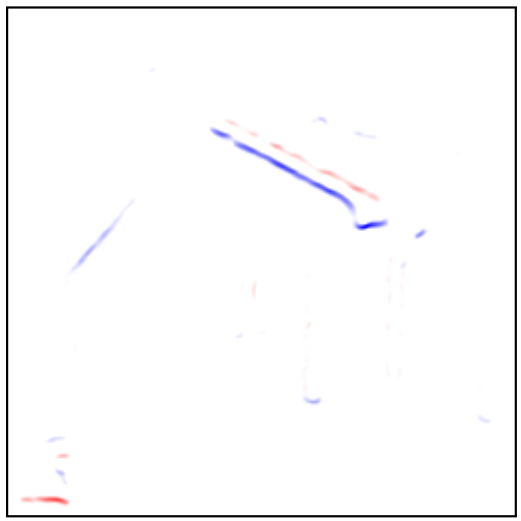
(c) v_y^k

Iteration $k = 5$

(a) f^k



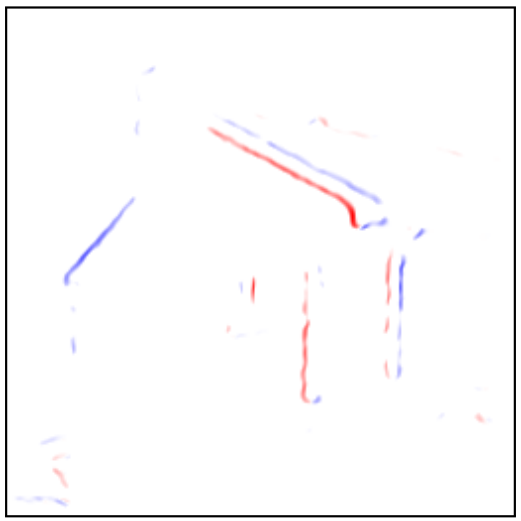
(b) v_x^k



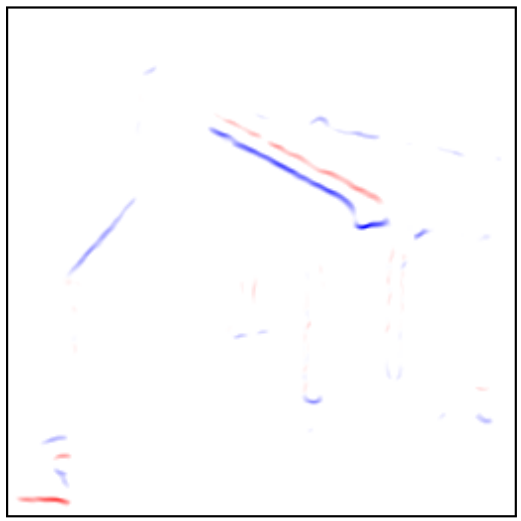
(c) v_y^k

Iteration $k = 6$

(a) f^k



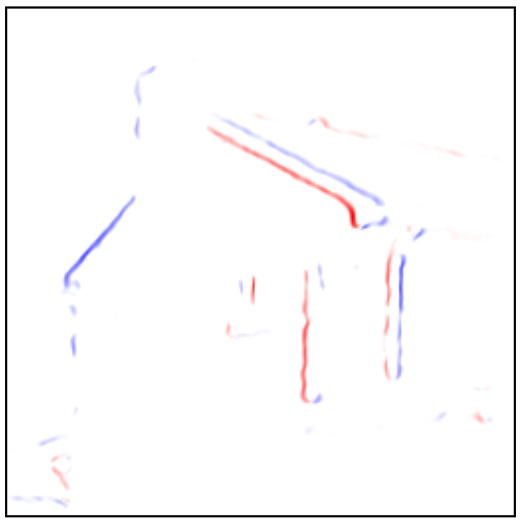
(b) v_x^k



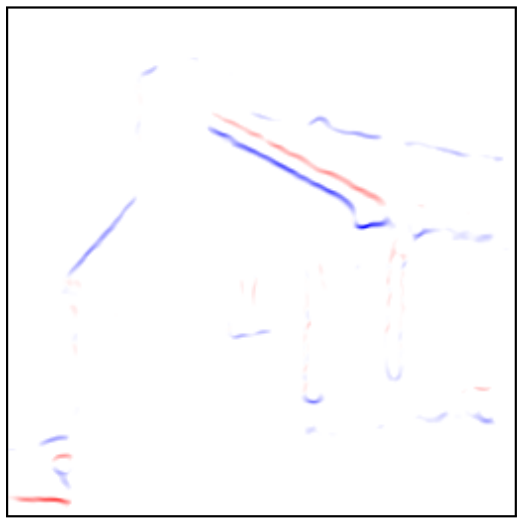
(c) v_y^k

Iteration $k = 7$

(a) f^k



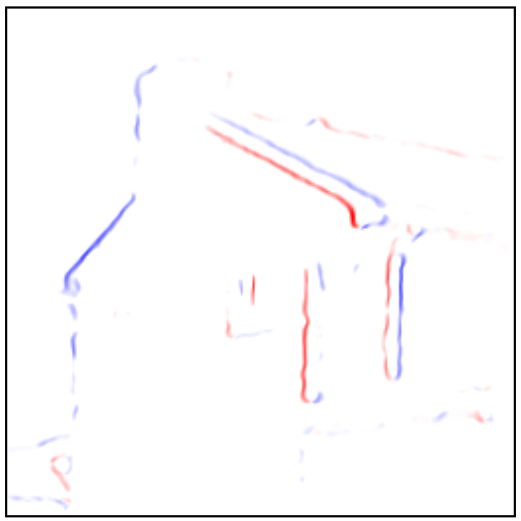
(b) v_x^k



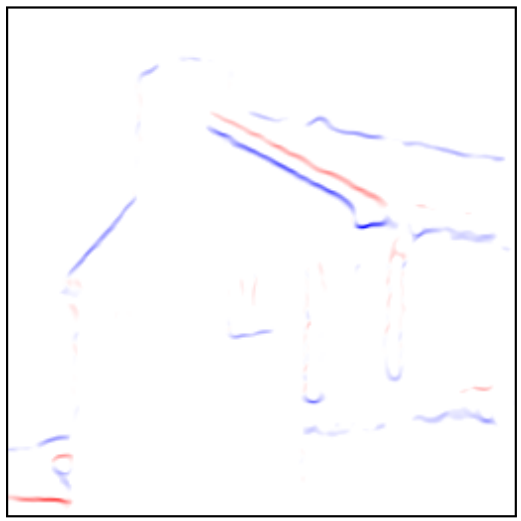
(c) v_y^k

Iteration $k = 8$

(a) f^k

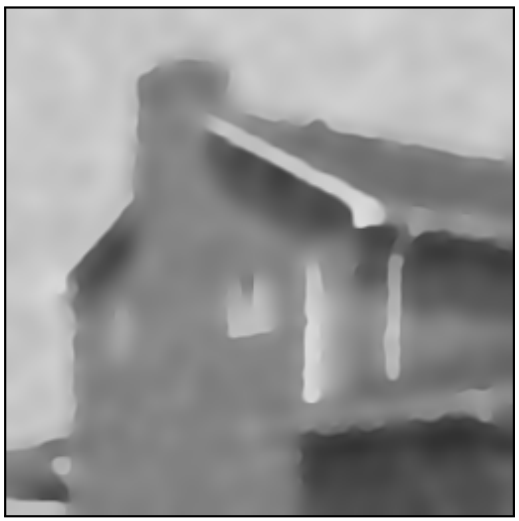


(b) v_x^k

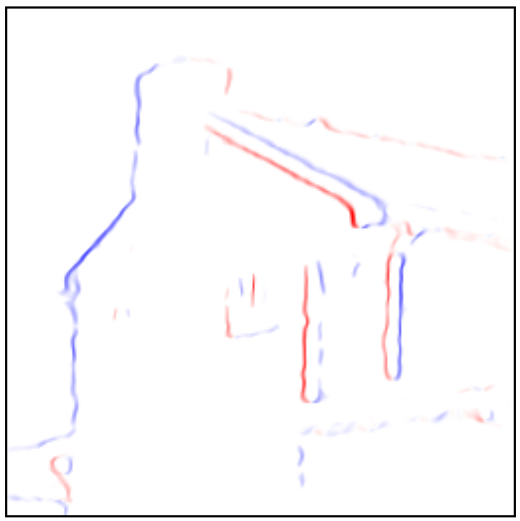


(c) v_y^k

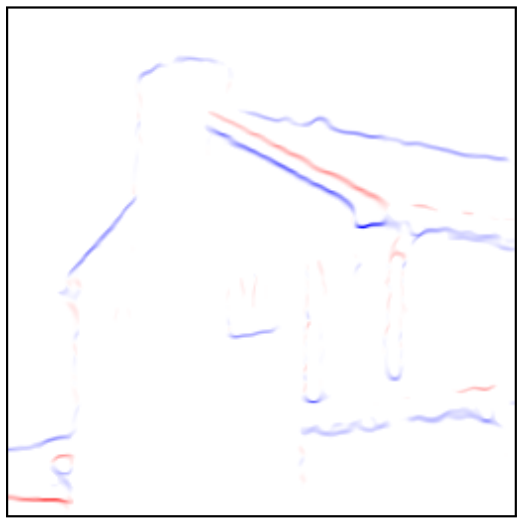
Iteration $k = 10$



(a) f^k

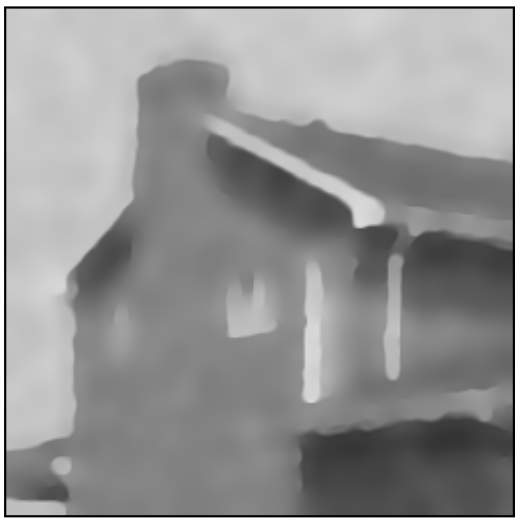


(b) v_x^k

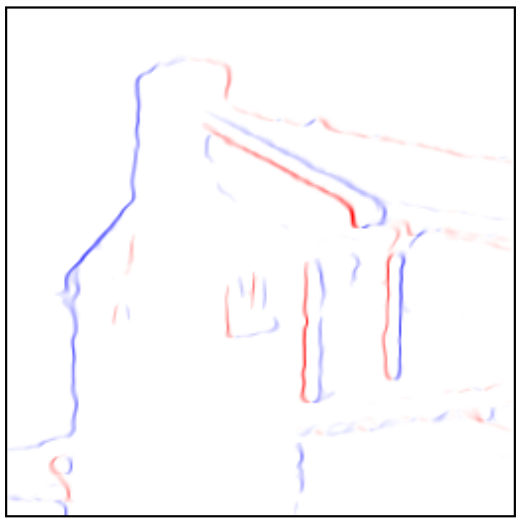


(c) v_y^k

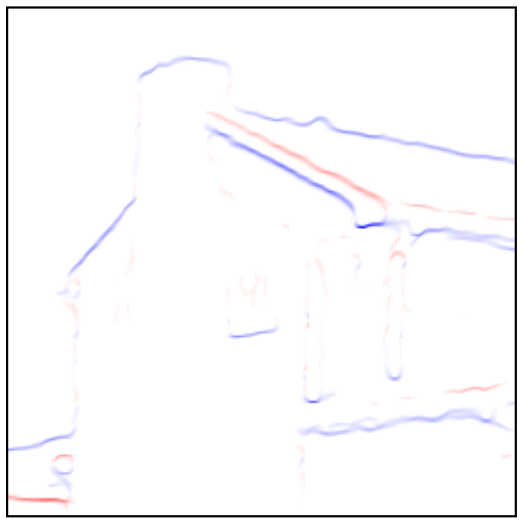
Iteration $k = 13$



(a) f^k

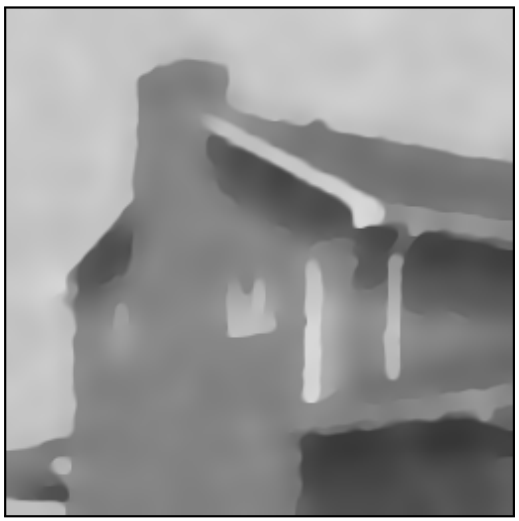


(b) v_x^k

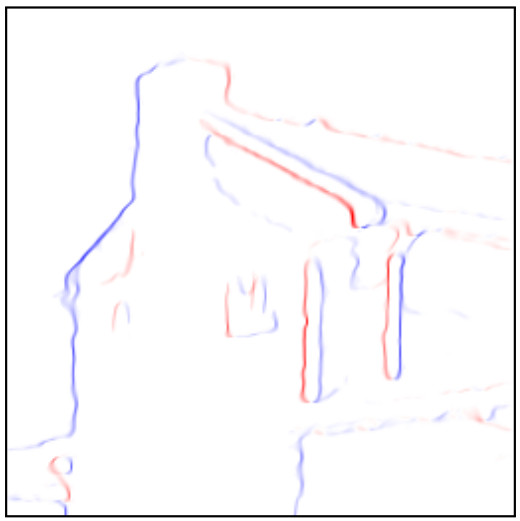


(c) v_y^k

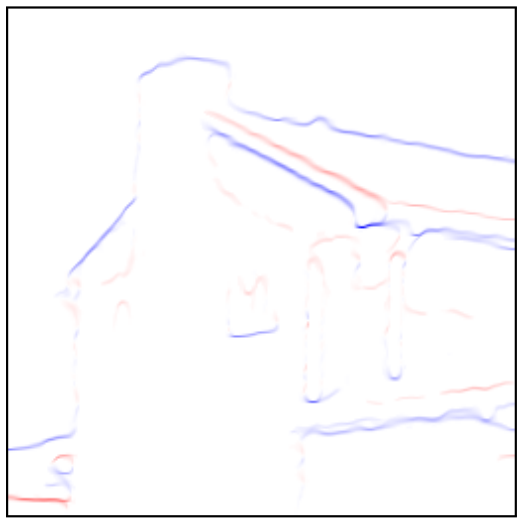
Iteration $k = 17$



(a) f^k

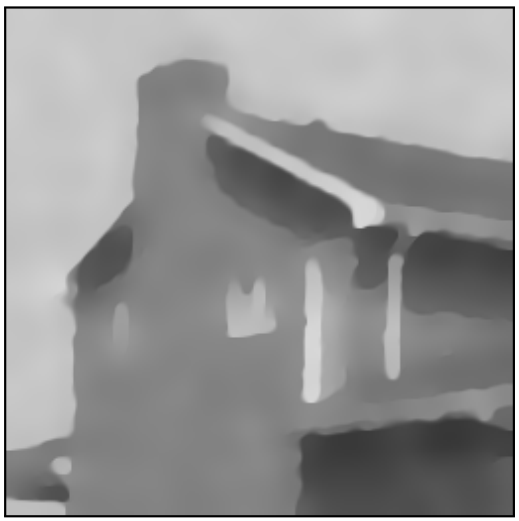


(b) v_x^k

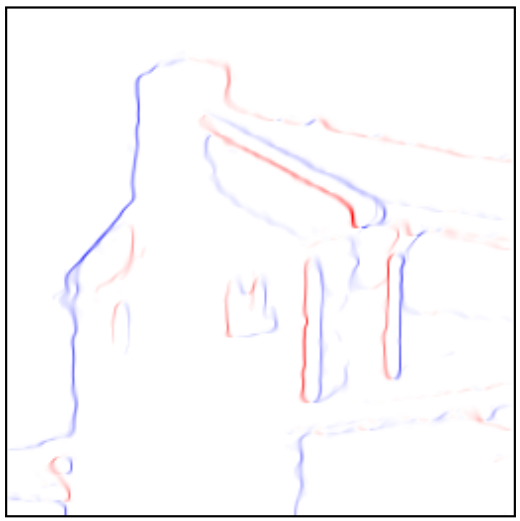


(c) v_y^k

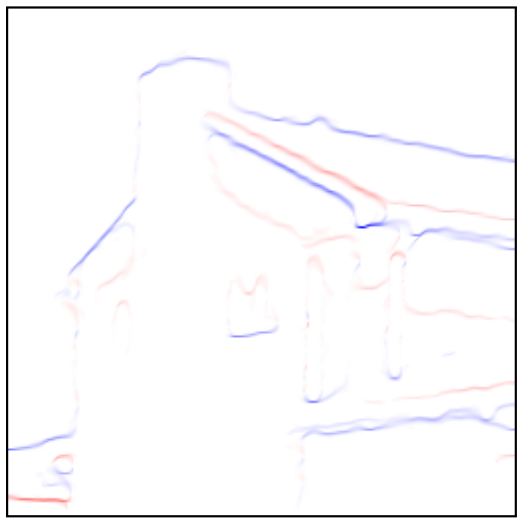
Iteration $k = 22$



(a) f^k

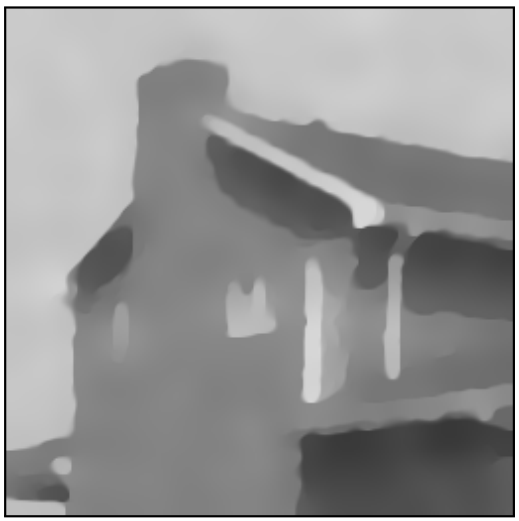


(b) v_x^k

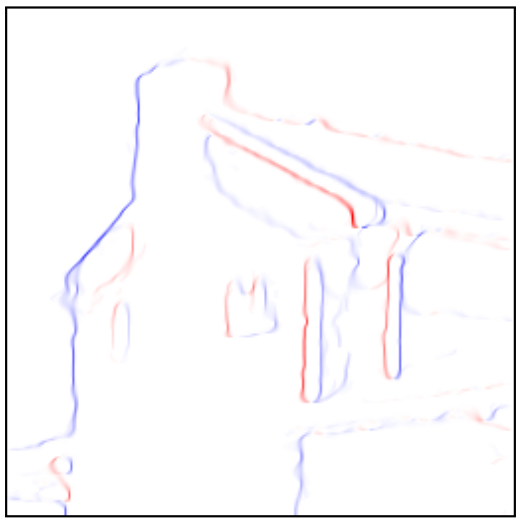


(c) v_y^k

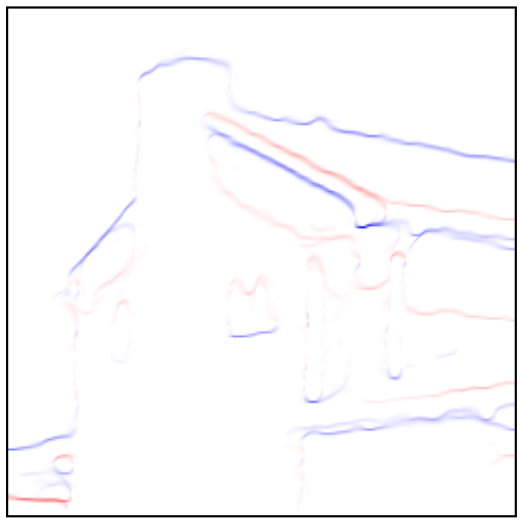
Iteration $k = 28$



(a) f^k

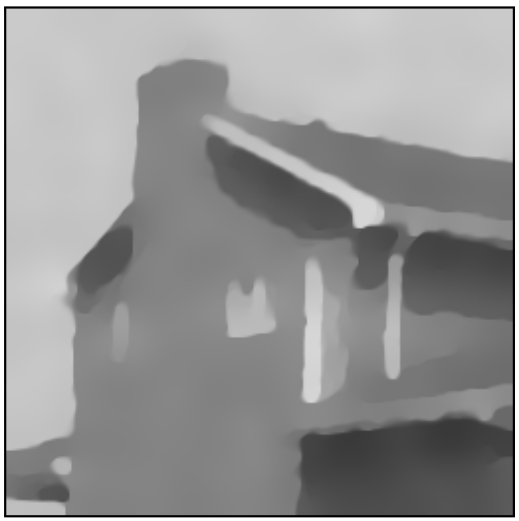


(b) v_x^k

