Numerical Methods for Variational Image Processing

Suh-Yuh Yang (楊肅煜)

Department of Mathematics, National Central University
Jhongli District, Taoyuan City 320317, Taiwan

First version: November 13, 2022/Last updated: May 14, 2024

Outline

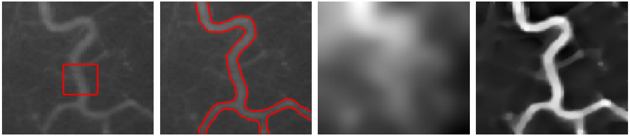
In these two lectures, I will briefly introduce

1. *Variational method for image denoising*
   - The Rudin-Osher-Fatemi total-variation model
   - Calculus of variations and the Euler-Lagrange equation
   - Implementation: a finite difference method
2. *Variational method for image segmentation*
   - The Mumford-Shah model and the Chan-Vese model
   - Implementation: the level set method + a finite difference method

What are these topics doing?

1. *Variational image denoising*

2. *Variational image segmentation*



The content of this lecture is mainly based on

- P. Getreuer, Rudin-Osher-Fatemi total variation denoising using split Bregman, *Image Processing On Line,* 2 (2012), pp. 74-95.
- L. I. Rudin, S. Osher, and E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D,* 60 (1992), pp. 259-268.

Total variation (TV)

Let $\Omega := (a, b) \subset \mathbb{R}$ be an open bounded interval. Let $\mathcal{P}_n = \{x_0, x_1, \cdots, x_n\}$, with $x_0 = a$ and $x_n = b$, be an arbitrary partition of $\overline{\Omega} = [a, b]$ and $\Delta x_i = x_i - x_{i-1}$, for $i = 1, 2, \cdots, n$. The total variation of a real-valued function $u : \overline{\Omega} \to \mathbb{R}$ is defined as the quantity,

$$\|u\|_{TV(\Omega)} := \sup_{\mathcal{P}_n} \sum_{i=1}^{n} |u(x_i) - u(x_{i-1})|.$$

If $\|u\|_{TV(\Omega)} < \infty$, then we say u is a function of bounded variation.

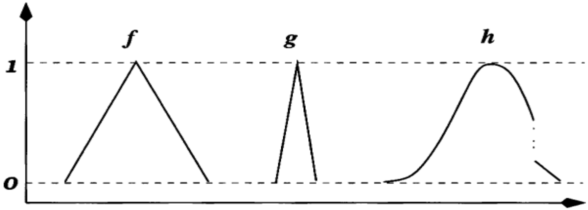
Remarks:

- If u is a smooth function, then we have

$$\|u\|_{TV(\Omega)} = \sup_{\mathcal{P}_n} \sum_{i=1}^{n} \left| \frac{u(x_i) - u(x_{i-1})}{\Delta x_i} \right| \Delta x_i = \int_{\Omega} |u'(x)|\, dx.$$

- $\|u\|_{TV(\Omega)} = 0$ does not imply $u \equiv 0$; any constant function u has $\|u\|_{TV(\Omega)} = 0 \implies \|u\|_{TV(\Omega)}$ is not a norm on any vector space.

Examples of bounded variation functions



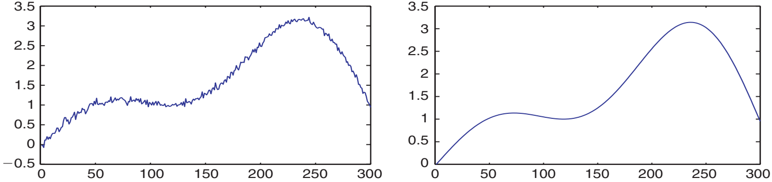
All these three functions f, g and h have total variation 2

Denoising

Total variation of $u = \|u\|_{TV(\Omega)} = \int_{\Omega} |u'(x)|\, dx$ if u is smooth.

Denoising is the problem of removing noise from an image.

minimizes $\left(\int_{\Omega} |u'(x)|\, dx + \text{ some data fidelity term} \right) \Longrightarrow$ *denoising!*



A noisy signal and its denoising result

The ROF total-variation model

Let $f : \overline{\Omega} \subset \mathbb{R}^2 \to \mathbb{R}$ be a given noisy image. Rudin, Osher, and Fatemi (*Physica D, 1992*) proposed the model for image denoising:

$$\min_{u \in BV(\Omega) \cap L^2(\Omega)} \Big(\underbrace{\|u\|_{TV(\Omega)}}_{regularizer} + \underbrace{\frac{\lambda}{2} \int_\Omega \big(u(\boldsymbol{x}) - f(\boldsymbol{x})\big)^2 \, d\boldsymbol{x}}_{data\ fidelity} \Big),$$

where $\lambda > 0$ is a tuning parameter which controls the regularization strength. Notice that

- A smaller value of λ will lead to a more regular solution.
- The space of functions with bounded variation help remove spurious oscillations (noise) and preserve sharp signals (edges).
- The *TV* term allows the solution to have discontinuities.

The bounded variation space $BV(\Omega)$

Let Ω be an open subset of $\mathbb{R}^2$. The space of functions of bounded variation $BV(\Omega)$ is defined as the space of real-valued function $u \in L^1(\Omega)$ such that the total variation is finite, i.e.,

$$BV(\Omega) = \{u \in L^1(\Omega) : \|u\|_{TV(\Omega)} < \infty\},$$

where

- $\|u\|_{TV(\Omega)} = \sup \left\{ \int_\Omega u \nabla \cdot \varphi d\boldsymbol{x} : \varphi \in C_c^1(\Omega, \mathbb{R}^2), \|\varphi\|_{(L^\infty(\Omega))^2} \leq 1 \right\}$
- $C_c^1(\Omega, \mathbb{R}^2)$ is the space of continuously differentiable vector functions with compact support in Ω.
- $L^1(\Omega)$ and $L^\infty(\Omega)$ are the usual $L^p(\Omega)$ space for $p = 1$ and $p = \infty$, respectively, equipped with the $\|\cdot\|_{L^p(\Omega)}$ norm.
- *Then $BV(\Omega)$ is a Banach space with the norm,*

$$\|u\|_{BV(\Omega)} := \|u\|_{L^1(\Omega)} + \|u\|_{TV(\Omega)}.$$

The existence, uniqueness and stability of solution

Theorem: Consider the ROF total-variation model. Then we have

(1) *If u is smooth, then $\|u\|_{TV(\Omega)} = \int_\Omega |\nabla u| \, d\boldsymbol{x}$.*

(2) *If $f \in L^2(\Omega)$, then the minimizer exists and is unique and is stable in L^2 with respect to perturbations in f.*

ROF model for image denoising: Below we assume that u is smooth, and we consider the model

$$\min_{u \in \mathcal{V}} \left(\int_\Omega |\nabla u| \, d\boldsymbol{x} + \frac{\lambda}{2} \int_\Omega \big(u(\boldsymbol{x}) - f(\boldsymbol{x})\big)^2 \, d\boldsymbol{x} \right).$$

Let $E[\cdot]$ be the energy functional over the vector space $\mathcal{V}$,

$$E[u] := \int_\Omega |\nabla u| \, d\boldsymbol{x} + \frac{\lambda}{2} \int_\Omega \big(u(\boldsymbol{x}) - f(\boldsymbol{x})\big)^2 \, d\boldsymbol{x}.$$

Calculus of variations

Let $\Omega \subset \mathbb{R}^2$ be an open bounded domain. We consider the following real-valued energy functional,

$$E[v] := \int_{\Omega} L\big(x, y, v(x,y), v_x(x,y), v_y(x,y)\big)\, d\boldsymbol{x},$$

where we assume that $v \in C^2(\overline{\Omega})$ and $L \in C^2$ with respect to its arguments $\boldsymbol{x} = (x, y)$, v, v_x and v_y.

- If $E[v]$ attains a local minimum (or maximum) at u and $\eta(x,y)$ is a smooth function on $\overline{\Omega}$, then for ε close to 0, we have

$$E[u] \leq E[u + \varepsilon\eta]. \qquad (\text{or } E[u] \geq E[u + \varepsilon\eta])$$

- Define $\Phi(\varepsilon) := E[u + \varepsilon\eta]$ in the variable ε. Then we have

$$\Phi'(0) = \left.\frac{d\Phi}{d\varepsilon}\right|_{\varepsilon=0} = \int_{\Omega} \left.\frac{dL}{d\varepsilon}\right|_{\varepsilon=0} d\boldsymbol{x} = 0. \quad (\textit{just a necessary condition})$$

The total derivative of L

Taking the total derivative of $L(x, y, v, v_x, v_y)$, where $v = u + \varepsilon\eta$ $v_x = u_x + \varepsilon\eta_x$ and $v_y = u_y + \varepsilon\eta_y$, we have

$$\frac{dL}{d\varepsilon} = \frac{\partial L}{\partial v}\eta + \frac{\partial L}{\partial v_x}\eta_x + \frac{\partial L}{\partial v_y}\eta_y = \frac{\partial L}{\partial v}\eta + \Big(\frac{\partial L}{\partial v_x}, \frac{\partial L}{\partial v_y}\Big)^\top \cdot \nabla\eta.$$

