Chapter 8

Total Variation Regularization

In section 1.3 of Chapter 1 we provided a very brief introduction to total variation regularization. In this chapter we take a closer look at both computational and theoretical issues.

8.1 Motivation

In a real analysis course [100], one sometimes sees the following definition of the total variation (TV) of a function f defined on the interval $[0, 1]$:

$$\text{TV}(f) \overset{\text{def}}{=} \sup \sum_i |f(x_i) - f(x_{i-1})|, \tag{8.1}$$

where the supremum is taken over all partitions $0 = x_0 < x_1 < \cdots < x_n = 1$ of the interval. If f is piecewise constant with a finite number of jump discontinuities, then $\text{TV}(f)$ gives the sum of magnitudes of the jumps. If f is smooth, one can multiply and divide the right-hand side of (8.1) by $\Delta x_i = x_i - x_{i-1}$ and take the limit as the $\Delta x_i \to 0$ to obtain the representation

$$\text{TV}(f) = \int_0^1 \left| \frac{df}{dx} \right| dx. \tag{8.2}$$

An obvious generalization of (8.2) to two space dimensions is

$$\text{TV}(f) = \int_0^1 \int_0^1 |\nabla f| \, dx \, dy, \tag{8.3}$$

where $\nabla f = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y})$ denotes the gradient and $|(x, y)| = \sqrt{x^2 + y^2}$ denotes the Euclidean norm. An extension of this representation, valid even when f is not smooth, is

$$\text{TV}(f) = \sup_{\vec{v} \in \mathcal{V}} \int_0^1 \int_0^1 f(x, y) \operatorname{div} \vec{v} \, dx \, dy, \tag{8.4}$$

where $\mathcal{V}$ consists of vector-valued functions $\vec{v} = (v_1(x, y), v_2(x, y))$ whose Euclidean norm is bounded by 1 and whose components v_i are continuously differentiable and vanish on the boundary of the unit square. $\operatorname{div} \vec{v} = \frac{\partial v_1}{\partial x} + \frac{\partial v_2}{\partial y}$ gives the divergence of $\vec{v}$. We will take (8.4) to be the definition of total variation; see [45] and section 8.4.

From the expression (8.4), one can develop theoretical properties of TV. For instance, one can establish that minimization of the Tikhonov–TV functional

$$(8.5) \qquad T_\alpha(f) = \frac{1}{2}\|Kf - g\|^2 + \alpha\,\mathrm{TV}(f)$$

yields a regularization scheme in the sense of section 2.2 for the operator equation $Kf = g$; see [2] and section 8.4.

TV(f) can be interpreted geometrically as the lateral surface area of the graph of f. In particular, let S be a region with a smooth boundary ∂S contained within the unit square. Take $f(x, y) = H > 0$ for (x, y) in the interior of S and $f(x, y) = 0$ in the exterior. TV(f) is then the length of the boundary ∂S multiplied by the height H of the jump in f. For example, if S is the disk of radius $1/4$ centered at $(1/2, 1/2)$, then TV$(f) = 2\pi \times 1/4 \times H$. With this geometric insight, one can begin to understand why total variation is an effective regularization functional. If f has many large amplitude oscillations, then it has large lateral surface area, and hence TV(f) is large. This is a property that TV shares with the more standard Sobolev H^1 "squared norm of the gradient" regularization functionals; see (2.47). Unlike the H^1 functional, with total variation one can effectively reconstruct functions with jump discontinuities. This is illustrated in one dimension in Figure 1.5. In two-dimensional image deblurring, total variation regularization tends to produce qualitatively correct reconstructions of blocky images [34]. By blocky, we mean the image is nearly piecewise constant with jump discontinuities, and the length of the curves on which the discontinuities occur is relatively small. The image in Figure 5.2 is blocky. The reconstruction in Figure 8.1, obtained with total variation regularization, does a much better job of preserving this blocky structure than do the reconstructions in Figure 5.3, which were generated using conventional regularization techniques.

We now turn our attention to numerical implementation.

8.2 Numerical Methods for Total Variation

We wish to obtain regularized solutions to operator equations $Kf = g$. In principle, this can be done by minimizing the Tikhonov–TV functional (8.5). However, the representations (8.2) and (8.3) are not suitable for the implementation of the numerical methods of Chapter 3, due to the nondifferentiability of the Euclidean norm at the origin. To overcome this difficulty, one can take an approximation to the Euclidean norm $|\mathbf{x}|$ like $\sqrt{|\mathbf{x}|^2 + \beta^2}$, where β is a small positive parameter. This yields the following approximation to TV(f), valid for a smooth function f defined on the unit interval in one dimension:

$$(8.6) \qquad J_\beta(f) = \int_0^1 \sqrt{\left(\frac{df}{dx}\right)^2 + \beta^2}\, dx.$$

In two space dimensions, this becomes

$$(8.7) \qquad J_\beta(f) = \int_0^1\int_0^1 \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 + \beta^2}\, dx\, dy.$$

In the following section we consider minimization of the functional

$$(8.8) \qquad T(\mathbf{f}) = \frac{1}{2}\|K\mathbf{f} - \mathbf{d}\|^2 + \alpha J(\mathbf{f}),$$

where J is a discretization of an approximation to the one-dimensional TV functional like (8.6), $\mathbf{d}$ represents discrete data, and K is a matrix; see the example in section 1.1.

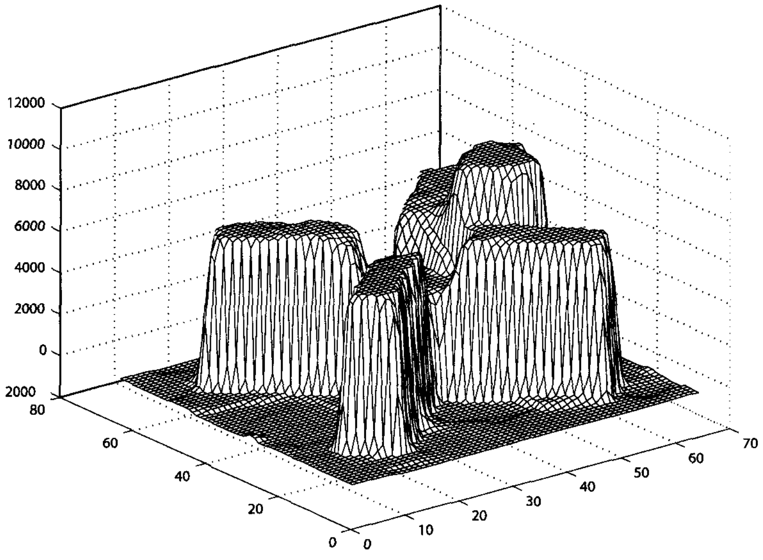

Figure 8.1. *Reconstructed image obtained with total variation regularization. The data for this reconstruction are described in section* 5.1.1. *A comparison with reconstructions in Figure* 5.3 *obtained with standard regularization techniques clearly shows that edge discontinuities and blocky structures are much better preserved with total variation.*

8.2.1 A One-Dimensional Discretization

To make the presentation less abstract, suppose $f(x)$ is a smooth function defined on the unit interval in $\mathbb{R}^1$ and $\mathbf{f} = (f_0, \ldots, f_n)$ with $f_i \approx f(x_i)$, $x_i = i\Delta x$, $\Delta x = 1/n$. Take the derivative approximation

$$D_i\mathbf{f} = (f_i - f_{i-1})/\Delta x, \qquad i = 1, \ldots, n. \tag{8.9}$$

Note the $(n+1) \times 1$ matrix representation, $D_i = [0, \ldots, 0, -1/\Delta x, 1/\Delta x, 0, \ldots, 0]$.

We assume a discretized penalty functional of the form

$$J(\mathbf{f}) = \frac{1}{2}\sum_{i=1}^{n} \psi\left((D_i\mathbf{f})^2\right) \Delta x, \tag{8.10}$$

where ψ is a smooth approximation to twice the square root function with the property

$$\psi'(t) > 0 \quad \text{whenever} \quad t > 0. \tag{8.11}$$

For example, the choice

$$\psi(t) = 2\sqrt{t + \beta^2} \tag{8.12}$$

leads to an approximation to (8.6). Another example [33] is

$$\psi(t) = \begin{cases} \frac{t}{\epsilon}, & t \le \epsilon^2, \\ 2\sqrt{t} - \epsilon, & t > \epsilon^2. \end{cases} \tag{8.13}$$

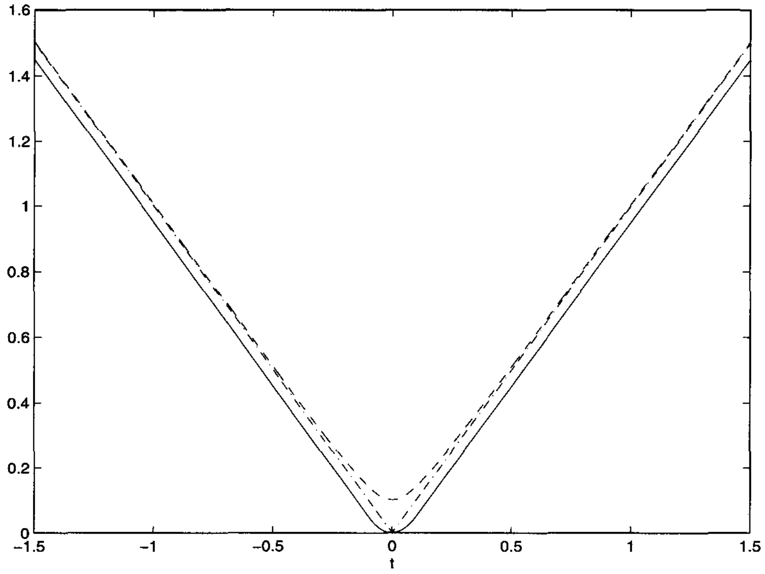

Figure 8.2. *Smooth approximations to the absolute value function. The dot-dashed curve represents the absolute value function; the solid curve represents the Huber function* $\varphi_\epsilon(t) = \psi_\epsilon(t^2)/2$ *(see* (8.13)*) with parameter* $\epsilon = 0.1$*; and the dashed curve represents the approximation* $\varphi_\beta(t) = \sqrt{t^2 + \beta^2}$ *(see* (8.12)*) with parameter* $\beta = 0.1$.