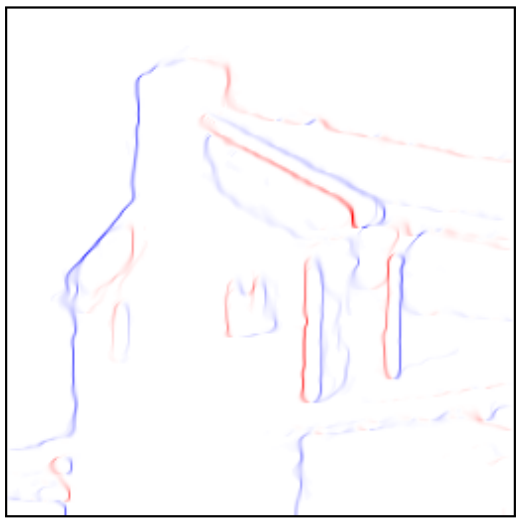


(c) v_y^k

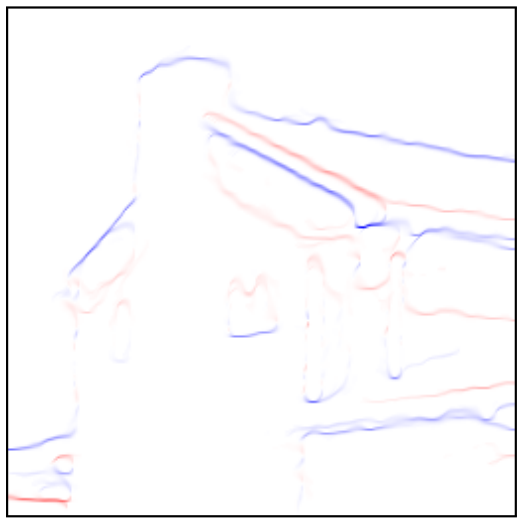
Iteration $k = 35$



(a) f^k

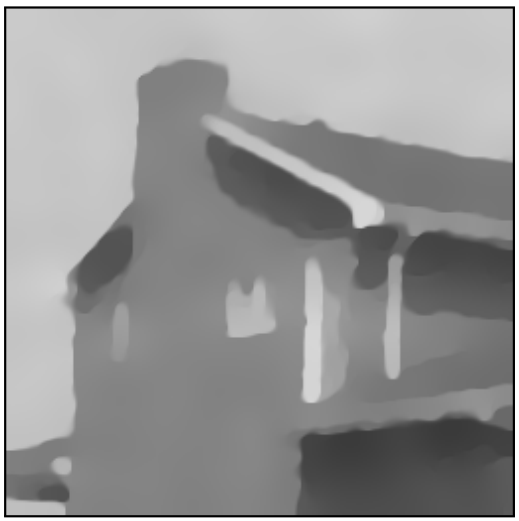


(b) v_x^k

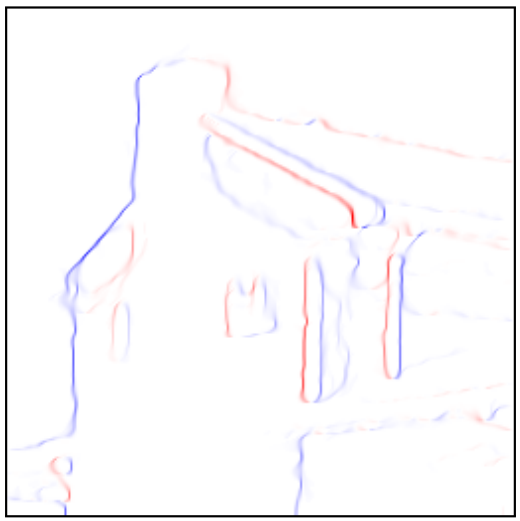


(c) v_y^k

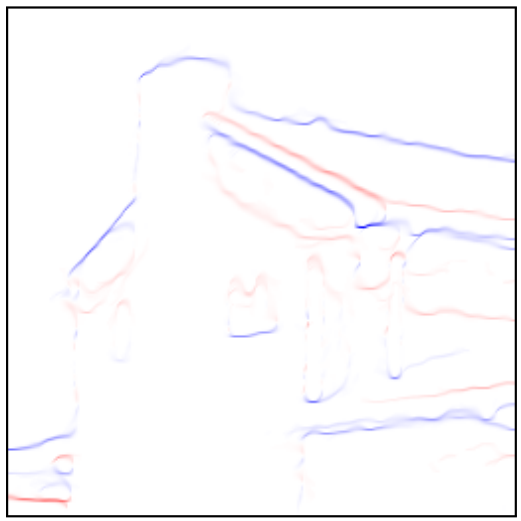
Iteration $k = 43$



(a) f^k



(b) v_x^k



(c) v_y^k

General comments on ADMM

- Easy-to-implement stopping criteria based on primal and dual residuum exist.
- Tuning of ρ is essential for fast convergence, but there are automatic tuning rules based on primal and dual residuum.
- Sub-problems can (and should!) be solved approximately. This is the key issue in designing fast ADMM schemes.
- Various modifications and extensions exist.

The best (and very extensive) reference for ADMM is given by

S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, 2011
Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers
Foundations and Trends in Machine Learning, 3(1).

Outline

Let $g_{\text{true}} = Af_{true}$ be the true data and

$$f_\alpha := \operatorname*{argmin}_f \left\{ \frac{1}{2}\|Af - g_{\text{true}}\|_2^2 + \alpha\Psi(f) \right\}$$

Due to the minimizing properties, we have

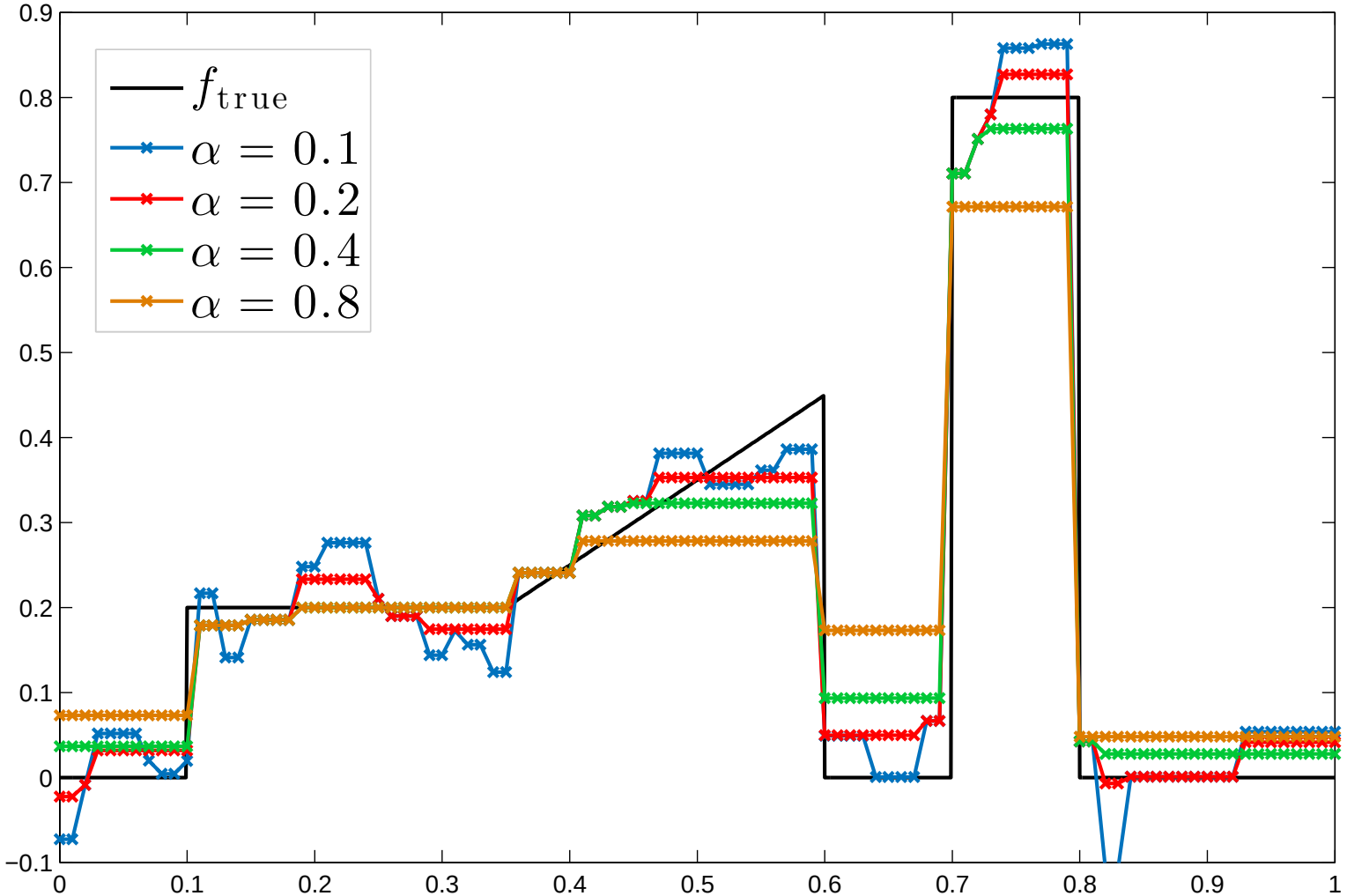
$$\frac{1}{2}\|Af_\alpha - g_{\text{true}}\|_2^2 + \alpha\Psi(f_\alpha) \leqslant \frac{1}{2}\|Af_{true} - g_{\text{true}}\|_2^2 + \alpha\Psi(f_{true}) = \alpha\Psi(f_{true})$$

and therefore, $\Psi(f_\alpha) \leqslant \Psi(f_{true})$.