By the integration by parts, we obtain

$$\begin{aligned} 0 &= \int_\Omega \frac{dL}{d\varepsilon}\Big|_{\varepsilon=0} d\boldsymbol{x} = \int_\Omega \frac{\partial L}{\partial u}\eta + \big(\frac{\partial L}{\partial u_x}, \frac{\partial L}{\partial u_y}\big)^\top \cdot \nabla\eta \, d\boldsymbol{x} \qquad \swarrow (\star) \\ &= \int_\Omega \frac{\partial L}{\partial u}\eta \, d\boldsymbol{x} + \int_{\partial\Omega} \Big(\big(\frac{\partial L}{\partial u_x}, \frac{\partial L}{\partial u_y}\big)^\top \cdot \boldsymbol{n}\Big)\eta \, d\boldsymbol{\sigma} - \int_\Omega \Big(\nabla \cdot \big(\frac{\partial L}{\partial u_x}, \frac{\partial L}{\partial u_y}\big)^\top\Big)\eta \, d\boldsymbol{x}, \end{aligned}$$

where $L(x, y, v, v_x, v_y) = L(x, y, u, u_x, u_y)$ when $\varepsilon = 0$. Taking arbitrary smooth functions η's with $\eta(\boldsymbol{x}) = 0$ on $\partial\Omega$, we have

$$\int_\Omega \eta\Big(\frac{\partial L}{\partial u} - \nabla \cdot \big(\frac{\partial L}{\partial u_x}, \frac{\partial L}{\partial u_y}\big)^\top\Big) \, d\boldsymbol{x} = 0.$$

The Euler-Lagrange equation

- *According to the fundamental lemma of calculus of variations, we have the Euler-Lagrange equation,*

$$\frac{\partial L}{\partial u} - \nabla \cdot \left(\frac{\partial L}{\partial u_x}, \frac{\partial L}{\partial u_y}\right)^\top = 0 \quad \text{in } \Omega, \qquad \leftarrow (\star\star)$$

  and

$$\frac{\delta E}{\delta u} := \frac{\partial L}{\partial u} - \nabla \cdot \left(\frac{\partial L}{\partial u_x}, \frac{\partial L}{\partial u_y}\right)^\top$$

  is called the functional derivative of $E[u]$.

- By substituting $(\star\star)$ into $(\star)$, we have

$$\int_{\partial\Omega} \eta \left(\frac{\partial L}{\partial u_x} n_1 + \frac{\partial L}{\partial u_y} n_2\right) d\sigma = 0,$$

  for any smooth function η on $\overline{\Omega}$, which implies the homogeneous Neumann boundary condition,

$$\frac{\partial L}{\partial u_x} n_1 + \frac{\partial L}{\partial u_y} n_2 = 0 \quad \text{on } \partial\Omega.$$

Euler-Lagrange equation of the ROF model

Consider the energy minimization problem (ROF model):

$$\min_{u \in \mathcal{V}} \left(\int_\Omega |\nabla u| \, d\boldsymbol{x} + \frac{\lambda}{2} \int_\Omega \left(u(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2 d\boldsymbol{x} \right),$$

where $\mathcal{V}$ is a suitable space and $\lambda > 0$ is the regularization parameter. Since $\int_\Omega |\nabla u| \, d\boldsymbol{x} = \int_\Omega \sqrt{u_x^2 + u_y^2} \, d\boldsymbol{x}$, we have

$$L(x, y, u, u_x, u_y) = \sqrt{u_x^2 + u_y^2} + \frac{\lambda}{2}(u - f)^2,$$

which leads to the Euler-Lagrange equation with the Neumann BC,

$$-\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \lambda u = \lambda f \quad \text{in } \Omega, \qquad \frac{\partial u}{\partial \boldsymbol{n}} = 0 \quad \text{on } \partial\Omega.$$

The homogeneous Neumann boundary condition comes from

$$0 = \frac{\partial L}{\partial u_x} n_1 + \frac{\partial L}{\partial u_y} n_2 = \left(\frac{\partial L}{\partial u_x}, \frac{\partial L}{\partial u_y}\right) \cdot \boldsymbol{n} = \left(\frac{\nabla u}{|\nabla u|} \right) \cdot \boldsymbol{n} = \frac{1}{|\nabla u|} \frac{\partial u}{\partial \boldsymbol{n}} \quad \text{on } \partial\Omega.$$

If $|\nabla u| = 0 \;\Rightarrow\; \nabla u = \mathbf{0} \;\Rightarrow\; \frac{\partial u}{\partial \boldsymbol{n}} = 0$. Otherwise, we still have $\frac{\partial u}{\partial \boldsymbol{n}} = 0$.

Nonlinear PDE-based denoising algorithm

The boundary value problem of the ROF model is given by

$$\begin{aligned} -\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \lambda u &= \lambda f \quad \text{in } \Omega, \\ \frac{\partial u}{\partial \boldsymbol{n}} &= 0 \quad \text{on } \partial\Omega. \end{aligned}$$

Therefore, the minimizer can be obtained numerically by evolving a *finite difference approximation* of the parabolic partial differential equation with the homogeneous Neumann boundary condition:

$$\overbrace{\frac{\partial u}{\partial t} - \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \lambda u = \lambda f}^{\text{Heat}-\text{type equation}} \quad \text{for } (t, \boldsymbol{x}) \in (0, T) \times \Omega,$$

$$u(0, \boldsymbol{x}) = f(\boldsymbol{x}) \text{ for } \boldsymbol{x} \in \overline{\Omega}, \qquad \text{(initial condition)}$$

$$\nabla u \cdot \boldsymbol{n} = 0 \quad \text{for } t \in [0, T] \text{ and } \boldsymbol{x} \in \partial\Omega. \qquad \text{(boundary condition)}$$

Numerical differentiation: 1-D

Let $v : [a, b] \to \mathbb{R}$ and let $a = x_0 < x_1 < \cdots < x_N = b$ be a uniform partition of $[a, b]$ with grid size $h = (b-a)/N > 0$.

- **Forward difference for $v'(x_i)$:** Assume that $v \in C^2[a, b]$. Then for $i = 1, 2, \cdots, N-1$, by Taylor's theorem, we have

  $v(x_i + h) = v(x_i) + v'(x_i)h + \frac{1}{2}v''(\xi_i)h^2$ for some $\xi_i \in (x_i, x_i + h)$.

  $\therefore v'(x_i) = \frac{1}{h}\left(v(x_i + h) - v(x_i)\right) - \frac{1}{2}v''(\xi_i)h$

  $\therefore v'(x_i) \approx \frac{1}{h}\left(v(x_{i+1}) - v(x_i)\right)$, it is a first-order approximation!

- **Backward difference for $v'(x_i)$:** Assume that $v \in C^2[a, b]$. Then for $i = 1, 2, \cdots, N-1$, by Taylor's theorem, we have

  $v(x_i - h) = v(x_i) - v'(x_i)h + \frac{1}{2}v''(\xi_i)h^2$ for some $\xi_i \in (x_i - h, x_i)$.

  $\therefore v'(x_i) = \frac{1}{h}\left(v(x_i) - v(x_i - h)\right) + \frac{1}{2}v''(\xi_i)h$

  $\therefore v'(x_i) \approx \frac{1}{h}\left(v(x_i) - v(x_{i-1})\right)$, it is a first-order approximation!

Numerical differentiation (cont'd)

- **Central difference for $v'(x_i)$:** Assume that $v \in C^3[a,b]$. Then for $i = 1, 2, \cdots, N-1$, by Taylor's theorem, we have

$$v(x_i + h) = v(x_i) + v'(x_i)h + \frac{1}{2}v''(x_i)h^2 + \frac{1}{6}v^{(3)}(\xi_{i1})h^3,$$
$$v(x_i - h) = v(x_i) - v'(x_i)h + \frac{1}{2}v''(x_i)h^2 - \frac{1}{6}v^{(3)}(\xi_{i2})h^3,$$

for some $\xi_{i1} \in (x_i, x_i + h)$ and $\xi_{i2} \in (x_i - h, x_i)$. Subtracting the second equation from the first equation, we have

$$v(x_i + h) - v(x_i - h) = 2v'(x_i)h + \frac{1}{6}h^3\left(v^{(3)}(\xi_{i1}) + v^{(3)}(\xi_{i2})\right).$$

$$\therefore\ v'(x_i) = \frac{1}{2h}\left(v(x_i + h) - v(x_i - h)\right) - \frac{1}{6}h^2\frac{1}{2}\left(v^{(3)}(\xi_{i1}) + v^{(3)}(\xi_{i2})\right)$$

$\because\ \frac{1}{2}\left(v^{(3)}(\xi_{i1}) + v^{(3)}(\xi_{i2})\right)$ is between $v^{(3)}(\xi_{i1})$ & $v^{(3)}(\xi_{i2})$

$\therefore$ By the intermediate value theorem, $\exists\ \xi_i \in (x_i - h, x_i + h)$ s.t.

$$v^{(3)}(\xi_i) = \frac{1}{2}\left(v^{(3)}(\xi_{i1}) + v^{(3)}(\xi_{i2})\right)$$

$$\therefore\ v'(x_i) = \frac{1}{2h}\left(v(x_i + h) - v(x_i - h)\right) - \frac{1}{6}h^2 v^{(3)}(\xi_i)$$

$\therefore\ v'(x_i) \approx \frac{1}{2h}\left(v(x_{i+1}) - v(x_{i-1})\right)$, 2nd-order approximation!