The composite function $\frac{1}{2}\psi(t^2)$ is the well-known Huber function from robust statistics. See Figure 8.2 for plots of (8.12) and (8.13).

To minimize (8.8) using the optimization techniques of Chapter 3, we need the gradient of J. For any $\mathbf{v} \in \mathbb{R}^{n+1}$,

$$\begin{aligned} &(8.14) && \frac{d}{d\tau}J(\mathbf{f} + \tau\mathbf{v}) = \sum_{i=1}^{n} \psi'\left([D_i\mathbf{f}]^2\right)\ (D_i\mathbf{f})(D_i\mathbf{v})\Delta x \\ & && = \Delta x\ (D\mathbf{v})^T\ \mathrm{diag}(\psi'(\mathbf{f}))\ (D\mathbf{f}) \\ &(8.15) && = \langle \Delta x\ D^T\ \mathrm{diag}(\psi'(\mathbf{f}))\ D\mathbf{f}, \mathbf{v}\rangle, \end{aligned}$$

where $\mathrm{diag}(\psi'(\mathbf{f}))$ denotes the $n \times n$ diagonal matrix whose ith diagonal entry is $\psi'\left((D_i\mathbf{f})^2\right)$, D is the $n \times (n+1)$ matrix whose ith row is D_i (see (8.9)), and $\langle\cdot,\cdot\rangle$ denotes the Euclidean inner product on $\mathbb{R}^{n+1}$. From this we obtain the gradient

$$(8.16) \qquad \mathrm{grad}\, J(\mathbf{f}) = L(\mathbf{f})\mathbf{f},$$

where

$$(8.17) \qquad L(\mathbf{f}) = \Delta x\ D^T\ \mathrm{diag}(\psi'(\mathbf{f}))\ D$$

is a symmetric $(n+1) \times (n+1)$ matrix. $L(\mathbf{f})$ is positive semidefinite provided condition (8.11) holds.

To obtain the Hessian of J, from (8.14),

$$\frac{\partial^2 J}{\partial\tau\partial\xi}(\mathbf{f}+\tau\mathbf{v}+\xi\mathbf{w})|_{\tau,\xi=0} = \sum_{i=1}^{n}\psi'([D_i\mathbf{f}]^2)(D_i\mathbf{w})(D_i\mathbf{v})\Delta x$$

$$+\sum_{i=1}^{n}\psi''\left([D_i\mathbf{f}]^2\right)(D_i\mathbf{f})(D_i\mathbf{v})\,2(D_i\mathbf{f})(D_i\mathbf{w})\Delta x$$

$$= \langle\Delta x\ [\mathrm{diag}(\psi'(\mathbf{f})) + \mathrm{diag}(2(D\mathbf{f})^2\psi_i''(\mathbf{f}))]\ D\mathbf{v},\ D\mathbf{w}\rangle, \tag{8.18}$$

where $\mathrm{diag}(2(D\mathbf{f})^2\psi''(\mathbf{f}))$ denotes the $n \times n$ diagonal matrix whose ith diagonal entry is $2(D_i\mathbf{f})^2\,\psi''\left([D_i\mathbf{f}]^2\right)$. Consequently,

$$\mathrm{Hess}\, J(\mathbf{f}) = L(\mathbf{f}) + L'(\mathbf{f})\mathbf{f}, \tag{8.19}$$

where $L(\mathbf{f})$ is given in (8.17) and

$$L'(\mathbf{f})\mathbf{f} = \Delta x\ D^T\ \mathrm{diag}(2(D\mathbf{f})^2\psi''(\mathbf{f}))\ D. \tag{8.20}$$

From (8.8) and (8.16)–(8.17), we obtain the gradient of the penalized least squares cost functional,

$$\mathrm{grad}\, T(\mathbf{f}) = K^T(K\mathbf{f}-\mathbf{d}) + \alpha L(\mathbf{f})\mathbf{f}. \tag{8.21}$$

From this and (8.19)–(8.20), we obtain the Hessian,

$$\mathrm{Hess}\ T(\mathbf{f}) = K^TK + \alpha L(\mathbf{f}) + \alpha L'(\mathbf{f})\mathbf{f}. \tag{8.22}$$

8.2.2 A Two-Dimensional Discretization

We now consider minimization of the penalized least squares functional (8.8), where J is a discretization of a two-dimensional total variation approximation like (8.7). The matrix K is a discretization of a linear operator which acts on functions of two variables, and the vector $\mathbf{d}$ denotes discrete data. See, for example, section 5.1.

Suppose $f = f_{ij}$ is defined on an equispaced grid in two space dimensions, $\{(x_i, y_j) \mid x_i = i\Delta x, y_j = j\Delta y,\ i = 0,\dots,n_x, j = 0,\dots,n_y\}$. In a manner analogous to the one-dimensional case, we define the discrete penalty functional $J : \mathbb{R}^{(n_x+1)\times(n_y+1)} \to \mathbb{R}$ by

$$J(f) = \frac{1}{2}\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\psi\left((D_{ij}^x f)^2 + (D_{ij}^y f)^2\right), \tag{8.23}$$

where

$$D_{ij}^x f = \frac{f_{i,j} - f_{i-1,j}}{\Delta x}, \qquad D_{ij}^y f = \frac{f_{i,j} - f_{i,j-1}}{\Delta y}. \tag{8.24}$$

To simplify notation, we dropped a factor of $\Delta x\ \Delta y$ from the right-hand side of (8.23). This factor can be absorbed in the regularization parameter α in (8.8). Gradient computations are similar to those in one dimension:

$$\frac{d}{d\tau}J(f+\tau v)|_{\tau=0} = \sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\psi_{ij}'\left[(D_{ij}^x f)(D_{ij}^x v) + (D_{ij}^y f)(D_{ij}^y v)\right], \tag{8.25}$$

where $\psi_{ij}' = \psi'((D_{ij}^x f)^2 + (D_{ij}^y f)^2)$.

Now let $\mathbf{f} = \mathbf{vec}(f)$ and $\mathbf{v} = \mathbf{vec}(v)$, corresponding to lexicographical column ordering of the two-dimensional array components (see Definition 5.25); let D_x and D_y denote the resulting $n_x n_y \times (n_x + 1)(n_y + 1)$ matrices corresponding to the grid operators in (8.24); let $\text{diag}(\psi'(\mathbf{f}))$ denote the $n_x n_y \times n_x n_y$ diagonal matrix whose diagonal entries are the ψ'_{ij}s; and let $\langle\cdot,\cdot\rangle$ denote the Euclidean inner product on $\mathbb{R}^{(n_x+1)(n_y+1)}$. Then

$$\frac{d}{d\tau} J(f + \tau v)|_{\tau=0} = \langle \text{diag}(\psi'(\mathbf{f}))\, D_x\mathbf{f}, D_x\mathbf{v}\rangle + \langle \text{diag}(\psi'(\mathbf{f}))D_y\mathbf{f}, D_y\mathbf{v}\rangle.$$

From this we obtain a gradient representation (8.16), but now

$$\begin{aligned} L(\mathbf{f}) &= D_x^T \,\text{diag}(\psi'(\mathbf{f}))\, D_x + D_y^T \,\text{diag}(\psi'(\mathbf{f}))\, D_y \\ &= [D_x^T\ D_y^T] \begin{bmatrix} \text{diag}(\psi'(\mathbf{f})) & 0 \\ 0 & \text{diag}(\psi'(\mathbf{f})) \end{bmatrix} \begin{bmatrix} D_x \\ D_y \end{bmatrix}. \end{aligned} \tag{8.26}$$

Remark 8.1. The matrix $L(\mathbf{f})$ can be viewed as a discretization of a steady-state diffusion operator

$$\begin{aligned} \mathcal{L}(f)u &= -\nabla \cdot \left(\psi' \,\nabla u\right) \\ &= -\frac{\partial}{\partial x}\left(\psi' \frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial y}\left(\psi' \frac{\partial u}{\partial y}\right) \end{aligned} \tag{8.27}$$

with the diffusion coefficient

$$\psi' = \psi'(|\nabla f|^2) = \psi'\left(\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right)$$

and with "natural" (homogeneous Neumann) boundary conditions. Expression (8.27) gives the directional derivative in the direction u of the functional

$$J(f) = \frac{1}{2}\int_0^1\int_0^1 \psi(|\nabla f|^2)\, dx\, dy. \tag{8.28}$$

An alternative approach to obtain the discrete operator $L(\mathbf{f})$ is to apply a discretization scheme directly to the continuous operator $\mathcal{L}(f)$ in (8.27). An example is the cell-centered finite difference scheme utilized in [120].