This means that regularized solutions always carry a systematic bias in terms of the regularization functional.

How does this look like for TV?

The shortcomings of TV regularization



We loose contrast and fine structure, a simple re-scaling won't help!

The shortcomings of TV regularization

$$\tilde{g} = Af_{true} + \nu, \qquad \nu \sim \mathcal{N}\left(0, \sigma^2 I_{n_x \times n_y}\right)$$

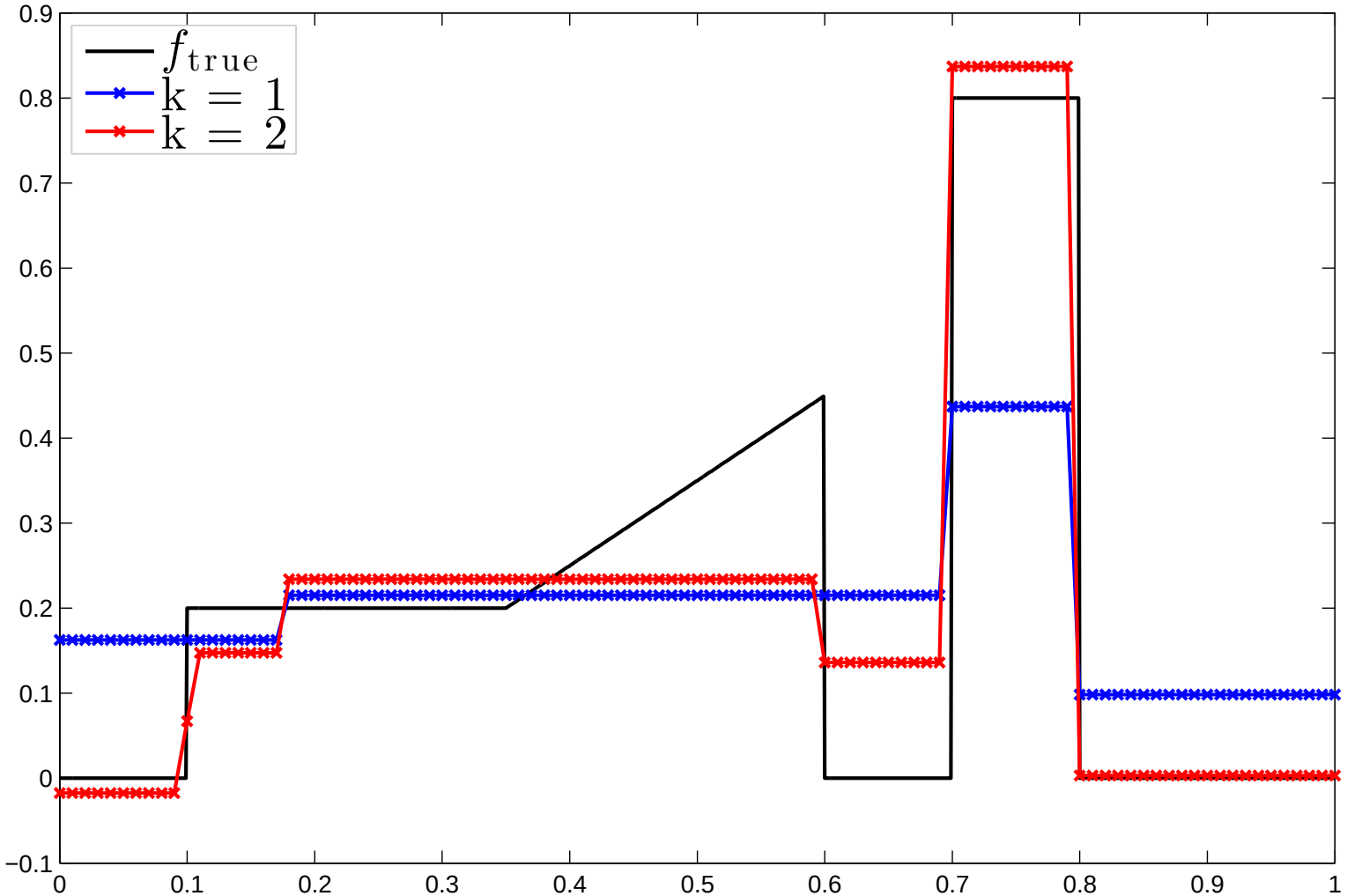
(a) f_{true} (b) f_α (c) $\tilde{g} - Af_\alpha$

$b = \tilde{g} - Af_\alpha$ should be white noise, but there is still a lot of structure in it! These structures have been shrunk too strongly. Can we prevent that by "adding them back" to the signal and re-re-run the regularization as

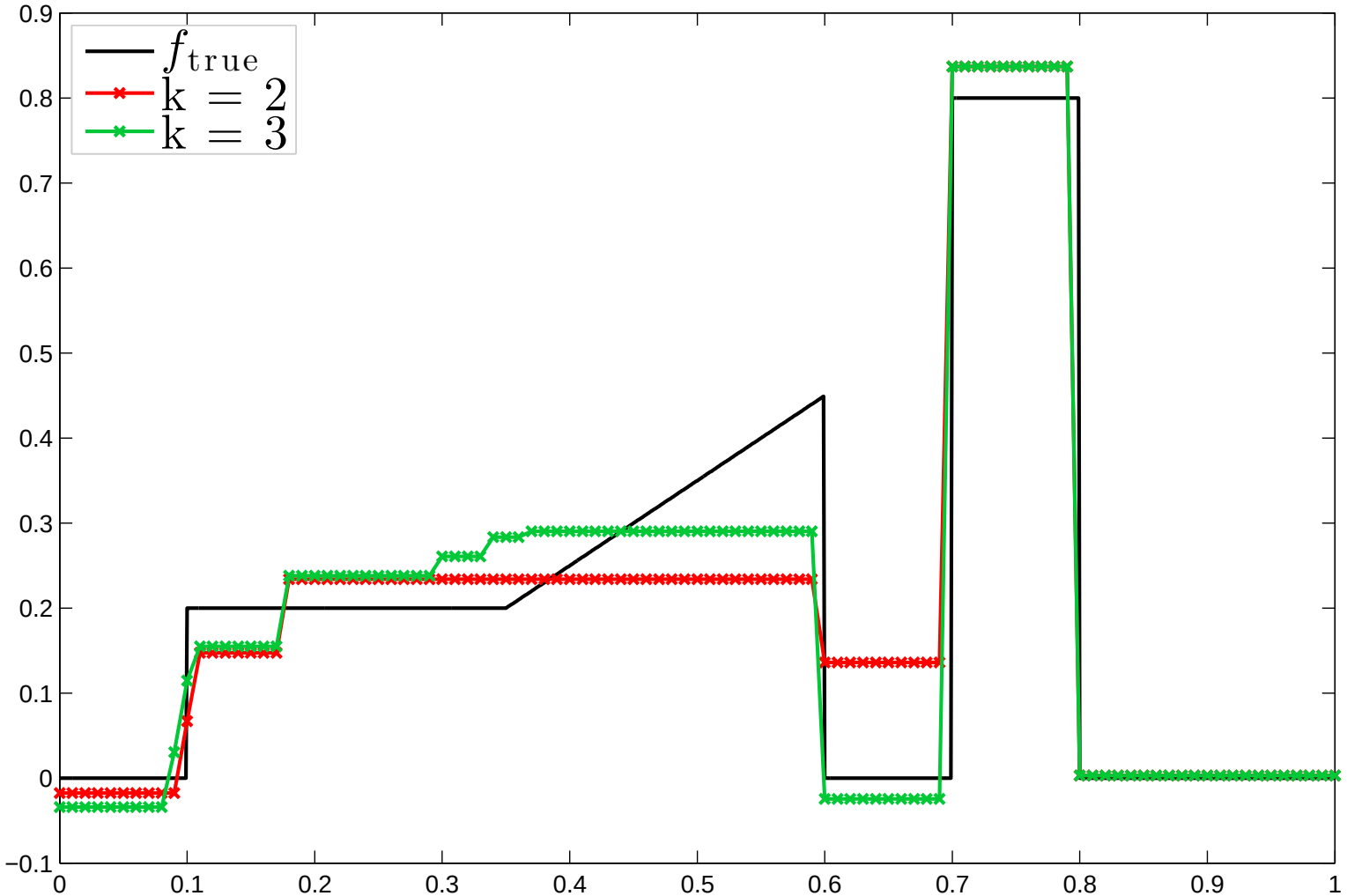
$$f_\alpha^2 := \underset{f}{\operatorname{argmin}} \left\{ \tfrac{1}{2} \|Af - (\tilde{g} + b)\|_2^2 + \alpha TV(f) \right\}$$

and even iterate this scheme?