Numerical differentiation (cont'd)

• **Central difference for $v''(x_i)$:** Assume that $v \in C^4[a,b]$. Then for $i = 1, 2, \cdots, N-1$, by Taylor's theorem, we have

$$v(x_i + h) = v(x_i) + v'(x_i)h + \frac{1}{2}v''(x_i)h^2 + \frac{1}{6}v^{(3)}(x_i)h^3 + \frac{1}{24}v^{(4)}(\xi_{i1})h^4,$$
$$v(x_i - h) = v(x_i) - v'(x_i)h + \frac{1}{2}v''(x_i)h^2 - \frac{1}{6}v^{(3)}(x_i)h^3 + \frac{1}{24}v^{(4)}(\xi_{i2})h^4,$$

for some $\xi_{i1} \in (x_i, x_i + h)$ and $\xi_{i2} \in (x_i - h, x_i)$. Therefore, we have

$$v(x_i + h) + v(x_i - h) = 2v(x_i) + v''(x_i)h^2 + \frac{1}{24}\{v^{(4)}(\xi_{i1}) + v^{(4)}(\xi_{i2})\}h^4.$$

$\therefore$

$$v''(x_i) = \frac{1}{h^2}\{v(x_i + h) - 2v(x_i) + v(x_i - h)\} - \frac{h^2}{24}\{v^{(4)}(\xi_{i1}) + v^{(4)}(\xi_{i2})\}$$

$\because v \in C^4[a,b]$, $\frac{1}{2}\{v^{(4)}(\xi_{i1}) + v^{(4)}(\xi_{i2})\}$ between $v^{(4)}(\xi_{i1})$ & $v^{(4)}(\xi_{i2})$

$\therefore$ By IVT, $\exists\ \xi_i$ between ξ_{i1} and ξ_{i2} ($\Rightarrow \xi_i \in (x_i - h, x_i + h)$) such that

$$v^{(4)}(\xi_i) = \frac{1}{2}\{v^{(4)}(\xi_{i1}) + v^{(4)}(\xi_{i2})\}$$

$$\therefore v''(x_i) = \frac{1}{h^2}\{v(x_i + h) - 2v(x_i) + v(x_i - h)\} - \frac{1}{12}h^2 v^{(4)}(\xi_i)$$

$\therefore v''(x_i) \approx \frac{1}{h^2}\{v(x_{i+1}) - 2v(x_i) + v(x_{i-1})\}$, 2nd-order approximation!

Let $u^n_{i,j}$ denote an approximation to $u(t_n, x_i, y_j)$

- $\dfrac{\partial u}{\partial x}(t_n, x_i, y_j) \approx \nabla_x^+ u^n_{i,j} := \dfrac{u^n_{i+1,j} - u^n_{i,j}}{h}$ (forward difference in x)
- $\dfrac{\partial u}{\partial x}(t_n, x_i, y_j) \approx \nabla_x^- u^n_{i,j} := \dfrac{u^n_{i,j} - u^n_{i-1,j}}{h}$ (backward difference in x)
- $\dfrac{\partial u}{\partial x}(t_n, x_i, y_j) \approx \nabla_x u^n_{i,j} := \dfrac{u^n_{i+1,j} - u^n_{i-1,j}}{2h} = \dfrac{1}{2}\left(\nabla_x^+ u^n_{i,j} + \nabla_x^- u^n_{i,j}\right)$ (central difference in x)
- $\dfrac{\partial u}{\partial y}(t_n, x_i, y_j) \approx \nabla_y^+ u^n_{i,j} := \dfrac{u^n_{i,j+1} - u^n_{i,j}}{h}$ (forward difference in y)
- $\dfrac{\partial u}{\partial y}(t_n, x_i, y_j) \approx \nabla_y^- u^n_{i,j} := \dfrac{u^n_{i,j} - u^n_{i,j-1}}{h}$ (backward difference in y)
- $\dfrac{\partial u}{\partial y}(t_n, x_i, y_j) \approx \nabla_y u^n_{i,j} := \dfrac{u^n_{i,j+1} - u^n_{i,j-1}}{2h} = \dfrac{1}{2}\left(\nabla_y^+ u^n_{i,j} + \nabla_y^- u^n_{i,j}\right)$ (central difference in y)

Central differences for second derivative

- Central difference for second derivative in x:

$$\begin{aligned}
\nabla_x^-(\nabla_x^+ u_{i,j}^n) &= \nabla_x^-\left(\frac{u_{i+1,j}^n - u_{i,j}^n}{h}\right) = \frac{1}{h}\left(\nabla_x^- u_{i+1,j}^n - \nabla_x^- u_{i,j}^n\right) \\
&= \frac{1}{h}\left(\frac{u_{i+1,j}^n - u_{i,j}^n}{h} - \frac{u_{i,j}^n - u_{i-1,j}^n}{h}\right) \\
&= \frac{1}{h^2}\left(u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n\right) \approx \frac{\partial^2 u}{\partial x^2}(t_n, x_i, y_j).
\end{aligned}$$

- Central difference for second derivative in y:

$$\nabla_y^-(\nabla_y^+ u_{i,j}^n) = \frac{1}{h^2}\left(u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n\right) \approx \frac{\partial^2 u}{\partial y^2}(t_n, x_i, y_j).$$

- $\nabla_x^+(\nabla_x^- u_{i,j}^n) = \nabla_x^-(\nabla_x^+ u_{i,j}^n)$, will also be denoted as $\nabla_x^2 u_{i,j}^n$.
  $\nabla_y^+(\nabla_y^- u_{i,j}^n) = \nabla_y^-(\nabla_y^+ u_{i,j}^n)$, will also be denoted as $\nabla_y^2 u_{i,j}^n$.

Forward Euler in time t

We will consider a finite difference scheme for approximating the solution of the IBVP for the Euler-Lagrange equation:

$$\frac{\partial u}{\partial t} - \nabla \cdot \left(\frac{\nabla u}{|\nabla u|}\right) + \lambda u = \lambda f \quad \text{for } (t, \boldsymbol{x}) \in (0, T) \times \Omega,$$
$$u(0, \boldsymbol{x}) = f(\boldsymbol{x}) \text{ for } \boldsymbol{x} \in \overline{\Omega},$$
$$\nabla u \cdot \boldsymbol{n} = 0 \quad \text{for } t \in [0, T] \text{ and } \boldsymbol{x} \in \partial\Omega.$$

Suppose that the image domain is given by $\overline{\Omega} = [0, 1] \times [0, 1]$. Let $x_i = ih$ and $y_j = jh$, $i, j = 0, 1, \cdots, N$, with $h = 1/N$, and $t_n = n\Delta t$. Let $f_{i,j} := f(x_i, y_j)$ and $u^n_{i,j}$ be the difference approximation to $u(t_n, x_i, y_j)$.

Forward Euler in time t:

$$\begin{aligned}\frac{\partial u}{\partial t}(t_n, x_i, y_j) &= \frac{1}{\Delta t}\left(u(t_{n+1}, x_i, y_j) - u(t_n, x_i, y_j)\right) - \frac{1}{2}\frac{\partial^2 u}{\partial t^2}(\tau_i, x_i, y_j)\Delta t \\ &\approx \frac{1}{\Delta t}\left(u^{n+1}_{i,j} - u^n_{i,j}\right).\end{aligned}$$

The forward Euler finite difference scheme

The proposed explicit finite difference scheme is given by:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \lambda(f_{i,j} - u_{i,j}^{n}) + \nabla_x^- \left(\frac{\nabla_x^+ u_{i,j}^n}{\sqrt{\left(\nabla_x^+ u_{i,j}^n\right)^2 + \left(m(\nabla_y^+ u_{i,j}^n, \nabla_y^- u_{i,j}^n)\right)^2}} \right)$$
$$+ \nabla_y^- \left(\frac{\nabla_y^+ u_{i,j}^n}{\sqrt{(\nabla_y^+ u_{i,j}^n)^2 + \left(m(\nabla_x^+ u_{i,j}^n, \nabla_x^- u_{i,j}^n)\right)^2}} \right), \quad 1 \le i,j \le N-1,$$

$$u_{0,j}^n = u_{1,j}^n,\ u_{N,j}^n = u_{N-1,j}^n,\ u_{i,0}^n = u_{i,1}^n,\ u_{i,N}^n = u_{i,N-1}^n, \quad 0 \le i,j \le N.$$

where $m(a,b) = \left(\frac{\operatorname{sign} a + \operatorname{sign} b}{2}\right) \min\{|a|,|b|\}$ is the *minmod operator;* see [ROF 1992] for more details.

- *The forward Euler scheme is conditionally stable, we need $\Delta t/h^2 \le c$.*
- Numerous other algorithms have been proposed to solve the *TV* denoising minimization problem, e.g., *the split Bregman iterations.*

Rescaling the finite difference scheme

Let $\delta_x^+ u_{i,j}^n := u_{i+1,j}^n - u_{i,j}^n$, $\delta_x^- u_{i,j}^n := u_{i,j}^n - u_{i-1,j}^n$, $\delta_y^+ u_{i,j}^n := u_{i,j+1}^n - u_{i,j}^n$, $\delta_y^- u_{i,j}^n := u_{i,j}^n - u_{i,j-1}^n$. Then the proposed finite difference scheme can be rewritten as

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} = \lambda(f_{i,j} - u_{i,j}^n) + \frac{1}{h}\delta_x^-\left(\frac{\delta_x^+ u_{i,j}^n}{\sqrt{\left(\delta_x^+ u_{i,j}^n\right)^2 + \left(m(\delta_y^+ u_{i,j}^n, \delta_y^- u_{i,j}^n)\right)^2}}\right)$$
$$+\frac{1}{h}\delta_y^-\left(\frac{\delta_y^+ u_{i,j}^n}{\sqrt{(\delta_y^+ u_{i,j}^n)^2 + \left(m(\delta_x^+ u_{i,j}^n, \delta_x^- u_{i,j}^n)\right)^2}}\right), \quad 1 \le i,j \le N-1,$$

$$u_{0,j}^n = u_{1,j}^n,\ u_{N,j}^n = u_{N-1,j}^n,\ u_{i,0}^n = u_{i,1}^n,\ u_{i,N}^n = u_{i,N-1}^n, \quad 0 \le i,j \le N.$$

Let
$$A_{i,j}^n := \frac{\delta_x^+ u_{i,j}^n}{\sqrt{\left(\delta_x^+ u_{i,j}^n\right)^2 + \left(m(\delta_y^+ u_{i,j}^n, \delta_y^- u_{i,j}^n)\right)^2}},$$
$$B_{i,j}^n := \frac{\delta_y^+ u_{i,j}^n}{\sqrt{(\delta_y^+ u_{i,j}^n)^2 + \left(m(\delta_x^+ u_{i,j}^n, \delta_x^- u_{i,j}^n)\right)^2}}.$$

Rescaling the finite difference scheme (cont'd)