As in the one-dimensional case, one can compute a representation (8.19) for the Hessian of the penalty functional, with $L(\mathbf{f})$ given in (8.26) and

$$L'(\mathbf{f})\mathbf{f} = [D_x^T\ D_y^T] \begin{bmatrix} \text{diag}(2(D_x\mathbf{f})^2\psi''(\mathbf{f})) & \text{diag}(2(D_x\mathbf{f})(D_y\mathbf{f})\psi''(\mathbf{f})) \\ \text{diag}(2(D_y\mathbf{f})(D_x\mathbf{f})\psi''(\mathbf{f})) & \text{diag}(2(D_y\mathbf{f})^2\psi''(\mathbf{f})) \end{bmatrix} \begin{bmatrix} D_x \\ D_y \end{bmatrix}. \tag{8.29}$$

8.2.3 Steepest Descent and Newton's Method for Total Variation

In either the one- or the two-dimensional case, the gradient of the regularized cost functional has the form (8.21). To minimize (8.8), Algorithm 3.1 then gives us the following.

Algorithm 8.2.1. Steepest Descent for Total Variation–Penalized Least Squares.

```
$\nu := 0$;
$\mathbf{f}_0 :=$ initial guess;
begin steepest descent iterations
    $\mathbf{g}_\nu := K^T(K\mathbf{f}_\nu - \mathbf{d}) + \alpha\ L(\mathbf{f}_\nu)\mathbf{f}_\nu$;      % gradient
    $\tau_\nu := \arg\min_{\tau>0} T(\mathbf{f}_\nu - \tau\ \mathbf{g}_\nu)$;      % line search
    $\mathbf{f}_{\nu+1} := \mathbf{f}_\nu - \tau_\nu \mathbf{g}_\nu$;      % update approximate solution
    $\nu := \nu + 1$;
end steepest descent iterations
```

Remark 8.2. Algorithm 8.2.3 is very similar to the discretized artificial time evolution approach of Rudin, Osher, and Fatemi [101]. In principle, to obtain a regularized solution to the operator equation $Kf = g$, they computed a steady-state solution of the time-dependent diffusion equation

$$\frac{\partial f}{\partial t} = -\alpha \mathcal{L}(f) f - K^*(Kf - g),$$

where $\mathcal{L}(f)$ is given in (8.27). After spatial discretization, they used explicit time marching with a fixed time step $\tau = \Delta t$ in place of the line search parameter τ_ν. See Exercise 8.4.

In both the one- and two-dimensional cases, the Hessian of the total variation–penalized least squares functional (8.8) has form (8.22). What follows is an implementation of Newton's method (section 3.3) to minimize (8.8). Some sort of globalization is essential to guarantee convergence of the Newton iterates [119]. Here we incorporate a line search.

Algorithm 8.2.2. Newton's Method for Total Variation–Penalized Least Squares.

```
$\nu := 0$;
$f_0 :=$ initial guess;
begin primal Newton iterations
    $\mathbf{g}_\nu := K^T(K\mathbf{f}_\nu - \mathbf{d}) + \alpha\ L(\mathbf{f}_\nu)\mathbf{f}_\nu$;      % gradient
    $H_J := L(\mathbf{f}_\nu) + L'(\mathbf{f}_\nu)\mathbf{f}_\nu$;      % Hessian of penalty functional
    $H := K^T K + \alpha\ H_J$;      % Hessian of cost functional
    $\mathbf{s}_\nu := -H^{-1}\mathbf{g}_\nu$;      % Newton step
    $\tau_\nu := \arg\min_{\tau>0} T(\mathbf{f}_\nu + \tau\ \mathbf{s}_\nu)$;      % line search
    $\mathbf{f}_{\nu+1} := \mathbf{f}_\nu + \tau_\nu \mathbf{s}_\nu$;      % update approximate solution
    $\nu := \nu + 1$;
end primal Newton iterations
```

8.2.4 Lagged Diffusivity Fixed Point Iteration

An alternative to the steepest descent method and Newton's method for the minimization of (8.8) is the lagged diffusivity fixed point iteration [119]:

$$\mathbf{f}_{\nu+1} = [K^T K + \alpha\ L(\mathbf{f}_\nu)]^{-1} K^T \mathbf{d} \tag{8.30}$$

$$= \mathbf{f}_\nu - [K^T K + \alpha\ L(\mathbf{f}_\nu)]^{-1} \text{grad}\, T(\mathbf{f}_\nu). \tag{8.31}$$

The fixed point form (8.30) can be derived by first setting $\text{grad}\, T(\mathbf{f}) = \mathbf{0}$ to obtain $(K^T K + \alpha\ L(\mathbf{f}))\mathbf{f} = K^T \mathbf{d}$; see (8.21). The discretized diffusion coefficient $\psi'(\mathbf{f})$ is then evaluated at $\mathbf{f}_\nu$ to obtain $L(\mathbf{f}_\nu)$; see expressions (8.17) and (8.26) and Remark 8.1. Hence the expression "lagged diffusivity." The equivalent quasi-Newton form (8.31) can also be derived by dropping the term $\alpha\, L'(\mathbf{f})\mathbf{f}$ from the Hessian; see (8.22).

The following algorithm is based on the quasi-Newton form (8.31). The quasi-Newton form tends to be less sensitive to roundoff error than the fixed point form (8.30).

Algorithm 8.2.3. Lagged Diffusivity Fixed Point Method for Total Variation–Penalized Least Squares.

```
$\nu := 0$;
$\mathbf{f}_0$ := initial guess;
begin fixed point iterations
    $L_\nu := L(\mathbf{f}_\nu)$;              % discretized diffusion operator
    $\mathbf{g}_\nu := K^T(K\mathbf{f}_\nu - \mathbf{d}) + \alpha\, L_\nu \mathbf{f}_\nu$;   % gradient
    $H = K^T K + \alpha\, L_\nu$;         % approximate Hessian
    $\mathbf{s}_{\nu+1} := -H^{-1}\mathbf{g}_\nu$;        % quasi-Newton step
    $\mathbf{f}_{\nu+1} := \mathbf{f}_\nu + \mathbf{s}_\nu$;        % update approximate solution
    $\nu := \nu + 1$;
end fixed point iterations
```

Remark 8.3. If $K^T K$ is positive definite, one can rigorously prove that this fixed point iteration converges globally [4, 41, 13, 120, 19], so no line search is needed. The approximate Hessian differs from the true Hessian (8.22) by the term $\alpha L'(\mathbf{f}_\nu)\mathbf{f}_\nu$. This term does not typically vanish as the iteration proceeds, so the rate of convergence of the lagged diffusivity iteration should be expected to be linear.

8.2.5 A Primal-Dual Newton Method

We first recall some basic results from convex analysis. In this discussion, φ is a convex functional defined on a convex set $C \subset \mathbb{R}^d$. For our purposes, $d = 1$ or 2, $C = \mathbb{R}^d$, and

$$\varphi(\mathbf{x}) = \frac{1}{2}\psi(|\mathbf{x}|^2) \tag{8.32}$$

with $|\mathbf{x}|^2 = \mathbf{x}^T\mathbf{x} = \sum_{i=1}^d x_i^2$. Relevant examples of the function ψ include $\psi(t) = 2\sqrt{t}$, which yields $\varphi(\mathbf{x}) = |\mathbf{x}|$, and the approximations (8.12) and (8.13).

Definition 8.4. The conjugate set C^* is defined by

$$C^* = \left\{ \mathbf{y} \in \mathbb{R}^d \mid \sup_{\mathbf{x}\in C}[\mathbf{x}^T\mathbf{y} - \varphi(\mathbf{x})] < \infty \right\}, \tag{8.33}$$

and the corresponding conjugate functional to φ is

$$\varphi^*(\mathbf{y}) = \sup_{\mathbf{x}\in C}\{\mathbf{x}^T\mathbf{y} - \varphi(\mathbf{x})\}. \tag{8.34}$$

This functional, which is also known as the Fenchel transform of φ, has the conjugate set C^* as its domain.

One can show [79, Proposition 1, p. 196] that the conjugate set C^* and the conjugate functional φ^* are, respectively, a convex set and a convex functional. The corresponding second conjugates are defined in the obvious manner:

$$\varphi^{**} = (\varphi^*)^* \quad \text{and} \quad C^{**} = (C^*)^*.$$

In our finite dimensional Hilbert space setting, one can show [79, Proposition 2, p. 198] that $\varphi^{**} = \varphi$ and $C^{**} = C$. Consequently, from (8.34) we obtain the dual representation

$$\varphi(\mathbf{x}) = \sup_{y \in C^*} \{\mathbf{x}^T \mathbf{y} - \varphi^*(\mathbf{y})\}. \tag{8.35}$$

We now derive the dual representation of the Euclidean norm, $\varphi(\mathbf{x}) = |\mathbf{x}|$, on $\mathbb{R}^d$. This is used in section 8.4 to define the TV functional. By the Cauchy–Schwarz inequality,

$$\mathbf{x}^T \mathbf{y} - |\mathbf{x}| \le (|\mathbf{y}| - 1)|\mathbf{x}|, \tag{8.36}$$

with equality if and only if $\mathbf{y} = c\mathbf{x}$ for some $c \in \mathbb{R}$. If $|\mathbf{y}| > 1$, one can make (8.36) arbitrarily large by taking $\mathbf{y} = c\mathbf{x}$ and letting c increase. If $|\mathbf{y}| \le 1$, then (8.36) is zero or negative, and its maximum value of zero is attained for $\mathbf{x} = 0$. Hence

$$\sup_{\mathbf{x} \in \mathbb{R}^d} \{\mathbf{x}^T \mathbf{y} - |\mathbf{x}|\} = \begin{cases} 0, & |\mathbf{y}| \le 1, \\ +\infty, & |\mathbf{y}| \ge 1. \end{cases}$$

Thus the conjugate set is the unit ball,

$$C^* = \{y \in \mathbb{R}^d \mid |y| \le 1\},$$

and the conjugate functional $\varphi^*(\mathbf{y}) = 0$ for each $\mathbf{y} \in C^*$. The dual representation (8.35) then yields

$$|\mathbf{x}| = \sup_{|\mathbf{y}| \le 1} \mathbf{x}^T \mathbf{y}. \tag{8.37}$$

The following two examples give dual representations of convex approximations to the Euclidean norm derived from (8.12) and (8.13) via (8.32).