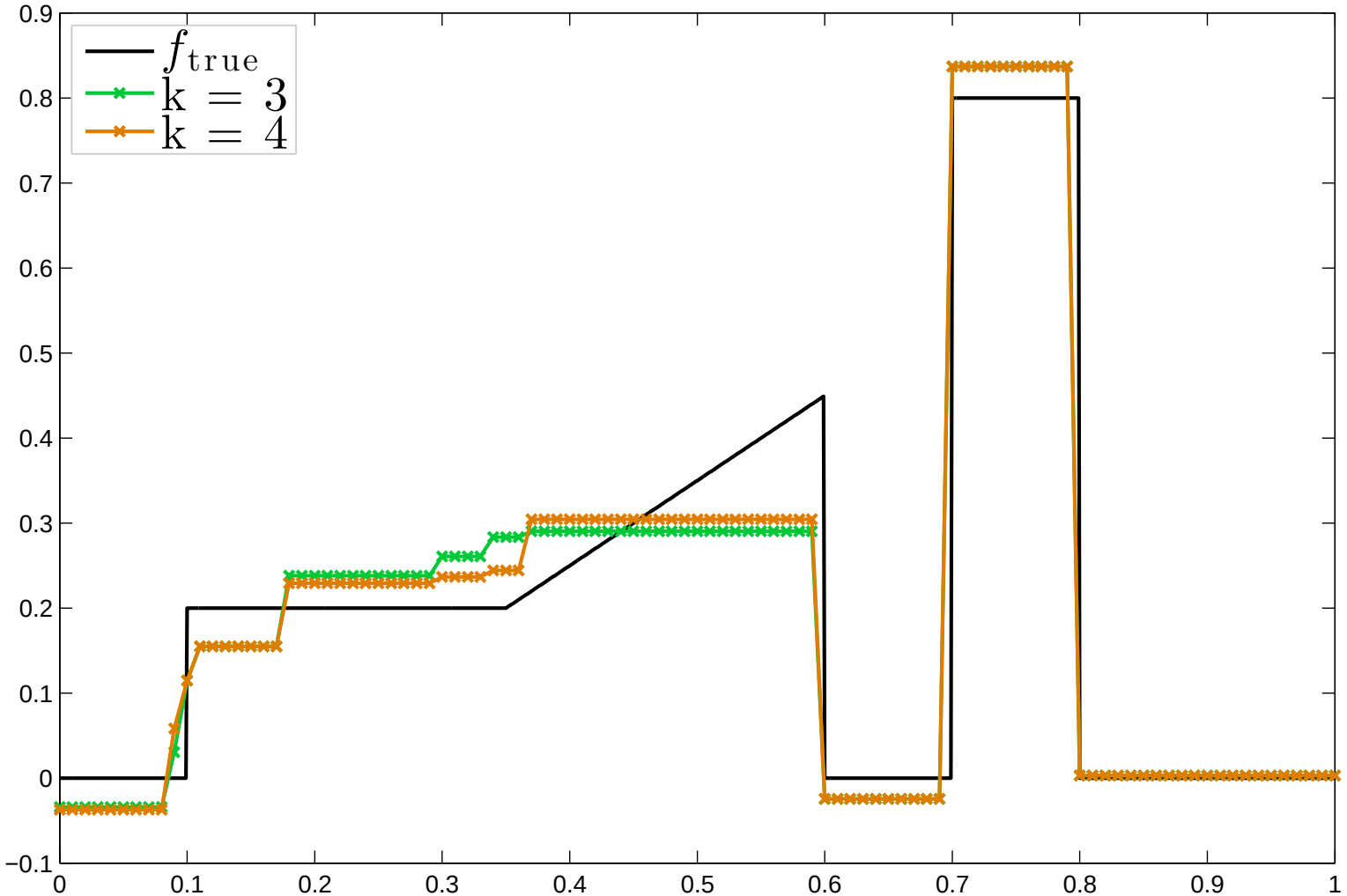
"Adding back noise" for 1D TV denoising



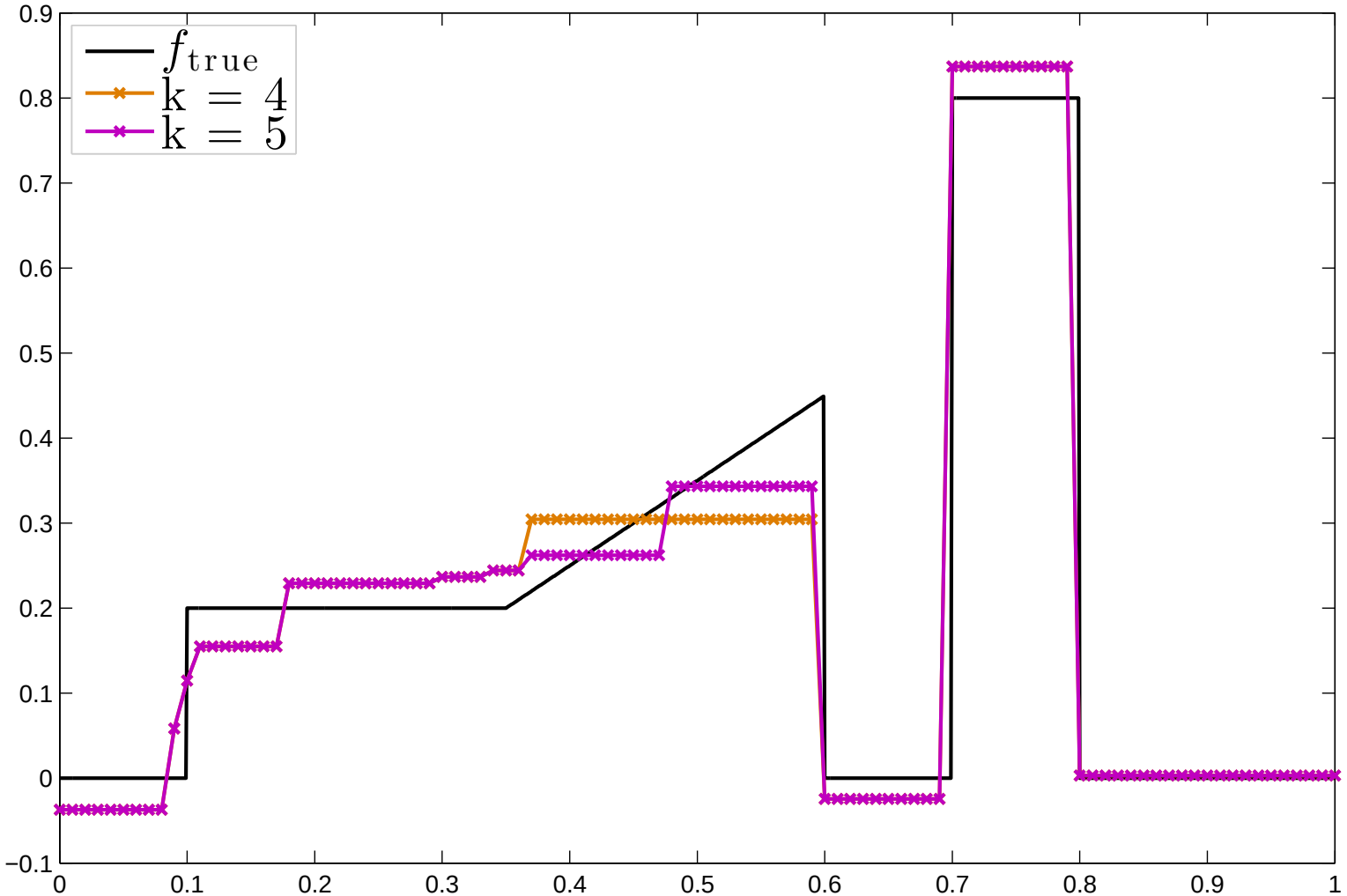
"Adding back noise" for 1D TV denoising



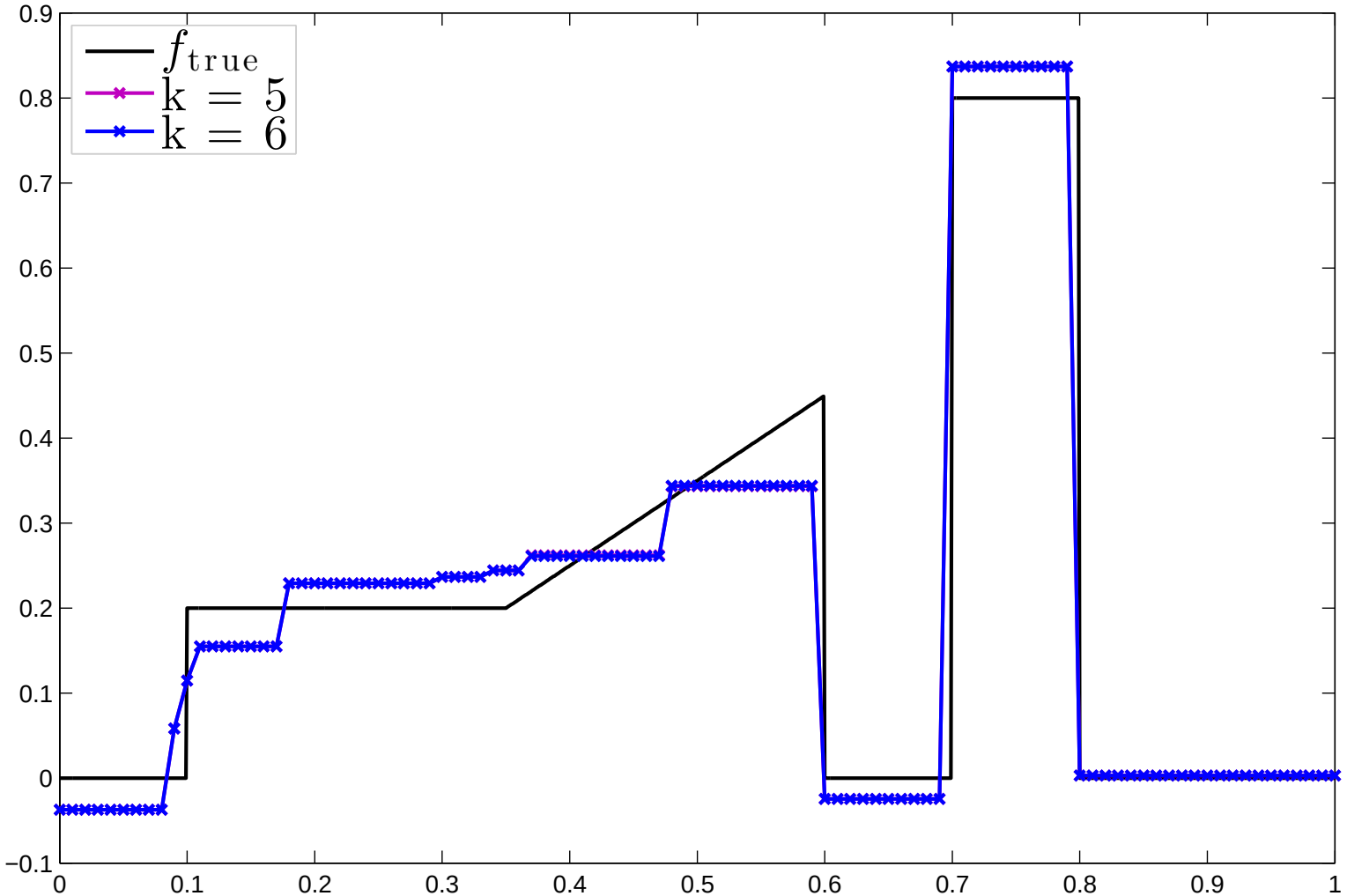
"Adding back noise" for 1D TV denoising



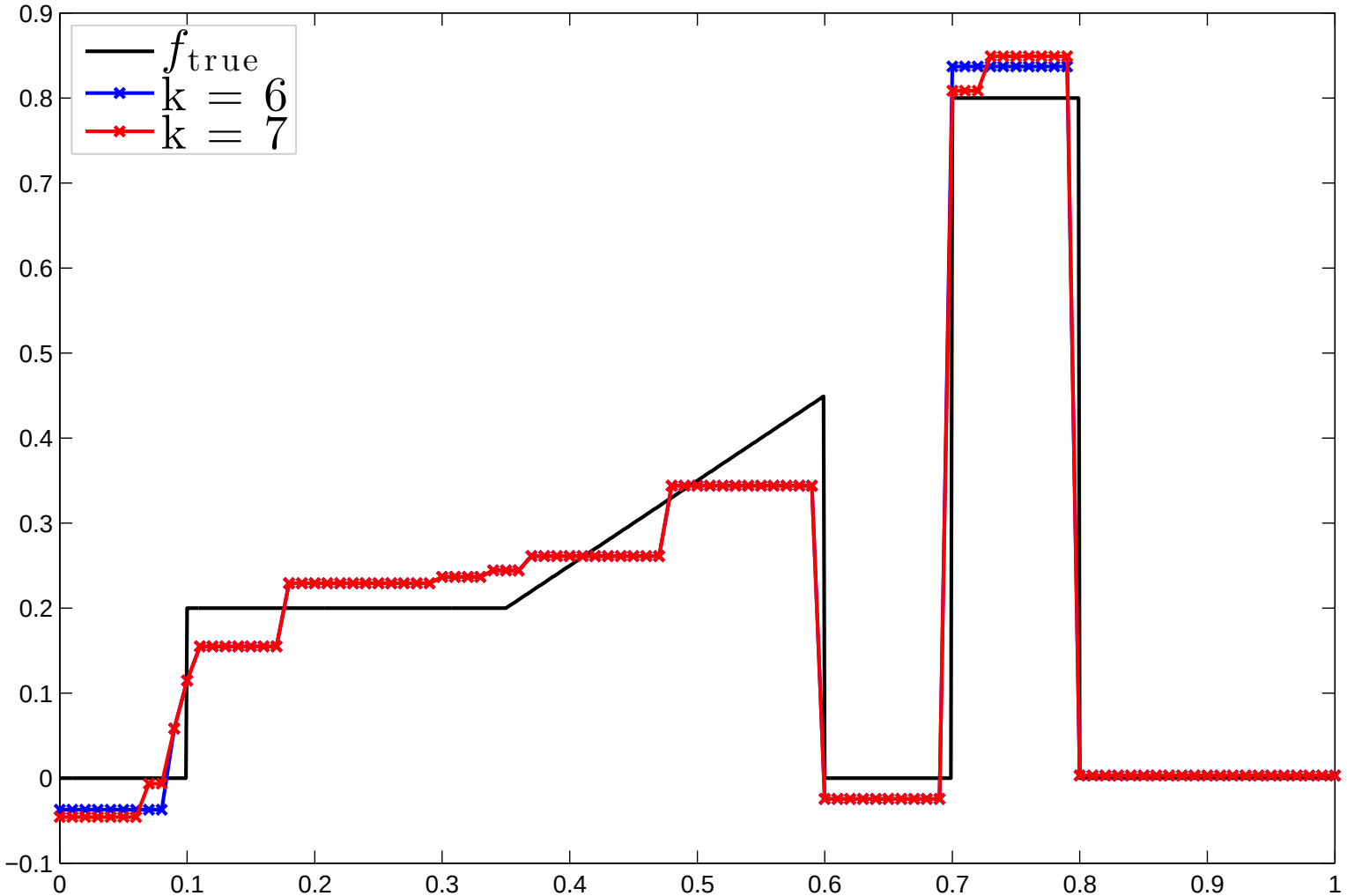
"Adding back noise" for 1D TV denoising



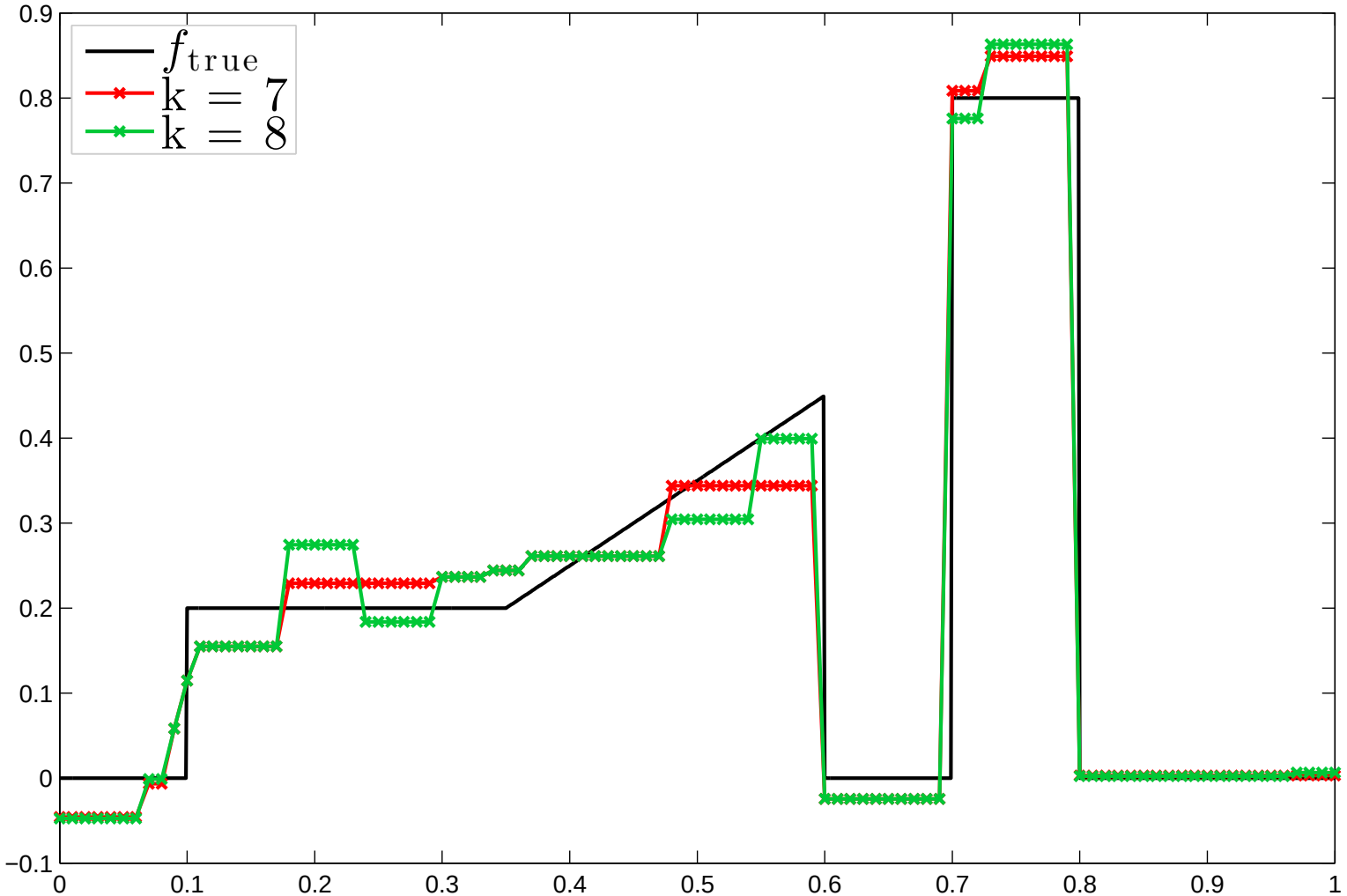
"Adding back noise" for 1D TV denoising



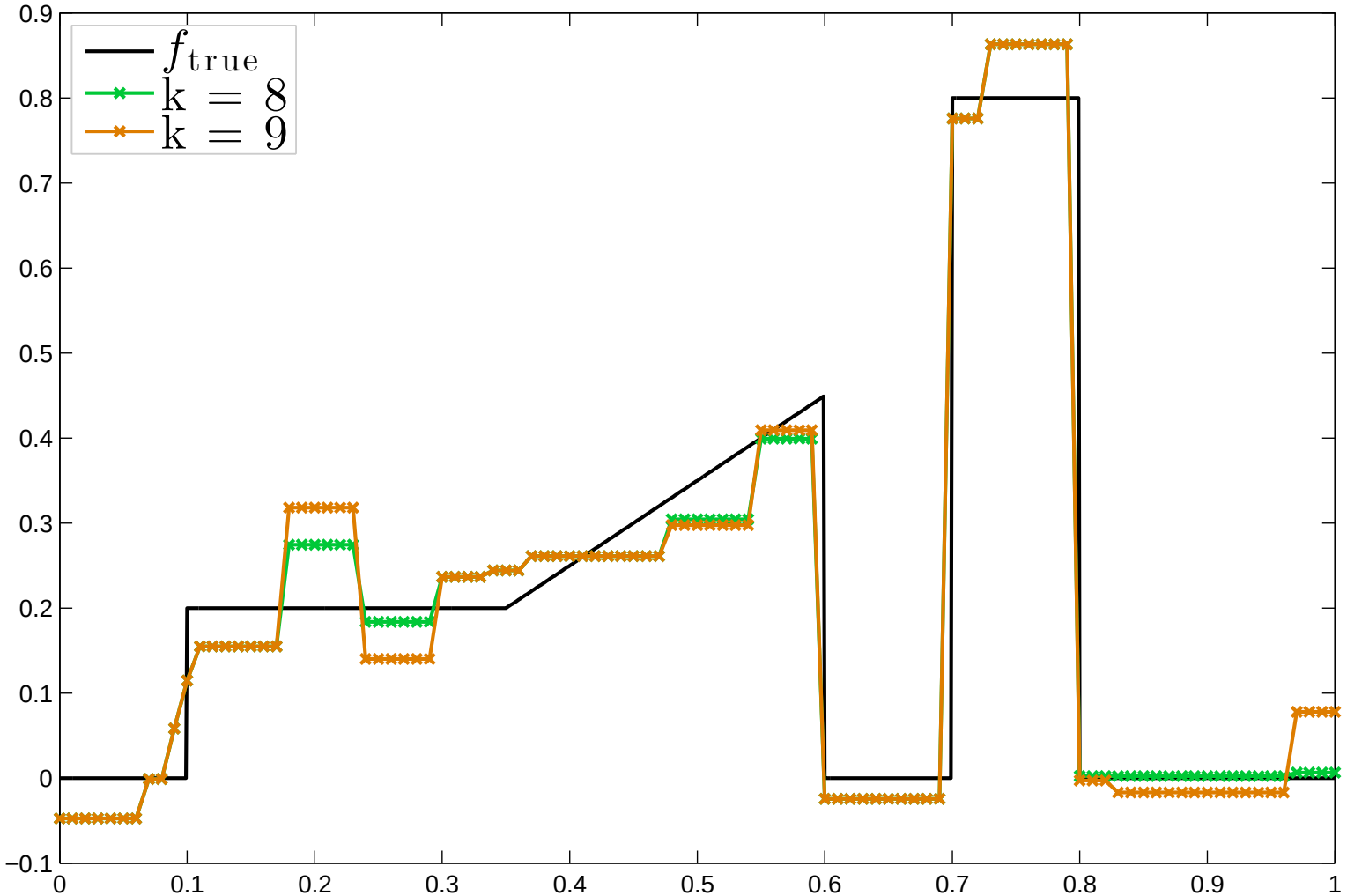