Then we have

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \lambda(f_{i,j} - u_{i,j}^{n}) + \tfrac{1}{h}\delta_x^- A_{i,j}^n + \tfrac{1}{h}\delta_y^- B_{i,j}^n, \quad 1 \le i,j \le N-1,$$

$$u_{0,j}^n = u_{1,j}^n,\ u_{N,j}^n = u_{N-1,j}^n,\ u_{i,0}^n = u_{i,1}^n,\ u_{i,N}^n = u_{i,N-1}^n, \quad 0 \le i,j \le N.$$

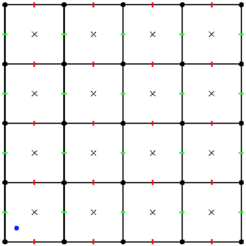
Setting $\widetilde{\Delta t} = \dfrac{\Delta t}{h}$ and $\widetilde{\lambda} = h\lambda$, the first equation becomes

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\widetilde{\Delta t}} = \widetilde{\lambda}(f_{i,j} - u_{i,j}^{n}) + \delta_x^- A_{i,j}^n + \delta_y^- B_{i,j}^n, \quad 1 \le i,j \le N-1.$$

Rearranging the equation, we finally obtain

$$u_{i,j}^{n+1} = u_{i,j}^{n} + \widetilde{\Delta t}\widetilde{\lambda}(f_{i,j} - u_{i,j}^{n}) + \widetilde{\Delta t}\delta_x^- A_{i,j}^n + \widetilde{\Delta t}\delta_y^- B_{i,j}^n, \quad 1 \le i,j \le N-1.$$

A uniform partition of $\Omega = (0,1) \times (0,1)$



Let $\bullet$ denote an arbitrary point (x, y) in $\overline{\Omega}$.

(1) In usual finite differences, the grid points (x_i, y_j) locate at $\bullet$.

(2) In image processing, however, a digital image is usually stored as a matrix. Thus, it is more convenient to use the *"cell-centered grids,"* i.e., grid points (x_i, y_j) located at $\times$ with the coordinates

$$x_i = \frac{h}{2} + (i-1)h, \;\; y_j = \frac{h}{2} + (j-1)h, \;\; i, j = (0), 1, \cdots N, (N+1).$$

And the homogeneous Neumann BC implies

$$u^n_{0,j} = u^n_{1,j}, \; u^n_{N+1,j} = u^n_{N,j}, \; u^n_{i,0} = u^n_{i,1}, \; u^n_{i,N+1} = u^n_{i,N}, \;\; 1 \le i, j \le N.$$

ROF finite difference solutions at different steps

Gaussian noise $(0, 0.005)$, $h = 1/256$, $\widetilde{\lambda} = h\lambda = 0.05$,
$\widetilde{\Delta t} = \Delta t/h = 0.01$, *at* $500, 1000, 1500, 2000$-*th steps*

Three indices to measure the quality

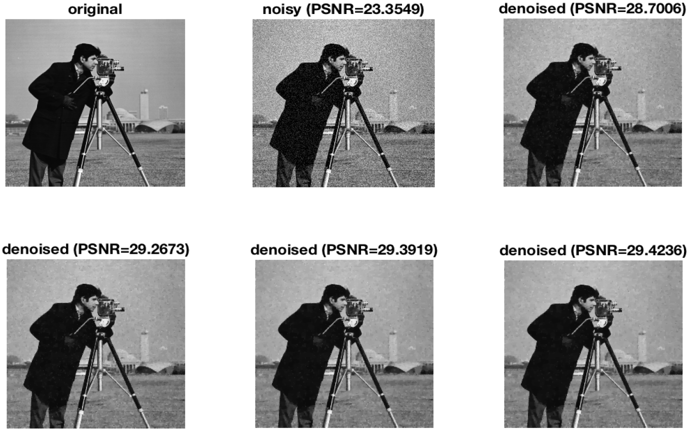
Below are three indices to measure the quality of images and to evaluate the denoising performance. Let $\widetilde{u}$ be the clean image, $\overline{u}$ be the mean intensity of the clean image, and u be the produced image.

$$
\begin{aligned}
MSE(\widetilde{u}, u) &:= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\widetilde{u}_{i,j} - u_{i,j})^2 && \text{(mean squared error)} \\
PSNR &:= 10 \log_{10} \left(\frac{255^2}{MSE(\widetilde{u}, u)} \right) && \text{(peak signal to noise ratio)} \\
SNR &:= 10 \log_{10} \left(\frac{MSE(\widetilde{u}, \overline{u})}{MSE(\widetilde{u}, u)} \right) && \text{(signal to noise ratio)}
\end{aligned}
$$

In general, the higher the value of $PSNR$ the better the quality of the produced image.

There is another index, structural similarity ($SSIM$). The maximum value of $SSIM$ is 1.

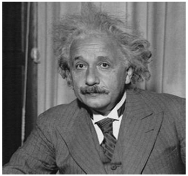
ROF finite difference solutions of different λ's (cameraman)



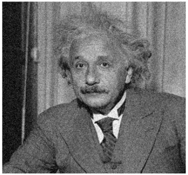
Gaussian noise $(0, 0.005)$, $h = 1/256$, $\widetilde{\lambda} = h\lambda = 1/10, 1/20, 1/30, 1/40$,
$\widetilde{\Delta t} = \Delta t/h = 0.01$, *at* 1000-*th step*

ROF finite difference solutions of different λ's (Einstein)

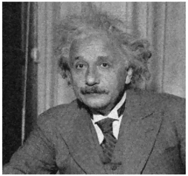
original



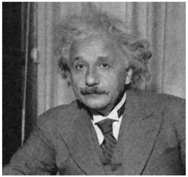
noisy (PSNR=23.0807)



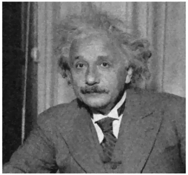
denoised (PSNR=29.4614)



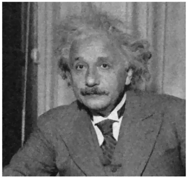
denoised (PSNR=30.5647)



denoised (PSNR=30.9154)



denoised (PSNR=31.0644)



Gaussian noise $(0, 0.005)$, $h = 1/340$, $\widetilde{\lambda} = h\lambda = 1/10, 1/20, 1/30, 1/40$,
$\widetilde{\Delta t} = \Delta t/h = 0.01$, *at* 1000-*th step*

Discretization of the ROF model using cell-centered grids

Using the cell-centered grids of $\overline{\Omega}$, we approximate the total variation term by

$$\|u\|_{TV(\Omega)} \approx h^2 \sum_{i=1}^{N} \sum_{j=1}^{N} |\nabla_h u_{i,j}|.$$

Here we define the discrete gradient operator ∇_h by

$$\nabla_h u_{i,j} := (\nabla_x u_{i,j}, \nabla_y u_{i,j})^\top$$

and recall that

$$\nabla_x u_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2h}, \qquad \nabla_y u_{i,j} = \frac{u_{i,j+1} - u_{i,j-1}}{2h}, \quad 1 \le i,j \le N,$$

$$u_{0,j} = u_{1,j},\ u_{N+1,j} = u_{N,j},\ u_{i,0} = u_{i,1},\ u_{i,N+1} = u_{i,N}, \quad 1 \le i,j \le N.$$

The constrained minimization of the ROF model

Introducing the new unknown vector function d, we have the constrained minimization problem:

$$\min_{u,\,d}\left(\int_\Omega |d|\,d\boldsymbol{x} + \frac{\lambda}{2}\int_\Omega \big(u(\boldsymbol{x}) - f(\boldsymbol{x})\big)^2\,d\boldsymbol{x}\right) \quad \text{subject to } d = \nabla u.$$

Therefore, the approximate constrained minimization of the ROF model can be posed as follows:

$$\min_{u,\,d}\left(\sum_{i,j=1}^{N} |d_{i,j}| + \frac{\lambda}{2}\sum_{i,j=1}^{N} (f_{i,j} - u_{i,j})^2\right) \quad \text{subject to } d_{i,j} = \nabla_h u_{i,j},$$

where u and d denote all $u_{i,j}$ and $d_{i,j}$. Introducing a penalty parameter $\gamma > 0$, we obtain the unconstrained minimization problem:

$$\min_{u,\,d}\left(\sum_{i,j=1}^{N} |d_{i,j}| + \frac{\lambda}{2}\sum_{i,j=1}^{N} (f_{i,j} - u_{i,j})^2 + \frac{\gamma}{2}\sum_{i,j=1}^{N} |d_{i,j} - \nabla_h u_{i,j} - b_{i,j}|^2\right),$$

where b (denotes all $b_{i,j}$) is an auxiliary variable, which can be expressed in terms of u and d, related to the Bregman iterations, and $|\cdot| := \|\cdot\|_2$ in $\mathbb{R}^2$.

An alternating direction approach: split Bregman method

Goldstein and Osher (2009) proposed to solve the above-mentioned problem by an alternating direction approach: (see Getreuer 2012)

u**-subproblem:** With d and b fixed, we solve

$$u^{k+1} = \arg\min_u \Big(\frac{\lambda}{2} \sum_{i,j} (f_{i,j} - u_{i,j})^2 + \frac{\gamma}{2} \sum_{i,j} |d_{i,j}^k - \nabla_h u_{i,j} - b_{i,j}^k|^2 \Big),$$

where the superscript k denotes the values evaluated at k-iteration. It can be viewed as the approximation of the minimization problem:

$$\min_u \frac{\lambda}{2} \int_\Omega (f-u)^2 \, d\boldsymbol{x} + \frac{\gamma}{2} \int_\Omega |d^k - \nabla u - b^k|^2 \, d\boldsymbol{x}.$$

The associated Euler-Lagrange equation of the above minimization problem (also called the screened Poisson equation) is given by

$$\lambda u - \gamma \nabla \cdot \nabla u = \lambda f - \gamma \nabla \cdot (d^k - b^k),$$

where ∇u is the gradient of u, $\nabla \cdot v$ is the divergence of vector function v, and $\Delta u := \nabla^2 u := \nabla \cdot \nabla u$ is the Laplacian of u.