Example 8.5. Consider the convex functional

$$\varphi_\beta(\mathbf{x}) = \sqrt{|\mathbf{x}|^2 + \beta^2}, \qquad \beta > 0, \tag{8.38}$$

defined on $C = \mathbb{R}^d$. One can show (see Exercise 8.7) that

$$\sup_{\mathbf{x} \in \mathbb{R}^d} \{\mathbf{x}^T \mathbf{y} - \varphi_\beta(\mathbf{x})\} = \begin{cases} -\beta\sqrt{1 - |\mathbf{y}|^2} & \text{if} \quad |\mathbf{y}| \le 1, \\ +\infty & \text{if} \quad |\mathbf{y}| > 1. \end{cases} \tag{8.39}$$

Hence, the conjugate set C^* is the unit ball in $\mathbb{R}^d$, $\varphi_\beta^*(\mathbf{y}) = -\beta\sqrt{1 - |\mathbf{y}|^2}$, and by (8.35),

$$\varphi_\beta(\mathbf{x}) = \sup_{|\mathbf{y}| \le 1} \left\{\mathbf{x}^T \mathbf{y} + \beta\sqrt{1 - |\mathbf{y}|^2}\right\}. \tag{8.40}$$

Example 8.6. Consider the functional

$$\varphi_\epsilon(\mathbf{x}) = \begin{cases} \frac{|\mathbf{x}|^2}{2\epsilon} & \text{if} \quad |\mathbf{x}| \le \epsilon, \\ |\mathbf{x}| - \frac{\epsilon}{2} & \text{if} \quad |\mathbf{x}| > \epsilon, \end{cases} \tag{8.41}$$

defined on $C = \mathbb{R}^d$. The conjugate set C^* is again the unit ball, and the conjugate functional is given by

$$\varphi_\epsilon^*(\mathbf{y}) = \frac{\epsilon}{2}|\mathbf{y}|^2, \qquad \mathbf{y} \in C^*.$$

See Exercise 8.8. Consequently,

$$\varphi_\epsilon(\mathbf{x}) = \sup_{|\mathbf{y}|\leq 1} \left\{ \mathbf{x}^T\mathbf{y} - \frac{\epsilon}{2}|\mathbf{y}|^2 \right\}. \tag{8.42}$$

The following theorem relates the gradient of a convex functional φ to the gradient of its conjugate φ^*. See [30, p. 290] for a proof.

Theorem 8.7. *Suppose that φ is differentiable in a neighborhood of $\mathbf{x}_0 \in C \subset \mathbb{R}^d$, and the mapping $F = \operatorname{grad}\varphi : \mathbb{R}^d \to \mathbb{R}^d$ is invertible in that neighborhood. Then φ^* is Frechet differentiable in a neighborhood of $\mathbf{y}_0 = \varphi(\mathbf{x}_0)$ with*

$$\operatorname{grad}\varphi^*(y) = F^{-1}(\mathbf{y}). \tag{8.43}$$

We now apply convex analysis to obtain a dual formulation for the two-dimensional penalty functional (8.23). Setting

$$\varphi(x_1, x_2) = \frac{1}{2}\psi(x_1^2 + x_2^2) \tag{8.44}$$

and employing the dual representation (8.35) with $\mathbf{y} = (u, v)$, we obtain

$$J(f) = \sum_{i,j} \sup_{(u_{ij}, v_{ij})\in C^*} \{(D_{ij}^x f)u_{ij} + (D_{ij}^y f)v_{ij} - \varphi^*(u_{ij}, v_{ij})\}.$$

As in section 8.2.2, we stack the array components f_{ij}, u_{ij}, and v_{ij} into column vectors $\mathbf{f}$, $\mathbf{u}$, and $\mathbf{v}$; we let D_x and D_y be matrix representers for the grid operators D_{ij}^x and D_{ij}^y; and we let $\langle\cdot,\cdot\rangle$ denote Euclidean inner product. Then the penalty functional can be rewritten as

$$\begin{aligned} J(\mathbf{f}) &= \sup_{(\mathbf{u},\mathbf{v})\in C^*} \{\langle D_x\mathbf{f}, \mathbf{u}\rangle + \langle D_y\mathbf{f}, \mathbf{v}\rangle - \langle\varphi^*(\mathbf{u},\mathbf{v}), \mathbf{1}\rangle\} \\ &= \sup_{(\mathbf{u},\mathbf{v})\in C^*} \tilde{J}(\mathbf{u}, \mathbf{v}, \mathbf{f}), \end{aligned} \tag{8.45}$$

where

$$\tilde{J}(\mathbf{u}, \mathbf{v}, \mathbf{f}) = \langle\mathbf{f}, D_x^T\mathbf{u} + D_y^T\mathbf{v}\rangle - \langle\varphi^*(\mathbf{u},\mathbf{v}), \mathbf{1}\rangle;$$

$\mathbf{1}$ denotes the vector of 1's; and by $(\mathbf{u}, \mathbf{v}) \in C^*$ we mean each of the component pairs (u_{ij}, v_{ij}) lies in C^*.

Minimization of the penalized least squares functional (8.8) is equivalent to computing the saddle point

$$(\mathbf{u}^*, \mathbf{v}^*, \mathbf{f}^*) = \arg\min_{\mathbf{f}} \max_{(\mathbf{u},\mathbf{v})\in C^*} \tilde{T}(\mathbf{u}, \mathbf{v}, \mathbf{f}), \tag{8.46}$$

where

$$\tilde{T}(\mathbf{u}, \mathbf{v}, \mathbf{f}) = \frac{1}{2}||K\mathbf{f} - \mathbf{d}||^2 + \alpha\tilde{J}(\mathbf{u}, \mathbf{v}, \mathbf{f}).$$

We refer to $\mathbf{f}$ as the primal variable and to $\mathbf{u}$ and $\mathbf{v}$ as the dual variables.

Since (8.46) is unconstrained with respect to $\mathbf{f}$, a first order necessary condition for a saddle point is

$$0 = \text{grad}_{\mathbf{f}} \tilde{T} = K^T(K\mathbf{f} - \mathbf{d}) + \alpha(D_x^T\mathbf{u} + D_y^T\mathbf{v}). \tag{8.47}$$

An additional necessary condition is that the duality gap in (8.35) must vanish, i.e., for each grid index i, j,

$$\varphi(D_{ij}^x\mathbf{f}, D_{ij}^y\mathbf{f}) = (D_{ij}^x\mathbf{f})u_{ij} + (D_{ij}^y\mathbf{f})v_{ij} - \varphi^*(u_{ij}, v_{ij}). \tag{8.48}$$

Finally, the dual variables must lie in the conjugate set; i.e.,

$$(u_{ij}, v_{ij}) \in C^*. \tag{8.49}$$

We next examine the implications of (8.48). Suppose (8.48) holds for a point (u_{ij}, v_{ij}) in the interior of C^*. This is the case in each of Examples 8.5 and 8.6; see Exercises 8.7 and 8.8. Then

$$\begin{aligned}(0,0) &= \text{grad}_{u_{ij},v_{ij}}[(D_{ij}^x\mathbf{f})u_{ij} + (D_{ij}^y\mathbf{f})v_{ij} - \varphi^*(u_{ij}, v_{ij})] \\ &= (D_{ij}^x\mathbf{f}, D_{ij}^y\mathbf{f}) - \text{grad}\,\varphi^*(u_{ij}, v_{ij}) \\ &= (D_{ij}^x\mathbf{f}, D_{ij}^y\mathbf{f}) - \frac{1}{\psi'((D_{ij}^x\mathbf{f})^2 + (D_{ij}^y\mathbf{f})^2)}(u_{ij}, v_{ij}).\end{aligned} \tag{8.50}$$

The last equality follows from the representation (8.44) and Theorem 8.7. Equation (8.50) is equivalent to

$$D_{ij}^x f = \frac{u_{ij}}{\psi'_{ij}}, \qquad D_{ij}^y f = \frac{v_{ij}}{\psi'_{ij}}, \tag{8.51}$$

where $\psi'_{ij} = \psi'_{ij}((D_{ij}^x\mathbf{f})^2 + (D_{ij}^y\mathbf{f})^2)$. Returning to matrix notation, we have

$$D_x\mathbf{f} = B(\mathbf{f})\,\mathbf{u}, \qquad D_y\mathbf{f} = B(\mathbf{f})\,\mathbf{v}, \tag{8.52}$$

where

$$B(\mathbf{f}) = \text{diag}(1/\psi'(\mathbf{f})). \tag{8.53}$$

We can reformulate the first order necessary conditions (8.47), (8.52) as a nonlinear system

$$\mathbf{G}(\mathbf{u},\mathbf{v},\mathbf{f}) \overset{\text{def}}{=} \begin{bmatrix} B(\mathbf{f})\mathbf{u} - D_x\mathbf{f} \\ B(\mathbf{f})\mathbf{v} - D_y\mathbf{f} \\ \alpha D_x^T\mathbf{u} + \alpha D_y^T\mathbf{v} + K^T(K\mathbf{f} - \mathbf{d}) \end{bmatrix} = \mathbf{0}. \tag{8.54}$$

The derivative of G can be expressed as

$$\mathbf{G}'(\mathbf{u},\mathbf{v},\mathbf{f}) = \begin{bmatrix} B(\mathbf{f}) & \mathbf{0} & B'(\mathbf{f})\mathbf{u} - D_x \\ \mathbf{0} & B(\mathbf{f}) & B'(\mathbf{f})\mathbf{v} - D_y \\ \alpha D_x^T & \alpha D_y^T & K^TK \end{bmatrix}. \tag{8.55}$$