"Adding back noise" for 1D TV denoising



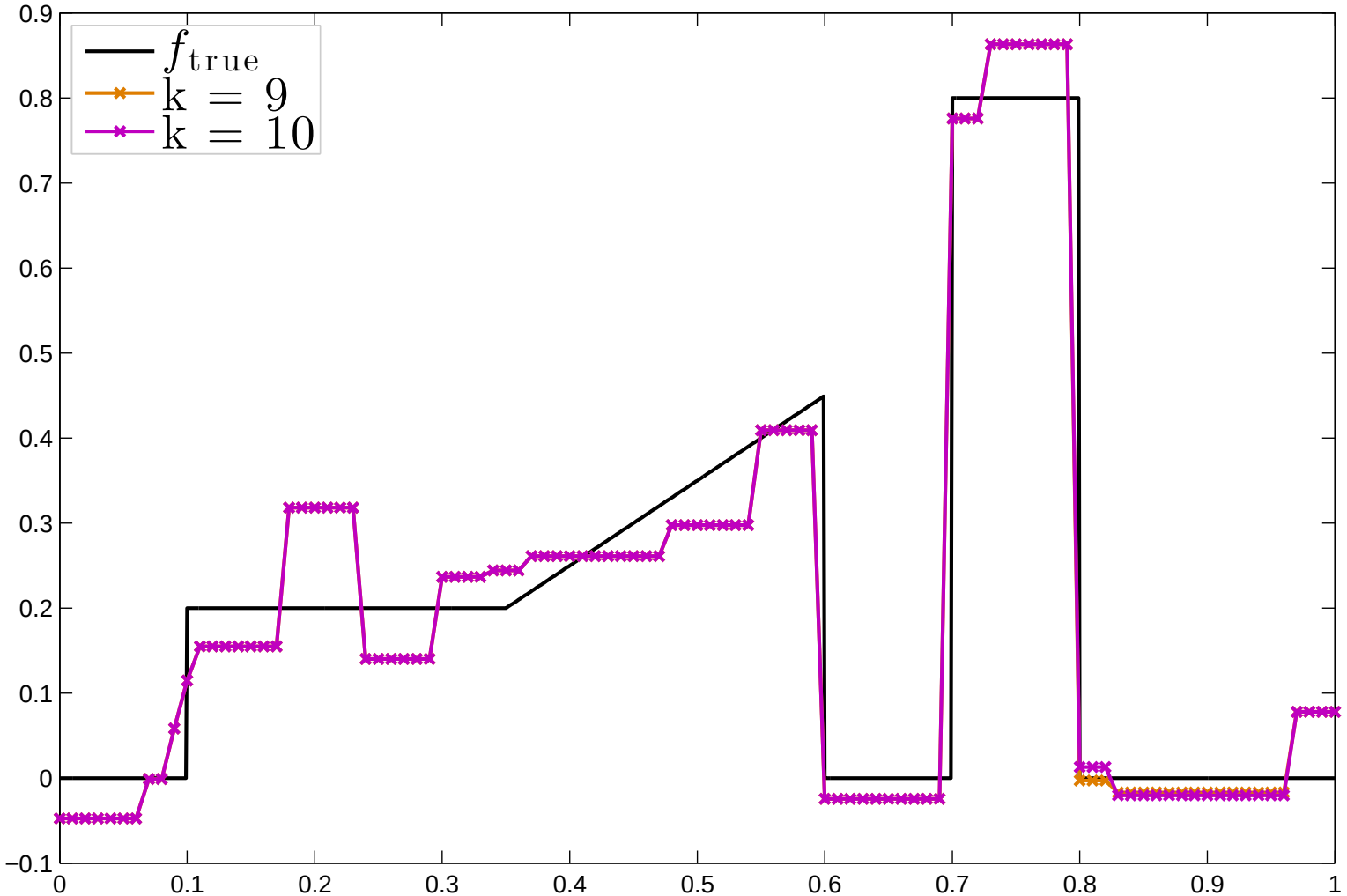
"Adding back noise" for 1D TV denoising



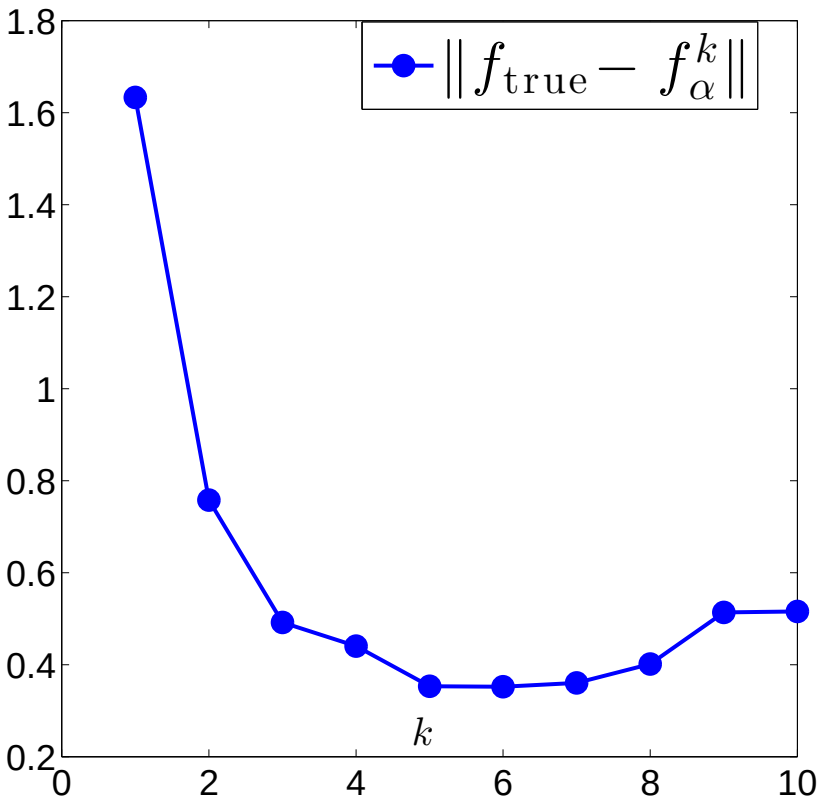
"Adding back noise" for 1D TV denoising

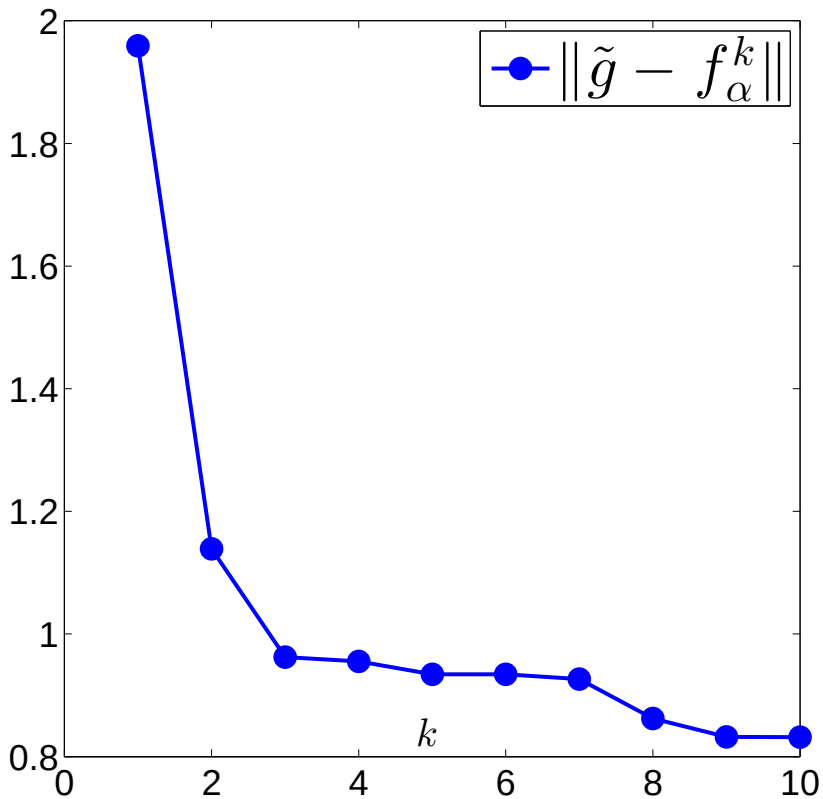


"Adding back noise" for 1D TV denoising



"Adding back noise" for 1D TV denoising



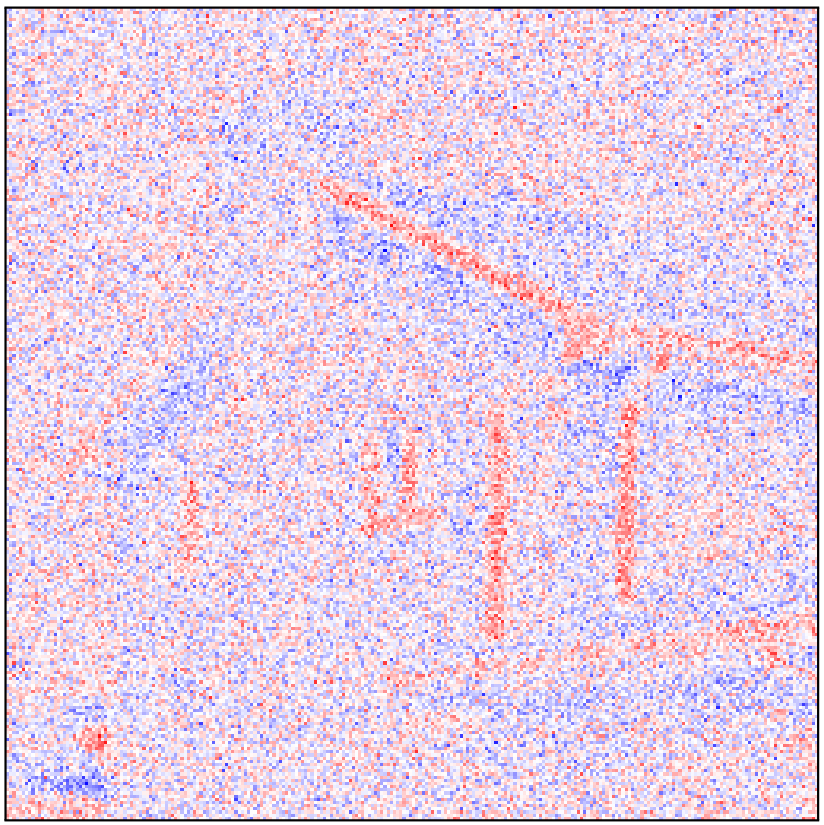


For $k = 1000$, $f_\alpha^k \approx \tilde{g}$!

What is happening in the heuristic iteration?