The discrete screened Poisson equation

The discrete screened Poisson equation is given by

$$\lambda u_{i,j} - \gamma \nabla_h^2 u_{i,j} = \lambda f_{i,j} - \gamma \nabla_h \cdot (d_{i,j}^k - b_{i,j}^k), \quad 1 \le i,j \le N,$$

which should be supplemented with the BC:

$u_{0,j} = u_{1,j},\ u_{N+1,j} = u_{N,j},\ u_{i,0} = u_{i,1},\ u_{i,N+1} = u_{i,N},\ \ 1 \le i,j \le N.$

- The term $\Delta_h u_{i,j} := \nabla_h^2 u_{i,j} := \nabla_h^- \cdot \nabla_h^+ u_{i,j}$

$$\begin{aligned}
\nabla_h^- \cdot \nabla_h^+ u_{i,j} &= (\nabla_x^-, \nabla_y^-)^\top \cdot (\nabla_x^+ u_{i,j}, \nabla_y^+ u_{i,j})^\top \\
&= \nabla_x^- (\nabla_x^+ u_{i,j}) + \nabla_y^- (\nabla_y^+ u_{i,j}) \\
&= \frac{1}{h^2}\Big((u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) + (u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) \Big) \\
&= \frac{1}{h^2}\Big(-4u_{i,j} + u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} \Big).
\end{aligned}$$

- Let $g_{i,j}^k = (g_{1,i,j}^k, g_{2,i,j}^k)^\top := d_{i,j}^k - b_{i,j}^k$. Then

$$\nabla_h \cdot g_{i,j}^k = \nabla_x g_{1,i,j}^k + \nabla_y g_{2,i,j}^k = \frac{g_{1,i+1,j}^k - g_{1,i-1,j}^k}{2h} + \frac{g_{2,i,j+1}^k - g_{2,i,j-1}^k}{2h}.$$

The resulting linear system: $Au = r$

Finally, the resulting linear system $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{r}$ will be given by

$$\begin{aligned}(\lambda + 4\frac{\gamma}{h^2})u_{i,j} - \frac{\gamma}{h^2}\Big(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}\Big) \\ = \lambda f_{i,j} - \frac{\gamma}{2h}\Big(g^k_{1,i+1,j} - g^k_{1,i-1,j} + g^k_{2,i,j+1} - g^k_{2,i,j-1}\Big), 1 \le i,j \le N.\end{aligned}$$

- Since $\lambda > 0$ and $\gamma > 0$, $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{r}$ will be symmetric and diagonally dominant. It can be solved by many different methods such as the iterative techniques.
- For example, the Gauss-Seidel iterative method gives

$$\Big(\lambda + 4\frac{\gamma}{h^2}\Big)u^{k+1}_{i,j} = c^k_{i,j} + \frac{\gamma}{h^2}\Big(u^{k+1}_{i-1,j} + u^k_{i+1,j} + u^{k+1}_{i,j-1} + u^k_{i,j+1}\Big),\ k \ge 0,$$

where

$$c^k_{i,j} := \lambda f_{i,j} - \frac{\gamma}{2h}\Big(g^k_{1,i+1,j} - g^k_{1,i-1,j} + g^k_{2,i,j+1} - g^k_{2,i,j-1}\Big).$$

d-subproblem

d-subproblem: With u fixed, we solve

$$d^{k+1} = \arg\min_{d} \left(\sum_{i,j=1}^{N} |d_{i,j}| + \frac{\gamma}{2} \sum_{i,j=1}^{N} |d_{i,j} - \nabla_h u_{i,j}^{k+1} - b_{i,j}^{k}|^2 \right),$$

which has a closed-form solution,

$$d_{i,j}^{k+1} = \frac{\nabla_h u_{i,j}^{k+1} + b_{i,j}^{k}}{|\nabla_h u_{i,j}^{k+1} + b_{i,j}^{k}|} \max \left\{ |\nabla_h u_{i,j}^{k+1} + b_{i,j}^{k}| - \frac{1}{\gamma},\ 0 \right\}, \quad 1 \le i,j \le N.$$

How to find the closed-form solution?

The solution of d-subproblem can be found componentwisely. For each (i,j), we consider the following minimization problem:

$$\min_{\boldsymbol{x}=(x_1,x_2)^\top \in \mathbb{R}^2} \left\{ |\boldsymbol{x}| + \frac{\gamma}{2} |\boldsymbol{x} - \boldsymbol{y}|^2 \right\},$$

where $\gamma > 0$ and $\boldsymbol{y} = (y_1, y_2)^\top \in \mathbb{R}^2$ are given.

Note that $|\cdot| := \|\cdot\|_2$ in $\mathbb{R}^2$.

Updating b and selecting γ

- **Updating b:** The auxiliary variable b is initialized to zero and updated as

$$b_{i,j}^{k+1} = b_{i,j}^{k} + \nabla_h u_{i,j}^{k+1} - d_{i,j}^{k+1}, \quad 1 \le i,j \le N.$$

- **Selecting γ:** A good choice of γ is one for which both u and d subproblems converge quickly and are numerically well-conditioned.
  - In u subproblem, the effect of $\nabla \cdot \nabla$ and $\nabla \cdot$ increase when γ gets larger. It is also ill-conditioned in the limit $\gamma \to \infty$.
  - In d subproblem, the shrinking effect is more dramatic when γ is small.
  - *γ should be neither extremely large nor small for good convergence.*

In our simulations, we take $\gamma / h = 0.1$.

Implementation details of split Bregman iterations

u**-subproblem:** We multiply the following identity with h,

$$\left(\lambda + 4\frac{\gamma}{h^2}\right)u_{i,j} - \frac{\gamma}{h^2}\Big(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}\Big)$$
$$= \lambda f_{i,j} - \frac{\gamma}{2h}\Big(g^k_{1,i+1,j} - g^k_{1,i-1,j} + g^k_{2,i,j+1} - g^k_{2,i,j-1}\Big), 1 \le i,j \le N.$$

Then we have

$$\left(\lambda h + 4\frac{\gamma}{h}\right)u_{i,j} - \frac{\gamma}{h}\Big(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}\Big)$$
$$= \lambda h f_{i,j} - \frac{\gamma}{2h}\Big(hg^k_{1,i+1,j} - hg^k_{1,i-1,j} + hg^k_{2,i,j+1} - hg^k_{2,i,j-1}\Big), 1 \le i,j \le N.$$

Notice that $g^k_{i,j} = (g^k_{1,i,j}, g^k_{2,i,j})^\top := d^k_{i,j} - b^k_{i,j}$. Define $\widetilde{\lambda} = \lambda h$, $\widetilde{\gamma} = \dfrac{\gamma}{h}$, $\widetilde{g}^k_{i,j} = (\widetilde{g}^k_{1,i,j}, \widetilde{g}^k_{2,i,j})^\top := hd^k_{i,j} - hb^k_{i,j} := \widetilde{d}^k_{i,j} - \widetilde{b}^k_{i,j}$. Then we have

$$(\widetilde{\lambda} + 4\widetilde{\gamma})u_{i,j} - \widetilde{\gamma}\Big(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}\Big)$$
$$= \widetilde{\lambda} f_{i,j} - \frac{\widetilde{\gamma}}{2}\Big(\widetilde{g}^k_{1,i+1,j} - \widetilde{g}^k_{1,i-1,j} + \widetilde{g}^k_{2,i,j+1} - \widetilde{g}^k_{2,i,j-1}\Big), 1 \le i,j \le N. \quad (\star_1)$$

Implementation details of split Bregman iterations (cont'd)

***d*-subproblem:** If we define

$$\widetilde{\nabla} u_{i,j} := (\delta_x u_{i,j}, \delta_y u_{i,j})^\top := \Big(\frac{u_{i+1,j} - u_{i-1,j}}{2}, \frac{u_{i,j+1} - u_{i,j-1}}{2}\Big)^\top,$$

then since

$$d_{i,j}^{k+1} = \frac{\nabla_h u_{i,j}^{k+1} + b_{i,j}^k}{|\nabla_h u_{i,j}^{k+1} + b_{i,j}^k|} \max\left\{|\nabla_h u_{i,j}^{k+1} + b_{i,j}^k| - \frac{1}{\gamma},\ 0\right\},$$

we have

$$\begin{aligned}\widetilde{d}_{i,j}^{k+1} &= h d_{i,j}^{k+1} = \frac{h\nabla_h u_{i,j}^{k+1} + hb_{i,j}^k}{|h\nabla_h u_{i,j}^{k+1} + hb_{i,j}^k|} h \max\left\{|\nabla_h u_{i,j}^{k+1} + b_{i,j}^k| - \frac{1}{\gamma},\ 0\right\} \\ &= \frac{\widetilde{\nabla} u_{i,j}^{k+1} + \widetilde{b}_{i,j}^k}{|\widetilde{\nabla} u_{i,j}^{k+1} + \widetilde{b}_{i,j}^k|} \max\left\{|\widetilde{\nabla} u_{i,j}^{k+1} + \widetilde{b}_{i,j}^k| - \frac{1}{\widetilde{\gamma}},\ 0\right\}. \quad (\star_2)\end{aligned}$$

Implementation details of split Bregman iterations (cont'd)