Here $B'(\mathbf{f})\mathbf{u}$ has component representation

$$[B'(\mathbf{f})\mathbf{u}]_{ij} = \frac{-\psi''_{ij}}{(\psi'_{ij})^2}\, 2\left((D_{ij}^x\mathbf{f})D_{ij}^x + (D_{ij}^y\mathbf{f})D_{ij}^y\right) u_{ij}.$$

In matrix form,

$$B'(\mathbf{f})\mathbf{u} - D_x = -E_{11}D_x - E_{12}D_y, \tag{8.56}$$

with

$$E_{11} = \operatorname{diag}\left(1 + \frac{2\,\psi''(\mathbf{f})\,(D_x\mathbf{f})\,\mathbf{u}}{\psi'(\mathbf{f})^2}\right), \qquad E_{12} = \operatorname{diag}\left(\frac{2\,\psi''(\mathbf{f})\,(D_y\mathbf{f})\,\mathbf{u}}{\psi'(\mathbf{f})^2}\right), \tag{8.57}$$

where the products and quotients are computed pointwise. Similarly,

$$B'(\mathbf{f})\mathbf{v} - D_y = -E_{21}D_x - E_{22}D_y, \tag{8.58}$$

with

$$E_{21} = \operatorname{diag}\left(\frac{2\,\psi''(\mathbf{f})\,(D_x\mathbf{f})\,\mathbf{v}}{\psi'(\mathbf{f})^2}\right), \qquad E_{22} = \operatorname{diag}\left(1 + \frac{2\,\psi''(\mathbf{f})\,(D_y\mathbf{f})\,\mathbf{v}}{\psi'(\mathbf{f})^2}\right). \tag{8.59}$$

Newton's method for the system (8.54) requires solutions of systems of the form $\mathbf{G}'(\mathbf{u},\mathbf{v},\mathbf{f})\ (\Delta\mathbf{u},\Delta\mathbf{v},\Delta\mathbf{f}) = -\mathbf{G}(\mathbf{u},\mathbf{v},\mathbf{f})$. Substituting (8.56)–(8.59) and applying block row reduction to convert $\mathbf{G}'$ to block upper triangular form, we obtain

$$\begin{bmatrix} B(\mathbf{f}) & \mathbf{0} & -E_{11}D_x - E_{12}D_y \\ \mathbf{0} & B(\mathbf{f}) & -E_{21}D_x - E_{22}D_y \\ \mathbf{0} & \mathbf{0} & K^TK + \alpha\overline{L} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u} \\ \Delta\mathbf{v} \\ \Delta\mathbf{f} \end{bmatrix} = \begin{bmatrix} -\mathbf{g}_1 \\ -\mathbf{g}_2 \\ \mathbf{r} \end{bmatrix}. \tag{8.60}$$

Here $\mathbf{g}_i$ denotes the ith component of $\mathbf{G}$ (see (8.54))

$$\begin{aligned} \overline{L} &= \begin{bmatrix} D_x^T & D_y^T \end{bmatrix} \begin{bmatrix} B(\mathbf{f})^{-1} & \mathbf{0} \\ \mathbf{0} & B(\mathbf{f})^{-1} \end{bmatrix} \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \begin{bmatrix} D_x \\ D_y \end{bmatrix} \\ &= D_x^TB(\mathbf{f})^{-1}E_{11}D_x + D_x^TB(\mathbf{f})^{-1}E_{12}D_y + D_y^TB(\mathbf{f})^{-1}E_{21}D_x \\ &\quad + D_y^TB(\mathbf{f})^{-1}E_{22}D_y \end{aligned} \tag{8.61}$$

and

$$\begin{aligned} \mathbf{r} &= -\mathbf{g}_3 + \alpha D_x^TB(\mathbf{f})^{-1}D_x\mathbf{g}_1 + \alpha D_y^TB(\mathbf{f})^{-1}D_y\mathbf{g}_2 \\ &= K^T(\mathbf{d} - K\mathbf{f}) - \alpha D_x^TB(\mathbf{f})^{-1}D_x\mathbf{f} - \alpha D_y^TB(\mathbf{f})^{-1}D_y\mathbf{f}. \end{aligned} \tag{8.62}$$

Consequently,

$$\Delta\mathbf{f} = [K^TK + \alpha\overline{L}]^{-1}\mathbf{r}, \tag{8.63}$$

$$\Delta\mathbf{u} = -\mathbf{u} + B(\mathbf{f})^{-1}[D_x\mathbf{f} + E_{11}D_x\Delta\mathbf{f} + E_{12}D_y\Delta\mathbf{f}], \tag{8.64}$$

$$\Delta\mathbf{v} = -\mathbf{v} + B(\mathbf{f})^{-1}[D_y\mathbf{f} + E_{21}D_x\Delta\mathbf{f} + E_{22}D_y\Delta\mathbf{f}]. \tag{8.65}$$

We employ backtracking to the boundary to maintain the constraint (8.49). In other words, we compute

$$\overline{\tau} = \max\{0 \le \tau \le 1 \mid (u_{ij} + \tau\Delta u_{ij},\ v_{ij} + \tau\Delta v_{ij}) \in C^* \text{ for all } i,\, j\}.$$

We then update

$$\mathbf{u} := \mathbf{u} + \overline{\tau}\Delta\mathbf{u}, \qquad \mathbf{v} := \mathbf{v} + \overline{\tau}\Delta\mathbf{v}.$$

Practical experience [21] suggests that no globalization is needed in the update $\mathbf{f} := \mathbf{f} + \Delta\mathbf{f}$.

Algorithm 8.2.4. Primal-Dual Newton's Method for Total Variation–Penalized Least Squares Minimization in Two Space Dimensions.

```
$\nu := 0$;
$\mathbf{f}_0$ := initial guess for primal variable;
$\mathbf{u}_0, \mathbf{v}_0$ := initial guesses for dual variables;
begin primal-dual Newton iterations
    $B_\nu^{-1} := \mathrm{diag}(\psi'(\mathbf{f}_\nu))$;
    $\mathbf{w} := 2\psi'(\mathbf{f}_\nu)./\psi''(\mathbf{f}_\nu)$;
    $E_{11} := \mathrm{diag}(\mathbf{w}.*(D_x\mathbf{f}_\nu).*\mathbf{u}_\nu)$;
    $E_{12} := \mathrm{diag}(\mathbf{w}.*(D_y\mathbf{f}_\nu).*\mathbf{u}_\nu)$;
    $E_{21} := \mathrm{diag}(\mathbf{w}.*(D_x\mathbf{f}_\nu).*\mathbf{v}_\nu)$;
    $E_{22} := \mathrm{diag}(\mathbf{w}.*(D_y\mathbf{f}_\nu).*\mathbf{v}_\nu)$;
    $\overline{L}_\nu := D_x^T B_\nu^{-1} E_{11} D_x + D_x^T B_\nu^{-1} E_{12} D_y + D_y^T B_\nu^{-1} E_{21} D_x$
        $+ D_y^T B_\nu^{-1} E_{22} D_y$; % discretized diffusion operator
    $\mathbf{r}_\nu := K^T(\mathbf{d} - K\mathbf{f}_\nu) - \alpha(D_x^T B_\nu^{-1} D_x + D_y^T B_\nu^{-1} D_y)\mathbf{f}_\nu$;
    $\Delta\mathbf{f} := (K^T K + \alpha\,\overline{L}_\nu)^{-1}\mathbf{r}_\nu$;    % Newton step
    $\Delta\mathbf{u} := -\mathbf{u}_\nu + B_\nu^{-1}[D_x\mathbf{f}_\nu + (E_{11}D_x + E_{12}D_y)\Delta\mathbf{f}]$;
    $\Delta\mathbf{v} := -\mathbf{u}_\nu + B_\nu^{-1}[D_y\mathbf{f}_\nu + (E_{21}D_x + E_{22}D_y)\Delta\mathbf{f}]$;
    $\mathbf{f}_{\nu+1} := \mathbf{f}_\nu + \Delta\mathbf{f}$;    % update primal variable
    $\tau_\nu := \max\{0 \le \tau \le 1 \mid (\mathbf{u}_\nu + \tau\Delta\mathbf{u}, \mathbf{v}_\nu + \tau\Delta\mathbf{v}) \in C^*\}$;
    $\mathbf{u}_{\nu+1} := \mathbf{u}_\nu + \tau_\nu\Delta\mathbf{u}$;    % update dual variables
    $\mathbf{v}_{\nu+1} := \mathbf{v}_\nu + \tau_\nu\Delta\mathbf{v}$;
    $\nu := \nu + 1$;
end primal-dual Newton iterations
```

Remark 8.8. For large-scale systems where the matrix $K^T K$ does not have a sparse matrix representation, the most expensive part of the algorithm is the inversion of the matrix $A \overset{\text{def}}{=} K^T K + \alpha\overline{L}$ in the computation of the component $\Delta\mathbf{f}$ of the Newton step. If K has block Toeplitz structure, then the techniques of section 5.2.5 can be used to compute matrix-vector products $A\mathbf{v}$ at a cost of $n \log n$ (note that the matrix $\overline{L}$ is sparse; see (8.61)). This suggests the use of iterative linear solvers like the CG Algorithm 3.2. Use of CG is precluded by the fact that $\overline{L}$ need not be symmetric, but one can replace $\overline{L}$ by its symmetric part, $(\overline{L} + \overline{L}^T)/2$, and still retain quadratic convergence of the primal-dual Newton iteration [21]. CG iteration can then be applied as a linear solver. Chan, Chan, and Wong [15] provided preconditioners for CG in this setting.