Iteration $k = 1$

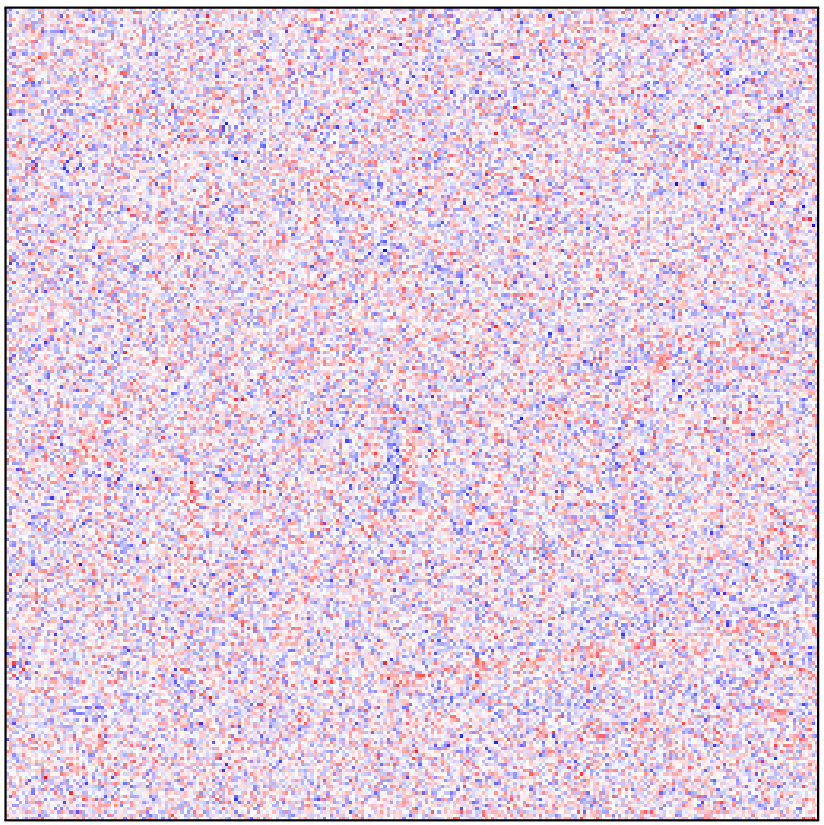
(a) f_α^k



(b) $\tilde{g} - Af_\alpha^k$

Iteration $k = 2$

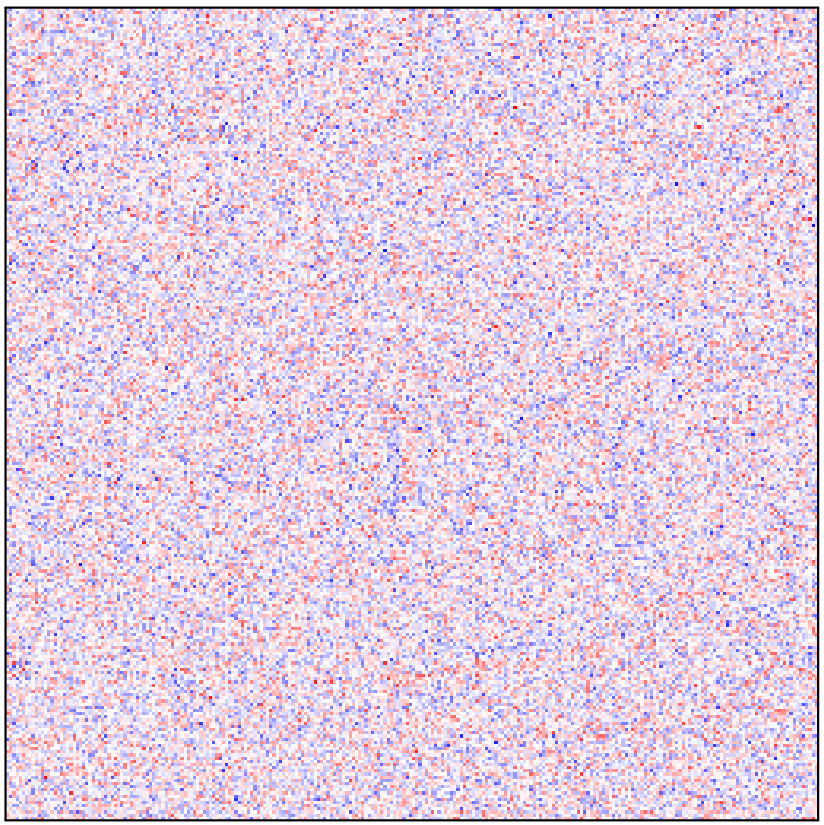
(a) f_α^k



(b) $\tilde{g} - Af_\alpha^k$

Iteration $k = 3$

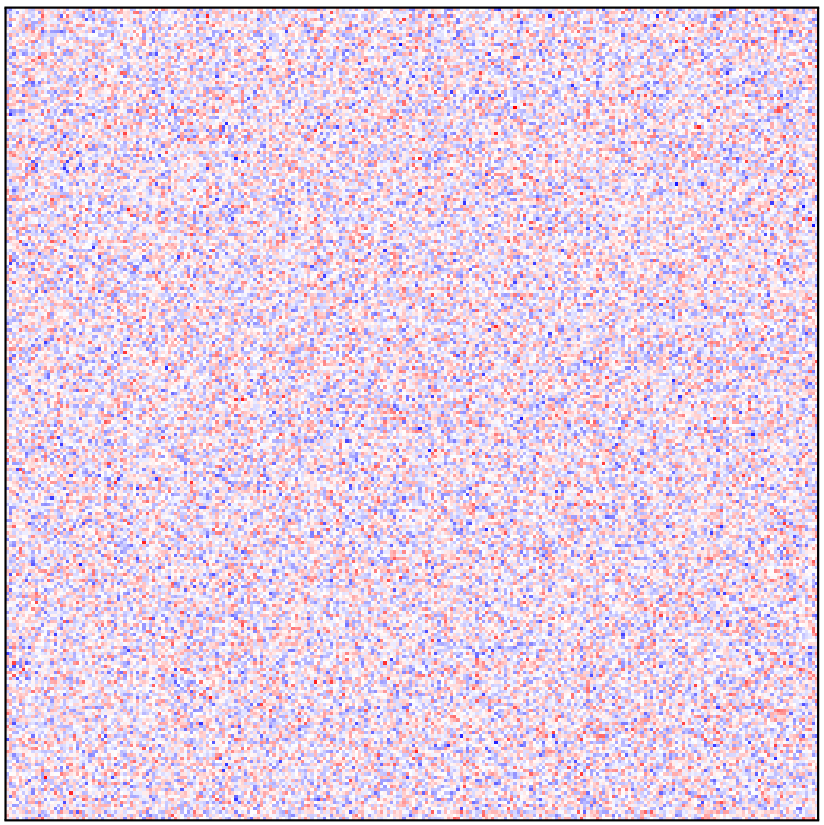
(a) f_α^k



(b) $\tilde{g} - Af_\alpha^k$

Iteration $k = 4$

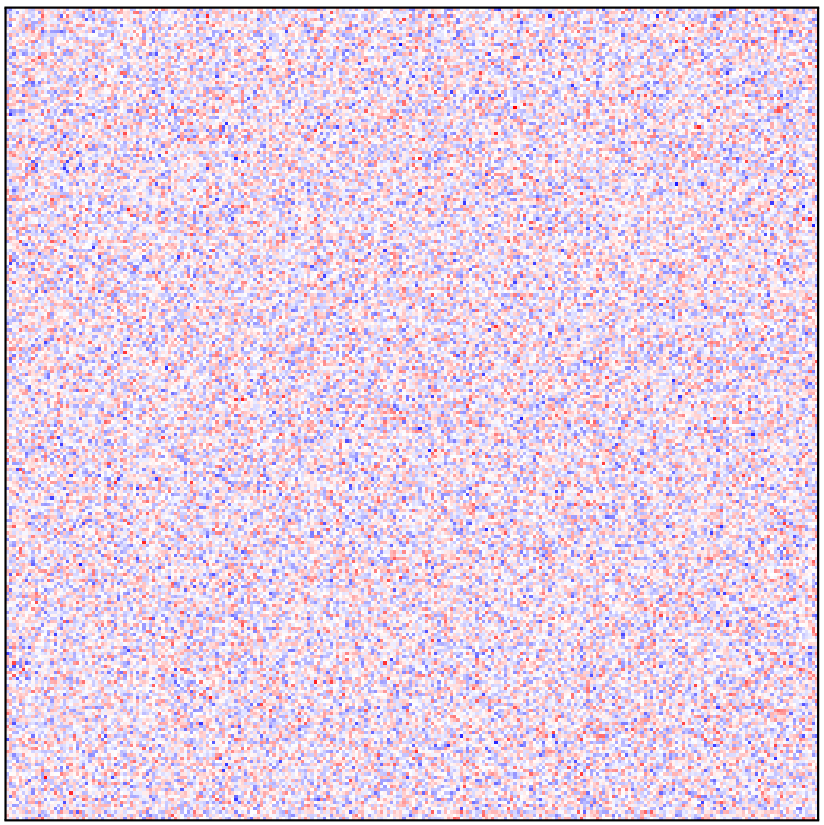
(a) f_α^k



(b) $\tilde{g} - Af_\alpha^k$

Iteration $k = 5$

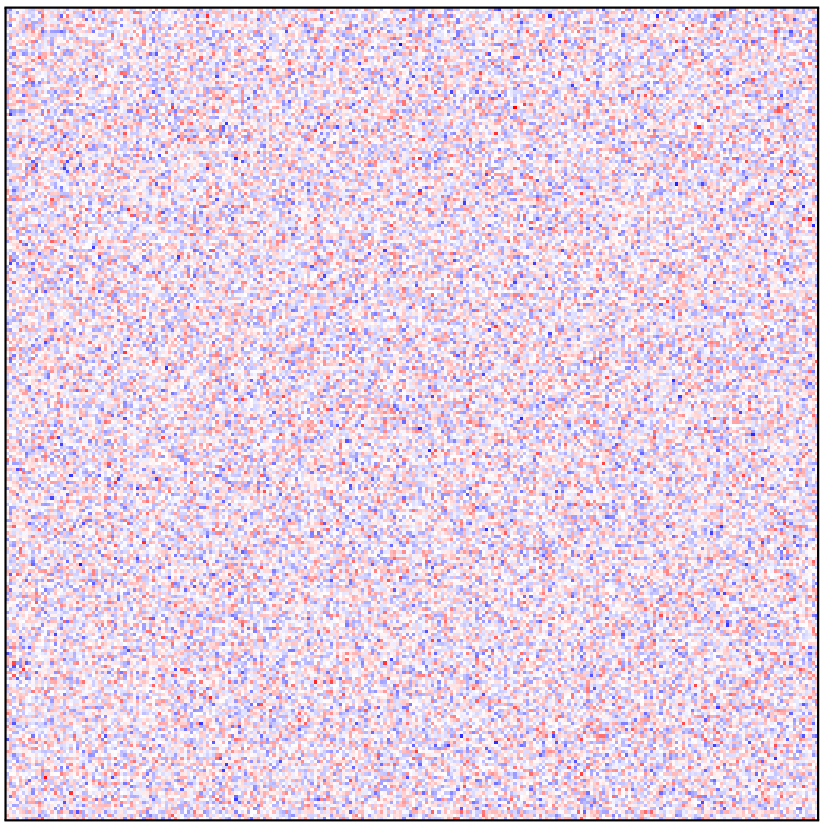
(a) f_α^k



(b) $\tilde{g} - Af_\alpha^k$

Iteration $k = 6$

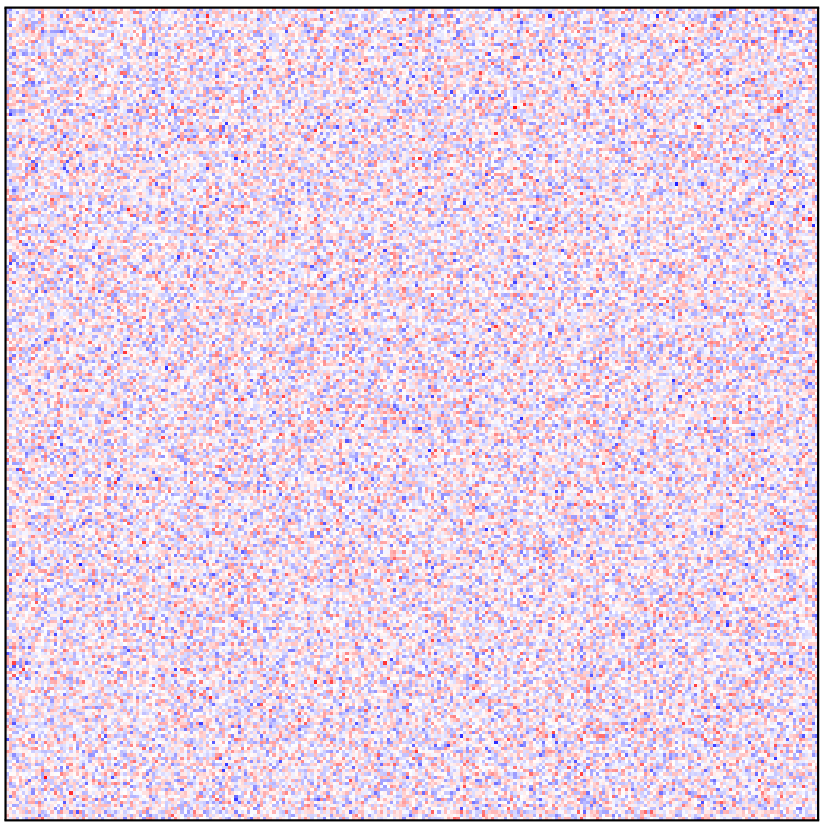
(a) f_α^k



(b) $\tilde{g} - Af_\alpha^k$

Iteration $k = 7$

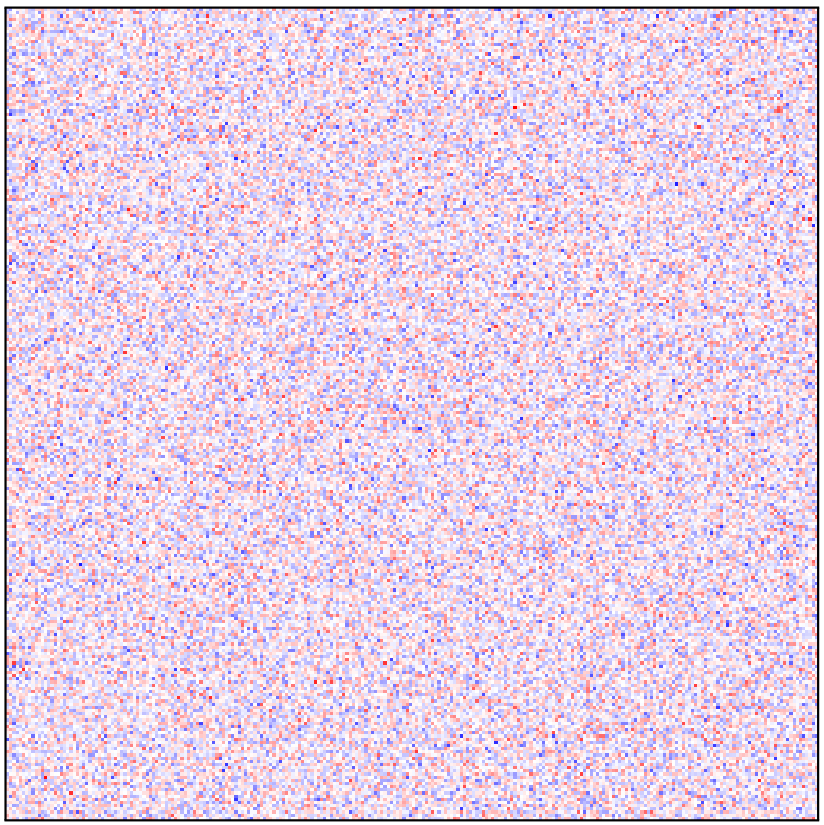
(a) f_α^k



(b) $\tilde{g} - Af_\alpha^k$

Iteration $k = 8$

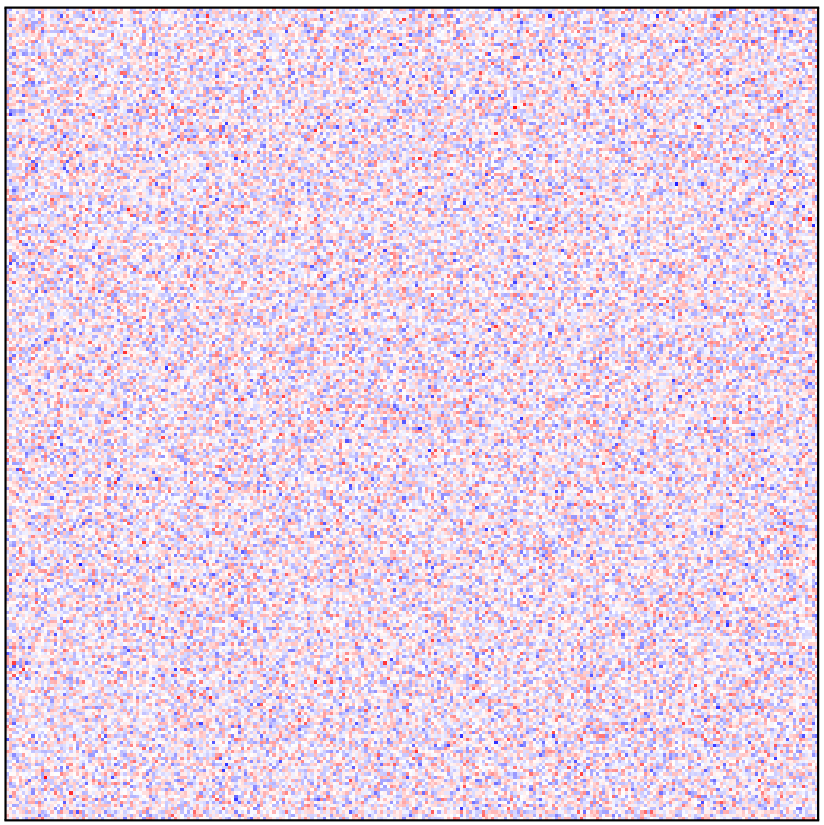
(a) f_α^k



(b) $\tilde{g} - Af_\alpha^k$

Iteration $k = 9$

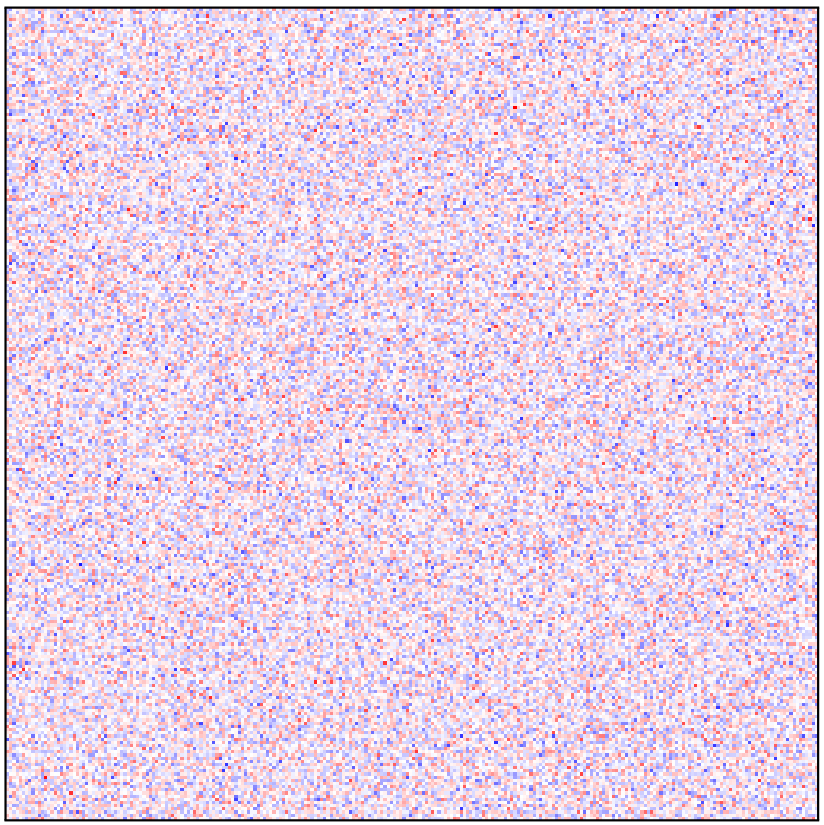
(a) f_α^k



(b) $\tilde{g} - Af_\alpha^k$

Iteration $k = 10$

(a) f_α^k



(b) $\tilde{g} - Af_\alpha^k$

To understand what is happening in the heuristic iteration, we need some convex analysis:

For a proper, convex functional $\Psi : \mathbb{R}^n \longrightarrow \mathbb{R} \cup \{\infty\}$, the *subdifferential* $\partial\Psi(u)$ at f is defined as

$$\partial\Psi(f) := \{p \in R^n \mid \Psi(g) \geqslant \Psi(f) + \langle p, g - f\rangle\,,\ \forall g \in \mathbb{R}^n\}\,.$$

- $p \in \partial\Psi(f)$ is called a *subgradient* of Ψ in f.
- Subdifferentiability extends (Fréchet-)differentiability for the important class of convex functionals: If Ψ is differentiable in f, then $\partial\Psi(f) = \{\Psi'(f)\}$

Subdifferentiability in 1D

- $\Psi(f) + p(g - f)$ describes a line through $(f, \Psi(f))$ with slope p.
- The set of all slopes p such that this line is either touching or below the graph of $\Psi(f)$ is the subderivative $\partial\Psi(f)$.
- It is a non-empty, closed interval $[p_-, p_+]$, where

$$p_- = \lim_{h \searrow 0} \frac{\Psi(f) - \Psi(f-h)}{h}, \qquad p_+ = \lim_{h \searrow 0} \frac{\Psi(f+h) - \Psi(f)}{h}.$$

- Both limits exist and fulfill $p_- \leqslant p_+$.
- If the subderivative contains only one element, i.e., $p_- = p_+$, then Ψ is differentiable at f and $\Psi'(f) = p_- = p_+$.

Classical example:

$$\partial|x| = \begin{cases} 1 & \text{for } x > 0 \\ [-1, 1] & \text{for } x = 0 \\ -1 & \text{for } x < 0 \end{cases}$$

Subdifferentiability and optimality conditions

A point $f \in R^n$ is a minimum of a *smooth* proper, convex functional $\Psi : \mathbb{R}^n \longrightarrow \mathbb{R} \cup \{\infty\}$ if and only if $0 = \Psi'(f)$.

Subdifferentiability and optimality conditions

A point $f \in R^n$ is a minimum of a proper, convex functional $\Psi : \mathbb{R}^n \longrightarrow \mathbb{R} \cup \{\infty\}$ if and only if $0 \in \partial\Psi(f)$.

A point $f \in R^n$ is a minimum of a proper, convex functional $\Psi : \mathbb{R}^n \longrightarrow \mathbb{R} \cup \{\infty\}$ if and only if .

Proof: If $0 \in \partial\Psi(f)$, we have that

$$0 = \langle 0, g - f \rangle \leqslant \Psi(g) - \Psi(f) \qquad \forall g \in \mathbb{R}^n,$$

and thereby, f is a global minimizer of Ψ. If $0 \notin \partial\Psi(f)$, there must be at least one $g \in R^n$ such that

$$\Psi(g) < \Psi(f) + \langle 0, g - f \rangle = \Psi(f),$$

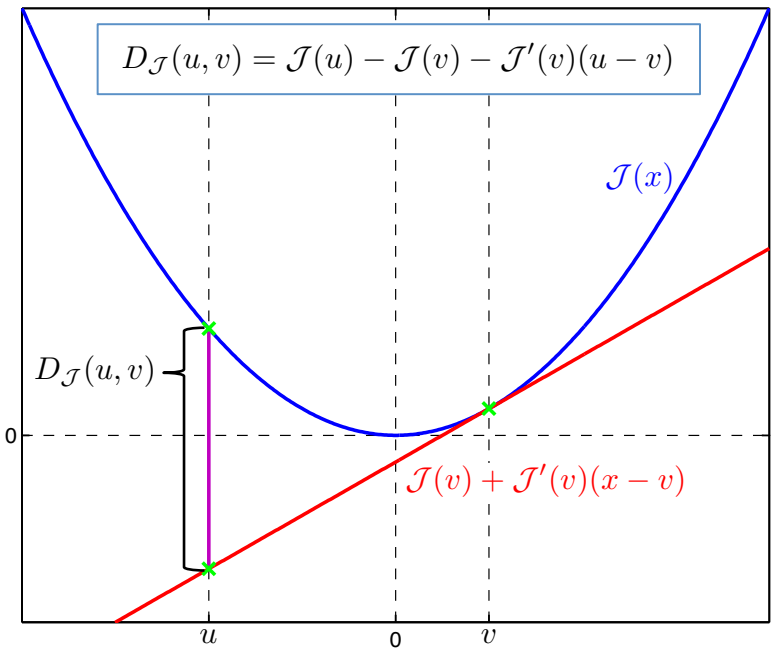
and thereby, f cannot be a global minimizer of Ψ.