Updating b: First, we have

$$b_{i,j}^{k+1} = b_{i,j}^{k} + \nabla_h u_{i,j}^{k+1} - d_{i,j}^{k+1}.$$

By multiplying the identity with h, we obtain

$$hb_{i,j}^{k+1} = hb_{i,j}^{k} + h\nabla_h u_{i,j}^{k+1} - hd_{i,j}^{k+1}.$$

In other words,

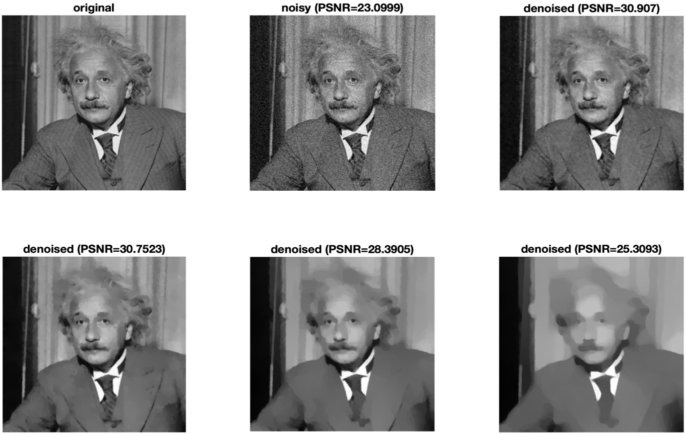
$$\widetilde{b}_{i,j}^{k+1} = \widetilde{b}_{i,j}^{k} + \widetilde{\nabla} u_{i,j}^{k+1} - \widetilde{d}_{i,j}^{k+1}. \quad (\star_3)$$

A summary

To sum up, we have the following remarks:

- By change of variables, the split Bregman iterations can be reformulated as $(\star_1)$, $(\star_2)$, $(\star_3)$, *where the grid size h can be absorbed by other variables!*
- Most engineering-oriented papers usually take the spatial grid size $h = 1$ in the finite differences. It is irrational from the approximation viewpoint because the error terms in Taylor's theorem may not be small if we take $h = 1$.
- However, if the grid size h has been absorbed by other variables as discussed above, then it is reasonable for us to say that, in some sense, the grid size $h = 1$.

Numerical experiments (Einstein)



Gaussian noise $(0, 0.005)$, $h = 1/340$, $\widetilde{\lambda} = h\lambda = 0.1, 0.05, 0.025, 0.01$,
$\widetilde{\gamma} = \gamma/h = 0.1$

A smaller value of λ implies stronger denoising. When λ is very small, the image becomes cartoon-like with sharp jumps between nearly flat regions.

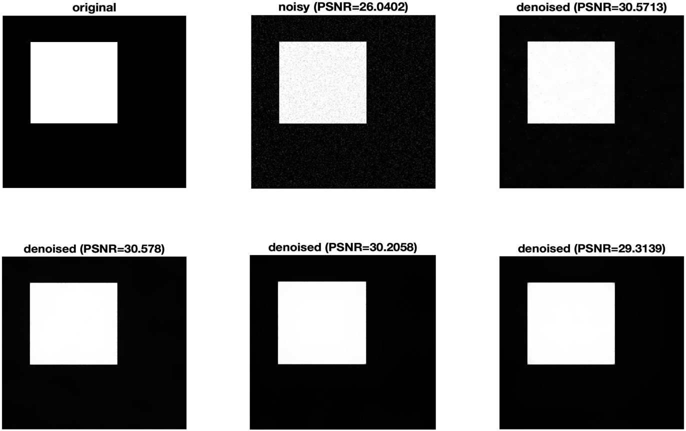
Numerical experiments (Cameraman)

Gaussian noise $(0, 0.005)$, $h = 1/256$, $\tilde{\lambda} = h\lambda = 0.1, 0.05, 0.025, 0.01$,
$\tilde{\gamma} = \gamma/h = 0.1$

Numerical experiments (Lena)

Gaussian noise $(0, 0.005)$, $h = 1/512$, $\tilde{\lambda} = h\lambda = 0.1, 0.05, 0.025, 0.01$,
$\tilde{\gamma} = \gamma/h = 0.1$

Numerical experiments (square)



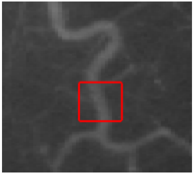
Gaussian noise $(0, 0.005)$, $h = 1/256$, $\widetilde{\lambda} = h\lambda = 0.1, 0.05, 0.025, 0.01$,
$\widetilde{\gamma} = \gamma/h = 0.1$

Lecture 2: Variational method for image segmentation

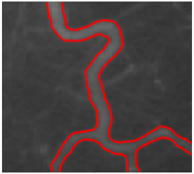
The content of this lecture is mainly based on

- T. F. Chan and L. A. Vese, An active contour model without edges, *Lecture Notes in Computer Science,* 1682 (1999), pp. 141-151.
- T. F. Chan and L. A. Vese, Active contours without edges, *IEEE Transactions on Image Processing,* 10 (2001), pp. 266-277.
- P. Getreuer, Chan-Vese segmentation, *Image Processing On Line,* 2 (2012), pp. 214-224.

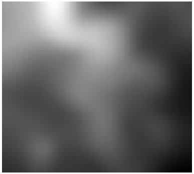
Image segmentation in medical imaging



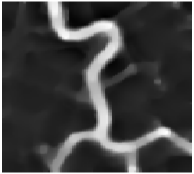
f & initialization C



segmented image



bias field b



corrected image I

Bias field model: $f = bI + n$, where n is the noise

In what follows, Ω denotes an open bounded subset in $\mathbb{R}^2$ and $f : \overline{\Omega} \to \mathbb{R}$ denotes the given grayscale image to be segmented.

Mumford-Shah model (CPAM 1989)

Mumford-Shah model: it finds a piecewise smooth function u and a curve set $\mathcal{C}$, which separates the image domain into disjoint regions, minimizing the energy functional:

$$\min_{u,\mathcal{C}} \left(\mu|\mathcal{C}| + \lambda \int_{\Omega} (f(\boldsymbol{x}) - u(\boldsymbol{x}))^2 \, d\boldsymbol{x} + \int_{\Omega \setminus \mathcal{C}} |\nabla u(\boldsymbol{x})|^2 \, d\boldsymbol{x} \right),$$

where $|\mathcal{C}|$ denotes the total length of the curves in $\mathcal{C}$.

- The first term plays the regularization role, which ensures the target objects can tightly be wrapped by $\mathcal{C}$.
- The second term is the data fidelity term, which forces u to be close to the input image f.
- The third term is the smoothing term, which forces the target function u to be piecewise smooth within each of the regions separated by the curves in $\mathcal{C}$.
- $\mu > 0$, $\lambda > 0$ *are tuning parameters to modulate these three terms.*

Simplified Mumford-Shah model

- *The non-convexity of energy functional in the Mumford-Shah model* makes the minimization problem difficult to analyze and the computational cost is much considerable.
- The piecewise smooth model suffers for its *sensitivity to the initialization of $\mathcal{C}$.*
- **Simplified Mumford-Shah model:** it finds *a piecewise constant function u* and a curve set $\mathcal{C}$ to minimize the energy functional:

$$\min_{u,\mathcal{C}} \left(\mu \left| \mathcal{C} \right| + \int_{\Omega} \left(f(\boldsymbol{x}) - u(\boldsymbol{x}) \right)^2 d\boldsymbol{x} \right).$$

Note that u is constant on each connected component of $\Omega \setminus \mathcal{C}$.
The minimization problem is still non-convex.

Chan (陳繁昌)-Vese two-phase model

In 1999, Chan and Vese proposed a two-phase segmentation model based on the level set formulation ("active contours without edges", LNCS 1999):

$$\min_{c_1,c_2,\mathcal{C}} \left(\mu|\mathcal{C}| + \nu|\Omega_{\text{in}}| + \lambda_1 \int_{\Omega_{\text{in}}} \left(f(\boldsymbol{x}) - c_1\right)^2 d\boldsymbol{x} + \lambda_2 \int_{\Omega_{\text{out}}} \left(f(\boldsymbol{x}) - c_2\right)^2 d\boldsymbol{x} \right),$$

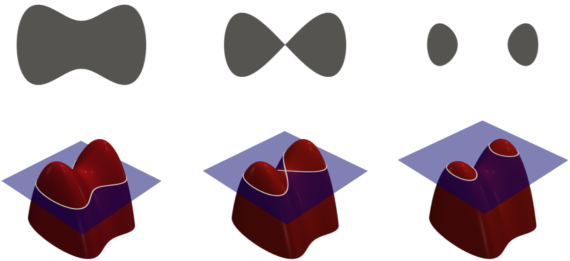
where

- Ω_{in} denotes the region enclosed by the curves in $\mathcal{C}$ with area $|\Omega_{\text{in}}|$, and $\Omega_{\text{out}} := \Omega \setminus \Omega_{\text{in}}$.
- $\mu > 0$, $\nu \geq 0$, $\lambda_1 > 0$, and $\lambda_2 > 0$ are tuning parameters (actually, one of them can be fixed as 1).
- Chan-Vese model finds a piecewise constant function u and a curve set $\mathcal{C}$ to minimize the energy functional, where u has only two constant values,

$$u(\boldsymbol{x}) = \begin{cases} c_1, & \boldsymbol{x} \text{ is inside } \mathcal{C}, \\ c_2, & \boldsymbol{x} \text{ is outside } \mathcal{C}. \end{cases}$$

Topological changes of $\mathcal{C}$

To solve the minimization problem of Chan-Vese model, we evolve $\mathcal{C}$ and find c_1, c_2 to minimize the energy functional. However, it is generally hard to handle topological changes of the curves in $\mathcal{C}$.



(quoted from wikipedia)

Level set function

Therefore, we represent $\mathcal{C}$ implicitly by the zero level contour of a level set function $\phi : \overline{\Omega} \to \mathbb{R}$, i.e.,

$$\mathcal{C} = \{\boldsymbol{x} \in \overline{\Omega} : \; \phi(\boldsymbol{x}) = 0\}.$$

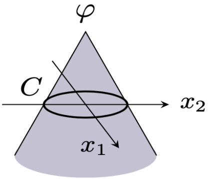
The zero level contour $\mathcal{C}$ partitions the image domain into two disjoint regions Ω_{in} and Ω_{out} such that

$$\phi(\boldsymbol{x}) \geq 0 \;\text{ for }\; \boldsymbol{x} \in \Omega_{\text{in}} \quad \text{and} \quad \phi(\boldsymbol{x}) < 0 \;\text{ for }\; \boldsymbol{x} \in \Omega_{\text{out}}.$$

For example, given $r > 0$, we define a level set function

$$\phi(\boldsymbol{x}) = \phi(x, y) = r - \sqrt{x^2 + y^2},$$

whose zero level contour is the circle of radius $r > 0$.