8.2.6 Other Methods

The computational methods presented in the previous sections are based on smooth approximations to the Euclidean norm of the gradient. Ito and Kunisch [62] presented an alternative approach that is based on the representation (8.4).

One can also replace the Euclidean norm of the gradient by other norms. Li and Santosa [74] made use of the ℓ^1 norm. After discretization in two dimensions, their penalty functional took the form

$$J(f) = \sum_i \sum_j |D_{ij}^x f| + |D_{ij}^y f|.$$

They then applied an interior point method to solve their minimization problem. It should be noted that unlike the Euclidean norm, the ℓ^1 norm is not rotationally invariant. This may have the unfortunate consequence of making the reconstruction dependent on the orientation of the computational grid.

The time evolution approach outlined in Remark 8.2 provides one example of a broad class of nonlinear PDE-based techniques called nonlinear diffusion methods. These methods have found important applications in computer vision and image processing. See [124] for details and references.

Finally, we note that there exist several other mathematical expressions for variation that are closely related but not equivalent to (8.4). See [73] for details and references.

8.3 Numerical Comparisons

In this section, we compare the performance of some of the solution methods presented in the previous sections.

8.3.1 Results for a One-Dimensional Test Problem

The one-dimensional test problem is described in section 1.1, and the data used are presented in Figure 1.1. To solve this problem, we minimized the discrete regularized least squares functional (8.8)–(8.10) with

$$\frac{1}{2}\psi((D_i\mathbf{f})^2) = \sqrt{\left(\frac{f_i - f_{i-1}}{\Delta x}\right)^2 + \beta^2} \tag{8.66}$$

using the four iterative solution methods presented in sections 8.2.3–8.2.5.

The matrix K in (8.8) is Toeplitz and not sparse. In the primal-dual Newton implementation, the conjugate set C^* is the interval $-1 \leq u \leq 1$; see Example 8.5.

The reconstruction obtained using a one-dimensional version of the primal-dual Newton Algorithm 8.2.5 is shown in Figure 1.5. The functional (8.8) is strictly convex for this test example, so the other methods yield essentially the same reconstruction provided the minimizer is computed to sufficient accuracy.

One of our primary measures of numerical performance is the relative iterative solution error norm,

$$e_\alpha^\nu = \frac{||\mathbf{f}_\alpha^\nu - \mathbf{f}_\alpha||}{||\mathbf{f}_\alpha||}, \tag{8.67}$$

where $\mathbf{f}_\alpha$ represents the minimizer of (8.8) and $\mathbf{f}_\alpha^\nu$ represents the numerical approximation to $\mathbf{f}_\alpha$ at iteration ν. In place of the exact $\mathbf{f}_\alpha$ we used an extremely accurate approximation obtained with the primal-dual Newton method.

The performances of the steepest descent method (Algorithm 8.2.3) and Newton's method with a line search (Algorithm 8.2.3) are compared in Figure 8.3. Initially the steepest descent method exhibits a rapid decrease in the iterative solution error, but almost no change in the reconstructions is observed after the first five steepest descent iterations. This is consistent with Theorem 3.5, since the Hessian is quite ill-conditioned.

With Newton's method very little progress occurs until about iteration 50. During the earlier iterations, the line search restricts the step size. In the last six iterations, there is a dramatic decrease in solution error, as the local quadratic convergence rate characteristic of Newton's method is finally attained. This behavior is consistent with the theory presented in section 3.3. Note that the convergence constant c_* in (3.18) becomes large as the minimum eigenvalue of the Hessian becomes small and the Lipschitz constant γ becomes large. The former event occurs when the regularization parameter α is small; the latter occurs when the parameter β in (8.66) becomes small. When c_* is large, $\mathbf{f}_\alpha^\nu$ must be quite close to $\mathbf{f}_\alpha$ before the iteration will converge without a line search.

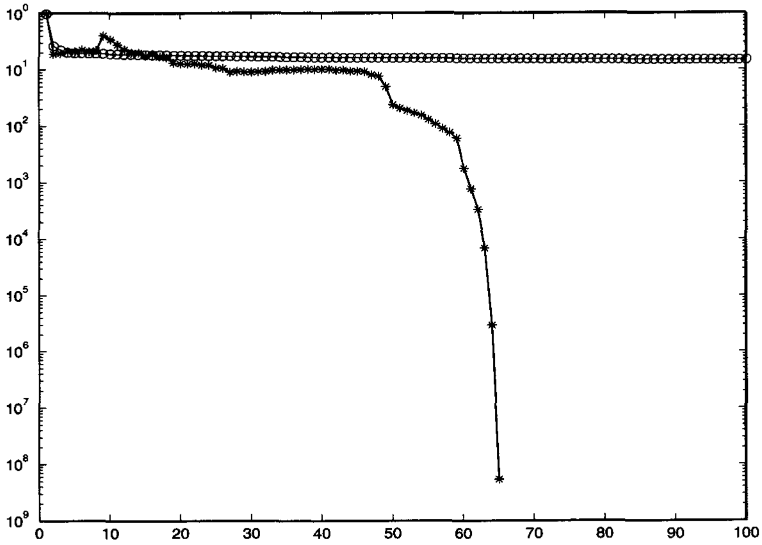

Figure 8.3. *Numerical performance of the steepest descent method and Newton's method on a one-dimensional test problem. The relative iterative solution error norm* (8.67) *is plotted against the iteration count. Circles represent the results for the steepest descent method (Algorithm* 8.2.3*), and asterisks represent the results for the primal Newton method (Algorithm* 8.2.3*).*

In Figure 8.4 the performance of the lagged diffusivity fixed point method (Algorithm 8.2.4) and the primal-dual Newton method (Algorithm 8.2.5) are compared. The lagged diffusivity fixed point method displays rapid decrease in the solution error during the first few iterations. Convergence then slows to a steady linear rate. The primal-dual Newton method also displays fairly fast initial convergence. After about eight iterations, a more rapid quadratic convergence rate can be seen.

Note that the steepest descent method requires a nonsparse matrix-vector multiplication at each iteration, since K is a full matrix. The other three methods require the inversion of nonsparse linear systems at each iteration. (We used Gaussian elimination to solve these systems.) Hence, the cost per iteration of the steepest descent method is significantly less than that of the other three methods. However, for this particular test problem, the extremely slow convergence rate of steepest descent negates the advantage of low computational cost per iteration. The other three methods all have roughly the same cost per iteration. Due to its more rapid convergence rate, the primal-dual Newton method is the most efficient method for this problem.

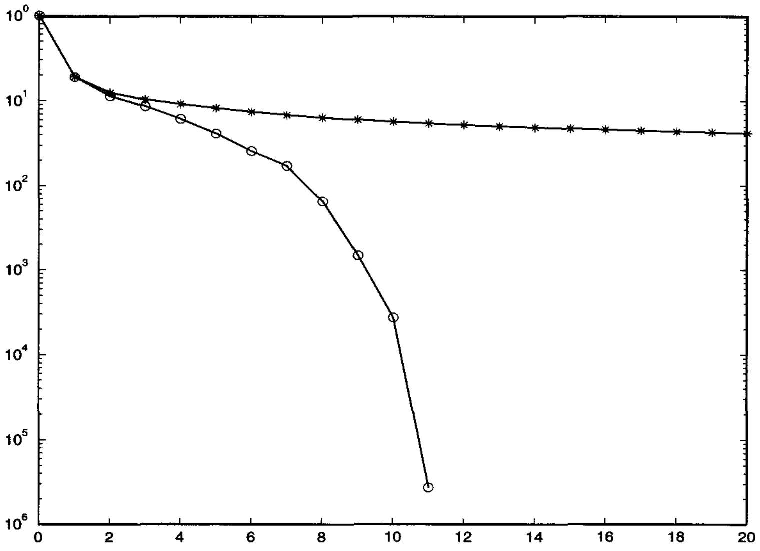

Figure 8.4. *Numerical performance of the lagged diffusivity fixed point iteration and the primal-dual Newton method on a one-dimensional test problem. Asterisks represent the relative iterative solution error norm for the fixed point method (Algorithm* 8.2.4*), and circles represent the results for the primal-dual Newton method (Algorithm* 8.2.5*).*

8.3.2 Two-Dimensional Test Results

The image deblurring test problem in this section is described in section 5.1.1, the test data are similar to that shown in Figure 5.2, and the reconstructions are similar to that shown in Figure 8.1. To obtain the reconstructions, we minimized a two-dimensional version of the penalized least squares functional (8.8)–(8.10) with

$$\frac{1}{2}\psi((D^x_{ij}\mathbf{f})^2 + (D^y_{ij}\mathbf{f})^2) = \sqrt{\left(\frac{f_{i,j} - f_{i-1,j}}{\Delta x}\right)^2 + \left(\frac{f_{i,j} - f_{i,j-1}}{\Delta x}\right)^2 + \beta^2}.$$

We present numerical performance results (see Figure 8.5) only for the lagged diffusivity fixed point Algorithm 8.2.4 and for the primal-dual Newton Algorithm 8.2.5. Comparison of the other methods is left to Exercise 8.14.

In the primal-dual Newton implementation, the conjugate set C^* is the unit ball in $\mathbb{R}^2$; see Example 8.5. The matrix K in (8.8) is block Toeplitz with Toeplitz blocks; see section 5.2.5. As in our one-dimensional test problem, K is not sparse.

As in one dimension, the lagged diffusivity fixed point convergence is rapid at first but slows to a steady linear rate after a few iterations. Primal-dual Newton displays fairly rapid initial convergence, with an increase in the convergence rate at later iterations. For this test problem, primal-dual Newton clearly converges at a much more rapid rate.