The uniqueness of the minimizer can only be guaranteed if $\Psi(f)$ is *strictly convex*.

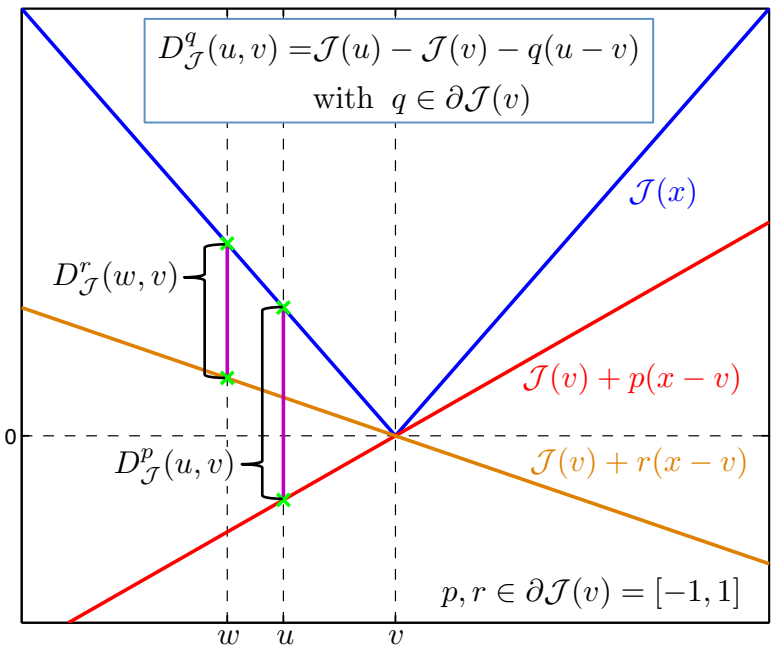
Bregman distances

For a proper, convex functional $\Psi : \mathbb{R}^n \longrightarrow \mathbb{R} \cup \{\infty\}$, the *Bregman distance* $D_\Psi^p(f, g)$ between $f, g \in \mathbb{R}^n$ for a subgradient $p \in \partial\Psi(g)$ is defined as

$$D_\Psi^p(f, g) = \Psi(f) - \Psi(g) - \langle p, f - g \rangle, \qquad p \in \partial\Psi(g)$$



(c) $\mathcal{J}(x) = x^2$



(d) $\mathcal{J}(x) = |x|$

Basically, $D_\Psi(f, g)$ measures the difference between Ψ and its linearization in f at another point g

Bregman distances

- The Bregman distance is not a distance in the usual mathematical sense (i.e., a metric) as it is, in general, neither symmetric nor satisfies the triangle inequality.
- $D_\Psi(f, g) \geqslant 0$ and for strictly convex $\Psi(f)$, $D_\Psi(f, g) = 0$ implies $f = g$.
- Bregman distances have become an important tool in variational regularization
  - to derive error estimates and convergence rates (Burger & Osher, 2004).
  - to derive optimization schemes like the Split-Bregman algorithm (Goldstein & Osher, 2009) which is closely related to ADMM.
  - to enhance inverse methods by Bregman iterations (what we're doing right now!).

"Adding back noise" as a Bregman iteration

The optimality condition for the first iteration

$$f_\alpha^1 = \operatorname*{argmin}_f \ \tfrac{1}{2}\|\tilde{g} - Af\|_2^2 + \alpha\Psi(f)$$

is given as

$$0 \in -A^T(\tilde{g} - Af_\alpha^1) + \alpha\partial\Psi(f_\alpha^1) \quad \Longleftrightarrow \quad -A^T(\tilde{g} - Af_\alpha^1) \in \alpha\partial\Psi(f_\alpha^1)$$
$$\Longleftrightarrow \quad A^T b^1 = \alpha p^1 \quad \text{with} \quad b^1 = \tilde{g} - Af_\alpha^1, \quad p^1 \in \partial\Psi(f_\alpha^1)$$

Adding back the noise once amounts to:

$$f_\alpha^2 = \operatorname*{argmin}_f \ \tfrac{1}{2}\|\tilde{g} + b^1 - Af\|_2^2 + \alpha\Psi(f)$$
$$\Longleftrightarrow f_\alpha^2 = \operatorname*{argmin}_f \ \tfrac{1}{2}\|\tilde{g} - Af\|_2^2 + \alpha\Psi(f) - \langle A^T b^1, f\rangle$$
$$\Longleftrightarrow f_\alpha^2 = \operatorname*{argmin}_f \ \tfrac{1}{2}\|\tilde{g} - Af\|_2^2 + \alpha\left(\Psi(f) - \Psi(f_\alpha^1) - \langle p^1, f - f_\alpha^1\rangle\right)$$
$$\Longleftrightarrow f_\alpha^2 = \operatorname*{argmin}_f \ \tfrac{1}{2}\|\tilde{g} - Af\|_2^2 + \alpha D_\Psi^{p^1}(f, f_\alpha^1)$$

(we added and subtracted terms not depending on f!)

"Adding back noise" as a Bregman iteration

More general, the "adding back noise" iteration

$$f_\alpha^{k+1} = \underset{f}{\operatorname{argmin}} \ \tfrac{1}{2}\|\tilde{g} + b^k - Af\|_2^2 + \alpha\Psi(f)$$
$$b^{k+1} = b^k + \left(\tilde{g} - Af_\alpha^{k+1}\right)$$

is a specific reformulation of the *Bregman iteration*

$$f_\alpha^{k+1} = \underset{f}{\operatorname{argmin}} \ \mathcal{H}(f, \tilde{g}) + \alpha D_\Psi^{p^k}(f, f_\alpha^k)$$
$$p^k \in \partial\Psi(f_\alpha^k)$$

to solve

$$\min_f \Psi(f) \quad \text{subject to} \quad f \in \underset{f}{\operatorname{argmin}} \ \mathcal{H}(f, \tilde{g}).$$

L.M. Bregman, 1967 *The relaxation method for finding the common point of convex sets and its application to the solution of problems in convex programming*
USSR Comp. Math. Math. Phys., 7.

$$\frac{1}{2}\|\tilde{g} - Af\|_2^2 + \alpha\Psi(f)$$

Monoton decrease of the residual: $\|\tilde{g} - Af_\alpha^{k+1}\|_2 \leqslant \|\tilde{g} - Af_\alpha^k\|_2$.