Chan-Vese model

- Let H denote the Heaviside function and δ the Dirac delta function. Then

$$H(s) = \begin{cases} 1 & s \geq 0, \\ 0 & s < 0, \end{cases} \qquad \text{and} \qquad \frac{d}{ds}H(s) = \delta(s).$$

- In terms of H, δ, and the level set function ϕ, the Chan-Vese model has the form

$$\begin{aligned} \min_{c1,\, c2,\, \phi} \Big(\mu \int_\Omega \delta(\phi(\boldsymbol{x})) |\nabla \phi(\boldsymbol{x})| \, d\boldsymbol{x} + \nu \int_\Omega H(\phi(\boldsymbol{x})) \, d\boldsymbol{x} \\ + \lambda_1 \int_\Omega (f(\boldsymbol{x}) - c_1)^2 H(\phi(\boldsymbol{x})) \, d\boldsymbol{x} \\ + \lambda_2 \int_\Omega (f(\boldsymbol{x}) - c_2)^2 (1 - H(\phi(\boldsymbol{x}))) \, d\boldsymbol{x} \Big). \end{aligned}$$

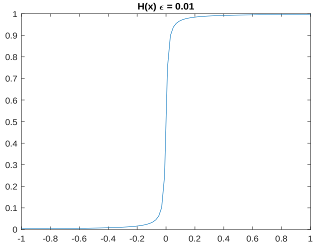
Original formulation:

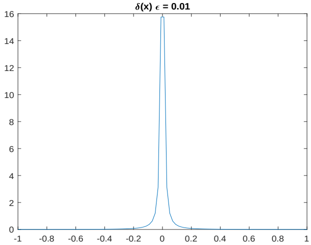
$$\min_{c_1, c_2, \mathcal{C}} \left(\mu |\mathcal{C}| + \nu |\Omega_{\text{in}}| + \lambda_1 \int_{\Omega_{\text{in}}} (f(\boldsymbol{x}) - c_1)^2 + \lambda_2 \int_{\Omega_{\text{out}}} (f(\boldsymbol{x}) - c_2)^2 \right).$$

The regularized Heaviside and delta functions

The Heaviside function H and the Dirac delta function δ can be approximately regularized as follows: for a sufficiently small $\epsilon > 0$,

$$\begin{aligned}
H_\epsilon(t) &:= \frac{1}{2}\Big(1+\frac{2}{\pi}\tan^{-1}\big(\frac{t}{\epsilon}\big)\Big), \\
\delta_\epsilon(t) &:= \frac{d}{dt}H_\epsilon(t) = \frac{\epsilon}{\pi(\epsilon^2+t^2)}, \\
\int_{-\infty}^{\infty}\delta_\epsilon(t)dt &= \int_{-\infty}^{\infty}\frac{\epsilon}{\pi(\epsilon^2+t^2)}dt = \cdots = 1.
\end{aligned}$$





Total length of $\mathcal{C}$

The first term of the energy functional is the length of $\mathcal{C}$, which can be expressed as the total variation of $H(\phi)$ and then informally

$$|\mathcal{C}| = \int_\Omega |\nabla H(\phi(\boldsymbol{x}))| = \int_\Omega \left|\frac{dH}{d\phi}(\phi(\boldsymbol{x}))\right| |\nabla \phi(\boldsymbol{x})| = \int_\Omega \delta(\phi(\boldsymbol{x}))|\nabla \phi(\boldsymbol{x})|.$$

A rough idea of the proof:

We partition Ω into very small subdomains, $\Omega = \cup_{i,j}\Omega_{i,j}$, and define $\mathcal{C}_{i,j} := \mathcal{C} \cap \Omega_{i,j}$. Then $\mathcal{C} = \cup_{i,j}\mathcal{C}_{i,j}$. We consider the approximation $H_\epsilon(\phi)$ of $H(\phi)$ for $0 < \epsilon \ll 1$. On $\Omega_{i,j}$, we have

$$\begin{aligned} |\mathcal{C}_{i,j}| &= |\mathcal{C}_{i,j}| \int_{-\infty}^{\infty} \delta_\epsilon(t)dt \approx \int_{\Omega_{i,j}} \delta_\epsilon(\phi(\boldsymbol{x}))|\nabla \phi(\boldsymbol{x})| = \int_{\Omega_{i,j}} \left|\frac{dH_\epsilon}{d\phi}(\phi(\boldsymbol{x}))\right| |\nabla \phi(\boldsymbol{x})| \\ &= \int_{\Omega_{i,j}} |\nabla H_\epsilon(\phi(\boldsymbol{x}))|. \end{aligned}$$

Taking summation over all i and j, we have

$$|\mathcal{C}| = \sum_{i,j} |\mathcal{C}_{i,j}| \approx \sum_{i,j} \int_{\Omega_{i,j}} |\nabla H_\epsilon(\phi(\boldsymbol{x}))| = \int_\Omega |\nabla H_\epsilon(\phi(\boldsymbol{x}))| \approx \int_\Omega |\nabla H(\phi(\boldsymbol{x}))|.$$

An alternating iterative scheme

The minimization is solved by *an alternating iterative scheme*, i.e., alternatingly updating c_1, c_2 and ϕ.

(S1) Fixed ϕ, the optimal values of c_1 and c_2 are the region averages,

$$c_1 = \frac{\int_\Omega f(\boldsymbol{x}) H(\phi(\boldsymbol{x}))\, d\boldsymbol{x}}{\int_\Omega H(\phi(\boldsymbol{x}))\, d\boldsymbol{x}}, \quad c_2 = \frac{\int_\Omega f(\boldsymbol{x}) \big(1 - H(\phi(\boldsymbol{x}))\big)\, d\boldsymbol{x}}{\int_\Omega \big(1 - H(\phi(\boldsymbol{x}))\big)\, d\boldsymbol{x}}.$$

(S2) Fixed c_1, c_2, we solve the initial-boundary value problem (IBVP) to reach a steady-state:

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi)\left(\mu \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} - \nu - \lambda_1 (f - c_1)^2 + \lambda_2 (f - c_2)^2\right), \quad \text{for } t > 0, \boldsymbol{x} \in \Omega,$$

$$\phi(0, \boldsymbol{x}) = \phi_0(\boldsymbol{x}), \boldsymbol{x} \in \Omega,$$

$$\frac{\partial \phi}{\partial \boldsymbol{n}} = 0 \text{ on } \partial\Omega, t \geq 0.$$

Euler-Lagrange equation

Fixed c_1 and c_2, the energy functional becomes

$$E[\phi] = \int_\Omega F(x, y, \phi, \phi_x, \phi_y)\, d\boldsymbol{x},$$

where the integrand is given by

$$\begin{aligned} F(x, y, \phi, \phi_x, \phi_y) &= \mu \delta_\epsilon(\phi)\, |\nabla \phi| + \nu H_\epsilon(\phi) + \lambda_1 (f - c_1)^2 H_\epsilon(\phi) \\ &\quad + \lambda_2 (f - c_2)^2 \big(1 - H_\epsilon(\phi)\big). \end{aligned}$$

By direct computations, we have

$$\frac{\partial F}{\partial \phi} = \mu \delta'_\epsilon(\phi)\, |\nabla \phi| + \nu \delta_\epsilon(\phi) + \lambda_1 (f - c_1)^2 \delta_\epsilon(\phi) - \lambda_2 (f - c_2)^2 \delta_\epsilon(\phi),$$

$$\frac{\partial F}{\partial \phi_x} = \mu \delta_\epsilon(\phi) \frac{\phi_x}{|\nabla \phi|}, \qquad \frac{\partial F}{\partial \phi_y} = \mu \delta_\epsilon(\phi) \frac{\phi_y}{|\nabla \phi|}.$$

The Euler-Lagrange equation with the Neumann BC are given by

$$\frac{\partial F}{\partial \phi} - \nabla \cdot \left(\frac{\partial F}{\partial \phi_x}, \frac{\partial F}{\partial \phi_y}\right)^\top = 0 \quad \text{in } \Omega, \qquad \left(\frac{\partial F}{\partial \phi_x}, \frac{\partial F}{\partial \phi_y}\right)^\top \cdot \boldsymbol{n} = 0 \quad \text{on } \partial\Omega.$$

Neumann boundary condition

It leads to the equation

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi)\left\{\mu\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \nu - \lambda_1(f-c_1)^2 + \lambda_2(f-c_2)^2\right\},$$

which has to be supplemented with an initial condition,

$$\phi(0,\boldsymbol{x}) = \phi_0(\boldsymbol{x}),\ \forall\, \boldsymbol{x}\in\Omega,$$

and the homogeneous Neumann boundary condition,

$$0 = \frac{\partial F}{\partial \phi_x} n_1 + \frac{\partial F}{\partial \phi_y} n_2 = \left(\frac{\partial F}{\partial \phi_x}, \frac{\partial F}{\partial \phi_y}\right)^\top \cdot \boldsymbol{n} = \delta_\epsilon(\phi)\frac{\nabla\phi}{|\nabla\phi|}\cdot\boldsymbol{n}.$$

That is, the BC for $t \geq 0$,

$$\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\boldsymbol{n}} = 0 \quad \text{on } \partial\Omega \quad \Longrightarrow \quad \frac{\partial\phi}{\partial\boldsymbol{n}} = 0 \quad \text{on } \partial\Omega.$$

Numerical implementation

- Assume that the image domain $\overline{\Omega}$ is the unit square $[0,1] \times [0,1]$.
- Let $\Omega_D := \{(x_i, y_j) |\ i,j = 0,1,\cdots,M\}$ be the set of grid points of a uniform partition of $\overline{\Omega}$ with size $h = 1/M$.
- Then $x_i = ih$ and $y_j = jh$, $i,j = 0,1,\cdots,M$. Let $\phi_{i,j}(t)$ be the spatial difference approximation to $\phi(t, x_i, y_j)$.
- Let $t_n = n\Delta t$, $n \geq 0$, and $\Delta t > 0$ be the time step, and let $\phi_{i,j}^n$ be the full difference approximation to $\phi(t_n, x_i, y_j)$.