Both lagged diffusivity and primal-dual Newton require the solution of nonsparse linear systems at each iteration. In this two-dimensional application, these systems are large enough to discourage the use of direct matrix decomposition methods. Instead we applied the CG Algorithm 3.2 with no preconditioning. (See Remark 8.8 for primal-dual Newton implementation details.) We found that we needed a very small (residual) CG stopping tolerance to maintain rapid convergence of the primal-dual Newton iterations. A much more relaxed CG stopping tolerance could be used without degrading the convergence of the lagged diffusivity iteration. Consequently, the cost per iteration of primal-dual Newton was significantly larger than the cost per iteration of lagged diffusivity fixed point. This may no longer be the case if preconditioning is applied; see [15] and Exercise 8.15. An overall cost comparison is difficult to carry out, since it depends on factors like stopping tolerances; values of parameters like α, β, and the system size; and the effectiveness of the preconditioner.

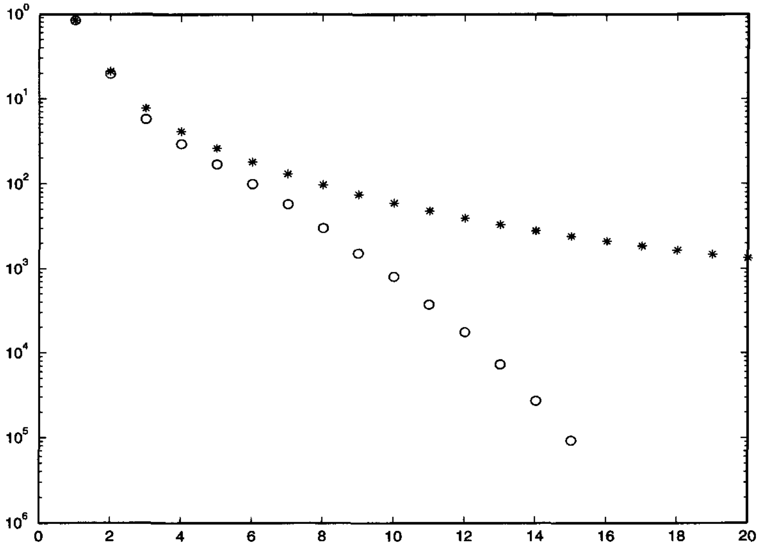

Figure 8.5. *Comparison of lagged diffusivity fixed point iteration and primal-dual Newton iteration for a two-dimensional image reconstruction problem. Asterisks denote relative iterative solution error for the fixed point iteration, and circles denote error for primal-dual Newton.*

Total variation methods have been applied to more general inverse problems. See [33] for an application to distributed parameter identification.

8.4 Mathematical Analysis of Total Variation

In this section, Ω denotes a simply connected, nonempty, open subset of $\mathbb{R}^d$, $d = 1, 2, \dots$, with Lipschitz continuous boundary. In imaging applications, Ω is typically the unit square in $\mathbb{R}^2$. We use the symbol ∇ to denote the gradient of a smooth function $f : \mathbb{R}^d \to \mathbb{R}^1$, i.e., $\nabla f = (\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_d})$. $C_0^1(\Omega; \mathbb{R}^d)$ denotes the space of vector-valued functions $\vec{v} = (v_1, \dots, v_d)$ whose component functions v_i are each continuously differentiable and compactly supported on Ω, i.e., each v_i vanishes outside some compact subset of Ω. The divergence of $\vec{v}$ is given by

$$\operatorname{div} \vec{v} = \sum_{i=1}^{d} \frac{\partial v_i}{\partial x_i}.$$

The Euclidean norm is denoted by $|\cdot|$. In particular, $|\vec{v}(x)| = [\sum_{i=1}^{d} v_i(x)^2]^{1/2}$. The Sobolev space $W^{1,1}(\Omega)$ denotes the closure of $C_0^1(\Omega)$ with respect to the norm

$$||f||_{1,1} = \int_\Omega \left[|f| + \sum_{i=1}^{d} \left| \frac{\partial f}{\partial x_i} \right| \right].$$

The following definition is taken from Giusti [45].

Definition 8.9. The total variation of a function $f \in L^1(\Omega)$ is defined by

$$\text{TV}(f) = \sup_{\vec{v} \in \mathcal{V}} \int_\Omega f \operatorname{div} \vec{v}\, dx, \tag{8.68}$$

where the space of test functions

$$\mathcal{V} = \{\vec{v} \in C_0^1(\Omega; \mathbb{R}^d) \mid |\vec{v}(x)| \le 1 \text{ for all } x \in \Omega\}. \tag{8.69}$$

Remark 8.10. Equation (8.68) can be viewed as a weak form of

$$\text{TV}(f) = \int_\Omega |\nabla f|\, dx.$$

Using the dual representation (8.37) for the Euclidean norm and formally applying integration by parts,

$$\begin{aligned} \int_\Omega |\nabla f|\, dx &= \int_\Omega \sup_{|\vec{v}| \le 1} \nabla f^T \vec{v}\, dx \\ &= \sup_{|\vec{v}| \le 1} \left[\int_{\partial\Omega} f\, \vec{v}^T \hat{n}\, dS - \int_\Omega f \operatorname{div} \vec{v}\, dx \right], \end{aligned}$$

where $\partial\Omega$ denotes the boundary of Ω and $\hat{n}$ denotes the outward unit normal to $\partial\Omega$. If $\vec{v}$ is compactly supported in Ω, then the boundary integral term vanishes. Note that $|\vec{v}| \le 1$ if and only if $|-\vec{v}| \le 1$, so we can drop the minus sign to obtain (8.68)–(8.69).

Example 8.11. Let $\Omega = [0, 1] \subset \mathbb{R}^1$, and define

$$f(x) = \begin{cases} f_0, & x < 1/2, \\ f_1, & x > 1/2, \end{cases}$$

where f_0, f_1 are constants. For any $v \in C_0^1[0, 1]$,

$$\int_0^1 f(x)v'(x)\, dx = \int_0^{1/2} f(x)v'(x)\, dx + \int_{1/2}^1 f(x)v'(x)\, dx = (f_0 - f_1)\, v(1/2).$$

This quantity is maximized over all $v \in \mathcal{V}$ when $v(1/2) = \text{sign}(f_0 - f_1)$. This yields $\text{TV}(f) = |f_1 - f_0|$, which agrees with (8.1).

Example 8.12. Let E be a set contained in $\Omega \subset \mathbb{R}^d$, with $d \ge 2$, and assume its boundary ∂E is C^2. Let $f(x) = f_0$ if $x \in E$ and $f(x) = 0$ otherwise. In this case,

$$\text{TV}(f) = f_0 \text{ Area}(\partial E),$$

where Area($\cdot$) denotes surface area. (This reduces to arc length when the dimension $d = 2$.) To verify, for any $\vec{v} \in C_0^1(\Omega; \mathbb{R}^d)$ the divergence theorem yields

$$\int_\Omega f \operatorname{div} \vec{v}\, dx = f_0 \int_E \operatorname{div} \vec{v}\, dx = f_0 \int_{\partial E} \vec{v}^T \hat{n}\, dS, \tag{8.70}$$

where $\hat{n}(x)$ denotes the outward unit normal to ∂E at x and dS denotes surface integration. Imposing $|\vec{v}(x)| \le 1$, we obtain from (8.68) that $\mathrm{TV}(f) \le f_0\mathrm{Area}(\partial S)$. Since E has C^2 boundary, its outward unit normal $\hat{n}(x)$ will be a C^1 vector-valued function, which can be extended to a function $\vec{v} \in C_0^1(\Omega; \mathbb{R}^d)$ for which $|\vec{v}(x)| \le 1$. Then by (8.68) and (8.70), $\mathrm{TV}(f) \ge f_0 \int_{\partial E} |\hat{n}(x)|^2\, dS = f_0\mathrm{Area}(\partial S)$.

Proposition 8.13. *If $f \in W^{1,1}(\Omega)$, then*

$$\mathrm{TV}(f) = \int_\Omega |\nabla f|. \tag{8.71}$$

Proof. If $f \in C^1(\Omega)$ and $\vec{v} \in C_0^1(\Omega; \mathbb{R}^d)$, then integration by parts yields

$$\int_\Omega f \operatorname{div} \vec{v}\, dx = -\int_\Omega \nabla f^T \vec{v}\, dx.$$

Take

$$\vec{w}(x) = \begin{cases} -\frac{\nabla f}{|\nabla f|} & \text{if } \nabla f(x) \neq 0, \\ 0 & \text{if } \nabla f(x) = 0. \end{cases}$$

One can pick $\vec{v} \in C_0^1(\Omega; \mathbb{R}^d)$ with components arbitrarily close to those of $\vec{w}$ with respect to the L^2 norm, and, hence, (8.71) holds for any $f \in C^1(\Omega)$. By a standard denseness argument, this also holds for $f \in W^{1,1}(\Omega)$. □

Definition 8.14. The space of functions of bounded variation, denoted by $BV(\Omega)$, consists of functions $f \in L^1(\Omega)$ for which

$$||f||_{BV} \overset{\text{def}}{=} ||f||_{L^1(\Omega)} + \mathrm{TV}(f) < \infty. \tag{8.72}$$

Theorem 8.15. $||\cdot||_{BV}$ *is a norm, and $BV(\Omega)$ is a Banach space under this norm. The TV functional is a seminorm on this space.*

See Giusti [45] for a proof of this theorem. Proposition 8.13 and Examples 8.11 and 8.12 show that $W^{1,1}(\Omega)$ is a proper subspace of $BV(\Omega)$.

The following three theorems pertain to the important properties of compactness, convexity, and semicontinuity. Proofs can be found in [2].