If there is a $f^\dagger \in \operatorname{argmin}\|\tilde{g} - Af\|_2^2$ with $\Psi(f) < \infty$, we have that

$$D_\Psi^{p^{k+1}}(f^\dagger, f_\alpha^{k+1}) \leqslant D_\Psi^{p^k}(f^\dagger, f_\alpha^k)$$

For denoising, f_α^k converges to $\tilde{g}$.

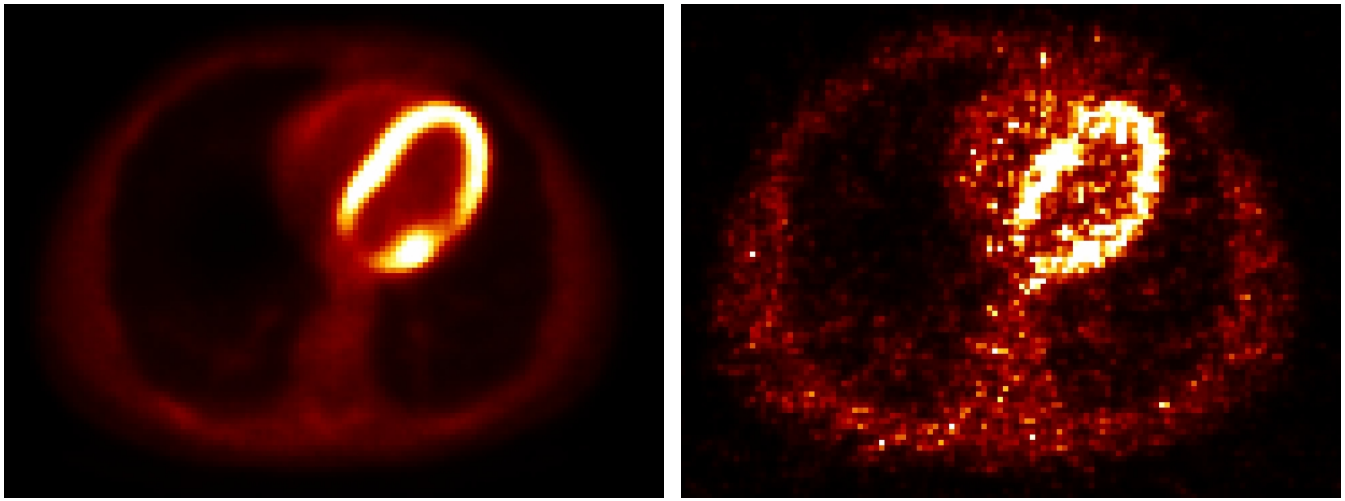
Assume that $Af^\dagger = Af_{true}$, $\|\tilde{g} - Af^\dagger\|_2^2 \leqslant \varepsilon^2$ and let f_α^k be the Bregman iteration with data $\tilde{g}$. As long as $\|\tilde{g} - Af_\alpha^k\|_2^2 > \varepsilon^2$ we have that

$$D_\Psi^{p^{k+1}}(f^\dagger, f_\alpha^{k+1}) \leqslant D_\Psi^{p^k}(f^\dagger, f_\alpha^k)$$

Semi-convergence to the real solution.

$\Longrightarrow$ A stopping criterion based on the discrepancy is reasonable.

The Bregman iteration in fast PET



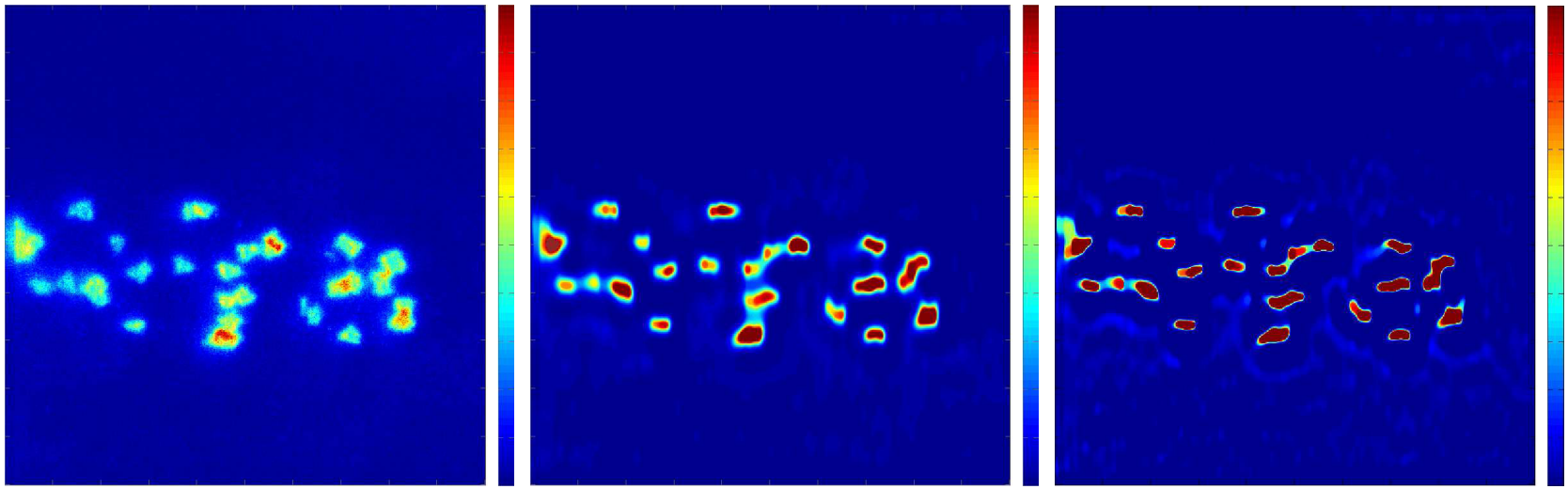
(a) EM, 20 minutes

(b) EM, 5 sec

(c) EM-TV, 5 sec

(d) Bregman EM-TV, 5 sec

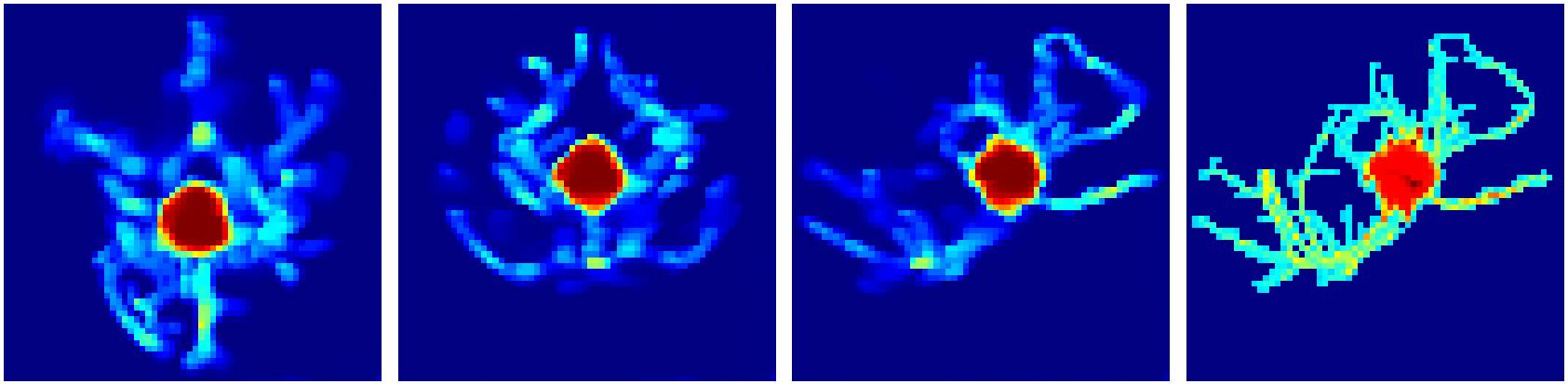
from: Jahn Müller, 2013. "Advanced Image Reconstruction and Denoising - Bregmanized (Higher Order) Total Variation and Application in PET", *PhD thesis*.



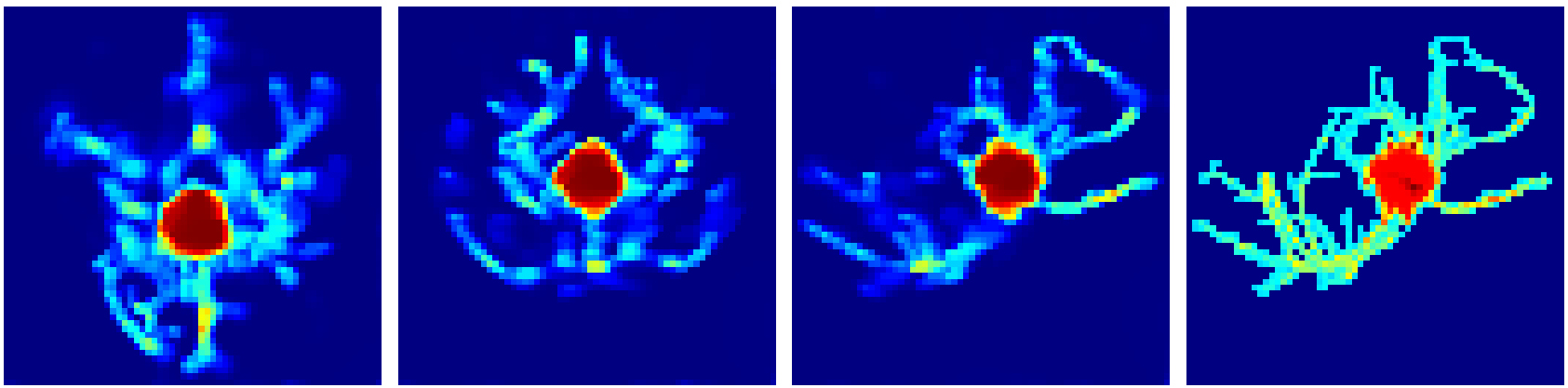
(e) Data (f) EM-TV (g) Bregman EM-TV

Protein Bruchpilot in active zones of neuromuscular synapses in larval Drosophila.

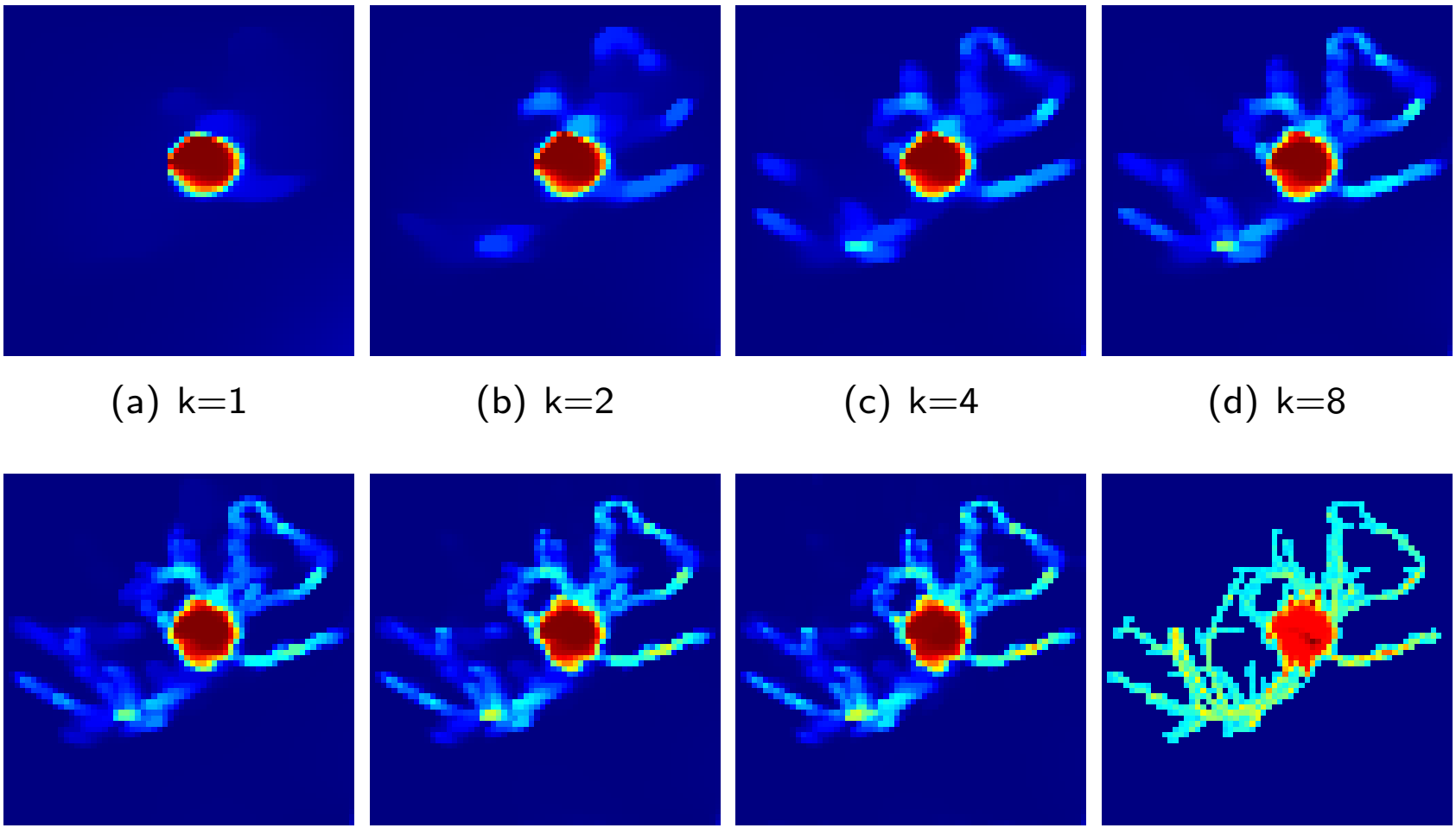
From: C. Brune, A. Sawatzky M. Burger, 2011. "Primal and Dual Bregman Methods with Application to Optical Nanoscopy", *Int. J. Comput. Vis.*, 92.



(a) TV, X (b) TV, Y (c) TV, Z (d) Phantom, Z



(e) TV-Bregman, X (f) TV-Bregman, Y (g) TV-Bregman, Z (h) Phantom, Z



(a) k=1 (b) k=2 (c) k=4 (d) k=8

(e) k=16 (f) k=3 (g) k=50 (h) Phantom, Z

General references and further reading


M. Burger and S. Osher, 2013. *A Guide to the TV Zoo*
in: *Level Set and PDE Based Reconstruction Methods in Imaging*, Lecture Notes in Mathematic. Springer International Publishing.


C.R. Vogel, 2002. *Computational Methods for Inverse Problems*
SIAM, Philadelphia, PA, USA.


S. Boyd and L. Vandenberghe, 2004 *Convex Optimization*
Cambridge University Press, New York, USA.


S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, 2011
Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers
Foundations and Trends in Machine Learning, 3(1).


M. Benning, C. Brune, M. Burger, J. Müller, 2013 *Higher-Order TV Methods - Enhancement via Bregman Iteration*
Journal of Scientific Computing, 54(2-3).