Discrete differential operators and BC

- *Define the discrete differential operators: for* $1 \le i,j \le M-1$,

$$
\begin{aligned}
\nabla_x^+ \phi_{i,j} &= \frac{\phi_{i+1,j} - \phi_{i,j}}{h}, \text{ (forward difference)} \\
\nabla_x^- \phi_{i,j} &= \frac{\phi_{i,j} - \phi_{i-1,j}}{h}, \text{ (backward difference)} \\
\nabla_y^+ \phi_{i,j} &= \frac{\phi_{i,j+1} - \phi_{i,j}}{h}, \text{ (forward difference)} \\
\nabla_y^- \phi_{i,j} &= \frac{\phi_{i,j} - \phi_{i,j-1}}{h}, \text{ (backward difference)} \\
\nabla_x^0 \phi_{i,j} &:= \left(\frac{\nabla_x^+ + \nabla_x^-}{2}\right)\phi_{i,j}, \quad \nabla_y^0 \phi_{i,j} := \left(\frac{\nabla_y^+ + \nabla_y^-}{2}\right)\phi_{i,j}. \\
&\quad \text{(central differences)}
\end{aligned}
$$

- *Discretize the homogeneous Neumann BC:* $\dfrac{\partial \phi}{\partial \boldsymbol{n}} = 0$ *on* $\partial\Omega$

$$
\phi_{0,j} = \phi_{1,j}, \quad \phi_{M,j} = \phi_{M-1,j}, \quad \phi_{i,0} = \phi_{i,1}, \quad \phi_{i,M} = \phi_{i,M-1}.
$$

Finite difference discretization: spatial variables

Performing the spatial discretization [Getreuer-2012], we have

$$
\begin{aligned}
\frac{\partial \phi_{i,j}}{\partial t} = \delta_\epsilon(\phi_{i,j})\Bigg\{ & \mu\Bigg(\nabla_x^- \frac{\nabla_x^+ \phi_{i,j}}{\sqrt{\eta^2 + (\nabla_x^+ \phi_{i,j})^2 + (\nabla_y^0 \phi_{i,j})^2}} \\
& + \nabla_y^- \frac{\nabla_y^+ \phi_{i,j}}{\sqrt{\eta^2 + (\nabla_x^0 \phi_{i,j})^2 + (\nabla_y^+ \phi_{i,j})^2}}\Bigg) \\
& - \nu - \lambda_1 (f_{i,j} - c_1)^2 + \lambda_2 (f_{i,j} - c_2)^2 \Bigg\},
\end{aligned}
$$

where $i, j = 1, 2, \cdots, M-1$.

The purpose of small positive parameter η in the denominators prevents division by zero.

Spatial discretization

Define

$$
\begin{aligned}
A_{i,j} &= \frac{\mu}{\sqrt{\eta^2 + (\nabla_x^+ \phi_{i,j})^2 + (\nabla_y^0 \phi_{i,j})^2}}, \\
B_{i,j} &= \frac{\mu}{\sqrt{\eta^2 + (\nabla_x^0 \phi_{i,j})^2 + (\nabla_y^+ \phi_{i,j})^2}}.
\end{aligned}
$$

Using the fact $\nabla_x^+ \phi_{i,j} = \frac{\phi_{i+1,j} - \phi_{i,j}}{h}$, $\nabla_y^+ \phi_{i,j} = \frac{\phi_{i,j+1} - \phi_{i,j}}{h}$ and taking the backward difference at $A_{i,j}(\phi_{i+1,j} - \phi_{i,j})$ and $B_{i,j}(\phi_{i,j+1} - \phi_{i,j})$, then the discretization can be written as

$$
\begin{aligned}
\frac{\partial \phi_{i,j}}{\partial t} = \delta_\epsilon(\phi_{i,j}) \Big\{ & \frac{1}{h^2}\Big(A_{i,j}(\phi_{i+1,j} - \phi_{i,j}) - A_{i-1,j}(\phi_{i,j} - \phi_{i-1,j})\Big) \\
& + \frac{1}{h^2}\Big(B_{i,j}(\phi_{i,j+1} - \phi_{i,j}) - B_{i,j-1}(\phi_{i,j} - \phi_{i,j-1})\Big) \\
& - \nu - \lambda_1 (f_{i,j} - c_1)^2 + \lambda_2 (f_{i,j} - c_2)^2 \Big\}.
\end{aligned}
$$

Temporal discretization

Define

$$\widetilde{A}_{i,j} = \frac{1}{h^2} A_{i,j}, \quad \widetilde{A}_{i-1,j} = \frac{1}{h^2} A_{i,j},$$
$$\widetilde{B}_{i,j} = \frac{1}{h^2} B_{i,j}, \quad \widetilde{B}_{i,j-1} = \frac{1}{h^2} B_{i,j-1}.$$

Time is discretized with a semi-implicit Gauss-Seidel method, values $\phi_{i,j}, \phi_{i-1,j}, \phi_{i,j-1}$ are evaluated at time t_{n+1} and all others at time t_n.

$$\begin{aligned} \frac{\phi_{i,j}^{n+1} - \phi_{i,j}^{n}}{\Delta t} = \; & \delta_\epsilon(\phi_{i,j}^n) \Big\{ \widetilde{A}_{i,j}\phi_{i+1,j}^{n} + \widetilde{A}_{i-1,j}\phi_{i-1,j}^{n+1} + \widetilde{B}_{i,j}\phi_{i,j+1}^{n} + \widetilde{B}_{i,j-1}\phi_{i,j-1}^{n+1} \\ & - \left(\widetilde{A}_{i,j} + \widetilde{A}_{i-1,j} + \widetilde{B}_{i,j} + \widetilde{B}_{i,j-1} \right) \phi_{i,j}^{n+1} \\ & - \nu - \lambda_1 (f_{i,j} - c_1)^2 + \lambda_2 (f_{i,j} - c_2)^2 \Big\}. \end{aligned}$$

Gauss-Seidel scheme

This allows ϕ at time t_{n+1} to be solved by one Gauss-Seidel *sweep from left to right, bottom to top:*

$$
\begin{aligned}
\phi_{i,j}^{n+1} \quad = \quad & \Big\{ \phi_{i,j}^{n} + \Delta t \delta_\epsilon(\phi_{i,j}^{n}) \Big(\widetilde{A}_{i,j}\phi_{i+1,j}^{n} + \widetilde{A}_{i-1,j}\phi_{i-1,j}^{n+1} + \widetilde{B}_{i,j}\phi_{i,j+1}^{n} \\
& + \widetilde{B}_{i,j-1}\phi_{i,j-1}^{n+1} - \nu - \lambda_1 (f_{i,j} - c_1)^2 + \lambda_2 (f_{i,j} - c_2)^2 \Big) \Big\} \\
& \times \Big\{ 1 + \Delta t \delta_\epsilon(\phi_{i,j}) \Big(\widetilde{A}_{i,j} + \widetilde{A}_{i-1,j} + \widetilde{B}_{i,j} + \widetilde{B}_{i,j-1} \Big) \Big\}^{-1},
\end{aligned}
$$

where

$$
\begin{aligned}
\widetilde{A}_{i,j} \quad &= \quad \frac{\mu}{h^2 \sqrt{\eta^2 + \Big((\phi_{i+1,j}^{n} - \phi_{i,j}^{n})/h \Big)^2 + \Big((\phi_{i,j+1}^{n} - \phi_{i,j-1}^{n+1})/(2h) \Big)^2}}, \\
\widetilde{B}_{i,j} \quad &= \quad \frac{\mu}{h^2 \sqrt{\eta^2 + \Big((\phi_{i+1,j}^{n} - \phi_{i-1,j}^{n+1})/(2h) \Big)^2 + \Big((\phi_{i,j}^{n} - \phi_{i+1,j}^{n})/h \Big)^2}}.
\end{aligned}
$$

Gauss-Seidel scheme

We can rewrite $\widetilde{A}_{i,j}$ and $\widetilde{B}_{i,j}$ as follows:

$$
\begin{aligned}
\widetilde{A}_{i,j} &= \frac{\mu}{h^2\sqrt{\eta^2 + \left((\phi^n_{i+1,j} - \phi^n_{i,j})/h\right)^2 + \left((\phi^n_{i,j+1} - \phi^{n+1}_{i,j-1})/(2h)\right)^2}}, \\
&= \frac{(\mu/h)}{\sqrt{(h\eta)^2 + (\phi^n_{i+1,j} - \phi^n_{i,j})^2 + \left((\phi^n_{i,j+1} - \phi^{n+1}_{i,j-1})/2\right)^2}}, \\
\widetilde{B}_{i,j} &= \frac{\mu}{h^2\sqrt{\eta^2 + \left((\phi^n_{i+1,j} - \phi^{n+1}_{i-1,j})/(2h)\right)^2 + \left((\phi^n_{i,j} - \phi^n_{i+1,j})/h\right)^2}} \\
&= \frac{(\mu/h)}{\sqrt{(h\eta)^2 + \left((\phi^n_{i+1,j} - \phi^{n+1}_{i-1,j})/2\right)^2 + (\phi^n_{i,j} - \phi^n_{i+1,j})^2}}.
\end{aligned}
$$

In numerical implementation, we take $(h\eta) = 10^{-8}$*.*

Numerical experiments