Theorem 8.16. *Let S be a BV-bounded set of functions. For $\Omega \subset \mathbb{R}^d$, S is a relatively compact subset of $L^p(\Omega)$ for $1 \le p < d/(d-1)$ and is weakly relatively compact in $L^{d/(d-1)}(\Omega)$. In case the dimension $d = 1$, we set $d/(d-1) = +\infty$.*

Theorem 8.17. *The* TV *functional* (8.68), *defined on the space $BV(\Omega)$, is convex but not strictly convex. The restriction of this functional to $W^{1,1}(\Omega)$ is strictly convex.*

Theorem 8.18. *The* TV *functional is weakly lower semicontinuous with respect to the L^p norm topology for $1 \le p < \infty$.*

We next examine the existence, uniqueness, and stability of minimizers of the BV-penalized least squares functional

$$T(f) = ||Kf - d||^2_{L^2(\Omega)} + \alpha||f||_{BV}, \qquad \alpha > 0. \tag{8.73}$$

Theorem 8.19. *Let* $1 \leq p < d/(d-1)$, *and let* C *be a closed, convex subset of* $L^p(\Omega)$. *Assume* $K : L^p(\Omega) \to L^2(\Omega)$ *is linear, bounded, and* $\text{Null}(K) = \{0\}$. *Then, for any fixed* $d \in L^2(\Omega)$, *the functional in* (8.73) *has a unique constrained minimizer,*

$$f_* = \arg\min_{f \in C} T(f).$$

Proof. Existence follows arguments similar to those of Theorem 2.30. See [2] for details. Note that since K is linear with a trivial null space and the squared Hilbert space norm is strictly convex, the mapping $f \mapsto ||Kf - d||^2_{L^2(\Omega)}$ is strictly convex. Uniqueness follows from strict convexity. □

The following stability result is proved in [2].

Theorem 8.20. *Suppose the hypotheses of Theorem* 8.19 *hold. Then the minimizer* f_* *is stable with respect to*

(i) *perturbations* d_n *of the data* d *for which* $||d_n - d||_{L^2(\Omega)} \to 0$;

(ii) *perturbations* K_n *of the operator* K *for which* $||K_n(f) - K(f)||_{L^2(\Omega)} \to 0$ *uniformly on compact subsets in* $L^p(\Omega)$;

(iii) *perturbations* α_n *of the regularization parameter* $\alpha > 0$.

Similar existence-uniqueness-stability results can be obtained when the BV norm (8.72) is replaced by the TV functional (8.68), yielding

$$T(f) = ||Kf - d||^2_{L^2(\Omega)} + \alpha \text{TV}(f). \tag{8.74}$$

The condition that K has a trivial null space can also be weakened somewhat. The following result is an example. See [2] for a proof.

Theorem 8.21. *Let* C *be a closed, convex subset of* $L^p(\Omega)$ *with* $1 \leq p < d/(d-1)$. *Let* $K : L^p(\Omega) \to L^2(\Omega)$ *be linear and bounded. Assume that* $K1 \neq 0$, *where* 1 *denotes the function* $1(x) = 1$ *for all* $x \in \Omega$. *Then the functional in* (8.74) *has a unique constrained minimizer over* C.

8.4.1 Approximations to the TV Functional

As in section 8.2.5, we replace the Euclidean norm $|\cdot|$ by a smooth, convex approximation φ. For smooth f one can define a corresponding approximation to the TV functional,

$$J(f) = \int_\Omega \varphi(\nabla f)\, dx, \tag{8.75}$$

which is analogous to the representation (8.71).

To obtain an extension of the functional in (8.75) that is valid for nonsmooth f in a manner analogous to (8.68), we make use of the dual representation

$$J(f) = \sup_{\vec{v} \in \mathcal{V}} \int_\Omega [-f \operatorname{div} \vec{v} - \varphi^*(\vec{v}(x))]\,dx, \tag{8.76}$$

where

$$\mathcal{V} = \{\vec{v} \in C_0^1(\Omega; \mathbb{R}^d) \mid \vec{v}(x) \in C^* \text{ for all } x \in \Omega\}.$$

Equation (8.76) is obtained from (8.35) by replacing x with ∇f, replacing y with $\vec{v}(x)$, and integrating by parts; see Remark 8.10. Motivated by Examples 8.5 and 8.6, we define

$$J_\beta(f) = \sup_{\vec{v} \in \mathcal{V}} \int_\Omega \left[-f \operatorname{div} \vec{v} + \beta\sqrt{1 - |\vec{v}(x)|^2}\right] dx \tag{8.77}$$

and

$$J_\epsilon(f) = \sup_{\vec{v} \in \mathcal{V}} \int_\Omega \left[-f \operatorname{div} \vec{v} - \frac{\epsilon}{2}|\vec{v}(x)|^2\right] dx, \tag{8.78}$$

where $\mathcal{V}$ is given in (8.69).

The following results establish stability of total variation regularized solutions with respect to perturbations (8.77) and (8.78) of the TV functional as the parameters β and ϵ tend to zero. See [2] for proofs.

Proposition 8.22. *Both J_β and J_ϵ are convex and weakly lower semicontinuous. Moreover, $J_\beta(f) \to \mathrm{TV}(f)$ as $\beta \to 0$ and $J_\epsilon(f) \to \mathrm{TV}(f)$ as $\epsilon \to 0$, uniformly on BV-bounded sets.*

Theorem 8.23. *Total variation regularized solutions are stable with respect to certain perturbations in the penalty functional. In particular, if $\mathrm{TV}(f)$ in (8.74) is replaced by either $J_\beta(f)$ or $J_\epsilon(f)$ and $\alpha > 0$ is fixed, then the corresponding regularized solutions $f_{\alpha,\beta}$ and $f_{\alpha,\epsilon}$ converge to the total variation regularized solution in L^p norm, $1 \le p < d/(d-1)$, as $\beta \to 0$ and $\epsilon \to 0$.*

Exercises

8.1. Prove that $L(\mathbf{f})$ in (8.17) is a positive semidefinite matrix. What is the null space of $L(\mathbf{f})$?

8.2. Let the functional J be as in (8.28). Recall from Remark 2.35 that its Gateaux, or directional, derivative at f in the direction h is given by

$$\delta J(f;h) = \frac{d}{d\tau} J(f + \tau h)|_{\tau=0}.$$

Show that for smooth f and h,

$$\delta J(f;h) = \int_0^1 \int_0^1 \psi'(|\nabla f|^2)\, \nabla f^T \nabla h \, dx\, dy.$$

Then show that $\delta J(f;h) = \langle \mathcal{L}(f) f, h\rangle$, provided that the normal derivative of f vanishes on the boundary of the unit square. Here $\mathcal{L}(f)$ is given in (8.27) and $\langle \cdot, \cdot \rangle$ denotes the L^2 inner product on the unit square.

8.3. Derive the two-dimensional representation for $L'(\mathbf{f})\mathbf{f}$ in (8.29).

8.4. Suppose that explicit time marching, or the forward Euler method, is applied to the system of ODEs:

$$\frac{d\mathbf{f}}{dt} = -\mathrm{grad}\, T(f),$$

where the functional T is given in (8.8). Show that the resulting iteration is equivalent to that of the steepest descent Algorithm 8.2.3, except that the line search parameter $\tau_\nu = \Delta t$ is fixed.

8.5. Show that the right-hand side of (8.30) is equivalent to (8.31).

8.6. Verify (8.35) directly. *Hint:* Use the Cauchy–Schwarz inequality to show that the left-hand side is bounded by the right-hand side. Then show that the bound is attained.

8.7. Verify equation (8.39).

8.8. With φ_ϵ given in Example 8.6, verify that

$$\sup_{\mathbf{x}\in\mathbb{R}^d}\{\mathbf{x}^T\mathbf{y} - \varphi_\epsilon(\mathbf{x})\} = \begin{cases} \frac{\epsilon}{2}|\mathbf{y}|^2 & \text{if} \quad |\mathbf{y}| \le 1, \\ +\infty & \text{if} \quad |\mathbf{y}| > 1. \end{cases}$$

8.9. By applying block Gaussian elimination to the right-hand side of (8.55), derive the expression for $\overline{L}$ in (8.61). Also, derive (8.62).

8.10. For the matrix $\overline{L}$ in (8.61), prove that the symmetric part $(\overline{L} + \overline{L}^T)/2$ is positive semidefinite.

8.11. For the one-dimensional test problem of section 8.3.1, conduct a numerical study of the effects of varying the parameters α and β on the performance of each of the four algorithms applied in that section. In particular, what are the effects on numerical performance of making β very small?

8.12. What is the qualitative effect on the reconstructions in the one-dimensional test problem of *increasing* the parameter β?

8.13. For the one-dimensional test problem, replace the approximation (8.38) to the absolute value by (8.41). Explain why one cannot then implement either the primal Newton method or the primal-dual Newton method. Implement and compare results for the remaining two methods, the steepest descent method and the lagged diffusivity fixed point method.

8.14. For the two-dimensional test problem of section 8.3.2, implement both the steepest descent Algorithm 8.2.3 and the Newton Algorithm 8.2.3. How do these methods compare in terms of convergence rates and computational cost?

8.15. In the implementation of the lagged diffusivity fixed point method and the primal-dual Newton method for two-dimensional test problem, replace the CG linear solver with preconditioned CG. Use the level 2 block circulant preconditioner of section 5.3.3.

8.16. Prove Proposition 8.22. Use the facts that $\mathrm{TV}(f) \le J_\beta(f) \le \mathrm{TV}(f) + \beta\ \mathrm{Vol}(\Omega)$ and $\mathrm{TV}(f) - \frac{\epsilon}{2}\ \mathrm{Vol}(\Omega) \le J_\epsilon(f) \le \mathrm{TV}(f)$. Here $\mathrm{Vol}(\Omega) = \int_\Omega dx$ denotes the volume of the set Ω.