SIAM J. SCI. COMPUT. © 2010 Society for Industrial and Applied Mathematics
Vol. 32, No. 3, pp. 1591–1616

SURFACE RECONSTRUCTION AND IMAGE ENHANCEMENT VIA L^1-MINIMIZATION*

VESELIN DOBREV†, JEAN-LUC GUERMOND‡, AND BOJAN POPOV§

Abstract. A surface reconstruction technique based on minimization of the total variation of the gradient is introduced. Convergence of the method is established, and an interior-point algorithm solving the associated linear programming problem is introduced. The reconstruction algorithm is illustrated on various test cases including natural and urban terrain data, and enhancement of low-resolution or aliased images.

Key words. finite elements, L^1-minimization, interior-point algorithm, data reconstruction, digital elevation models, image enhancement

AMS subject classifications. 41A29, 65N30, 90C25

DOI. 10.1137/09075408X

1. Introduction.

In geometric modeling and image reconstruction, one often tries to *extract* a shape or recover a piecewise smooth surface from a set of measurements. That is, one wants to find a surface that satisfies constraints or fits given data and is visually pleasing. The objectives could vary with the applications, but the intuitive goal is to preserve the *shape* of the object. For example, one may want to reconstruct a convex body if the underlying data comes from a convex object, or reconstruct a flat surface if the data is locally flat, or preserve a particular structure of the level sets. Sometimes, this type of problem is solved by minimizing an L^p-norm of the Hessian or the total variation of the gradient; see, for example, [5, 7, 18]. In this paper we take a different approach, which we think is well suited for man-made surfaces, digital elevation models (DEM), and enhancement of digital images. Namely, we minimize the total variation of the gradient of a function constructed on a finite element space satisfying interpolatory constraints. Similar minimization problems have been introduced by Lavery [16, 17, 18, 28] and are hereafter referred to as the L^1-spline techniques. Minimizing the total variation of the gradient of a smooth function amounts to minimizing the L^1-norm of its second derivatives. The key observation from Lavery's work is that using the L^1-norm in the minimization process produces oscillation-free surfaces.

In recent years, the idea of using the L^1-metric instead of the usual L^2-metric was exploited in many different areas with great success. For example, in compressed sensing [3, 4] the ℓ_1-metric is used in the decoding step, and in partial differential equations the L^1-norm is used to measure the residual of the equation [9, 10, 11, 12, 14, 15].

*Received by the editors March 26, 2009; accepted for publication (in revised form) February 12, 2010; published electronically June 9, 2010. This material is based upon work supported by the National Science Foundation grants DMS-0510650 and DMS-0811041. This publication is based on work partially supported by Award KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST).
http://www.siam.org/journals/sisc/32-3/75408.html

†Lawrence Livermore National Laboratory, Box 808, L-560, Livermore, CA 94551-0808 (dobrev@llnl.gov). This author's work was performed while the author was employed by Texas A&M University.

‡Department of Mathematics, Texas A&M University, 3368 TAMU, College Station, TX 77843, and LIMSI, UPRR 3251 CNRS, BP 133, 91403 Orsay cedex, France (guermond@math.tamu.edu).

§Department of Mathematics, Texas A&M University, 3368 TAMU, College Station, TX 77843 (popov@math.tamu.edu).

Some image denoising techniques are based on minimizing a K-functional comprising the L^p-distance between the reconstructed image and the noisy data plus the total variation of the reconstruction [6, 24]. In all of the above applications, using L^1 is critical to obtaining good numerical results and proving theoretical estimates.

A key ingredient in Lavery's work is the use of C^1-splines. The novelty of the approach introduced in the present paper is to *relax* the C^1-smoothness on the finite element space which is used in the data reconstruction process; i.e., we propose using a discrete space composed of continuous finite elements with possibly discontinuous gradients. This is the natural discretization setting for functions that are in $W^{1,1}$ and whose gradients have bounded total variation. The objective of the present paper is to describe in detail this new technique, characterize some of its mathematical properties, and evaluate it numerically on a series of standard benchmark tests.

The paper is organized as follows. The reconstruction technique is described in section 2; it essentially amounts to minimizing a functional measuring the norm of the Hessian in L^1. Key results of section 2 are (i) a convergence proof of the method in the BV-norm and (ii) Proposition 2.5, which states that the proposed reconstruction technique can generate L^1-like splines as a special case. In section 3 we go over implementation details; in particular, we describe the interior-point algorithm that we use to solve the minimization problem. Some properties of this algorithm are discussed. The performance of the method is evaluated on various types of data in sections 4 and 5. We reconstruct topography from terrain data in section 4, and we use the method to enhance the resolution of underresolved or aliased images in section 5. The proof of Proposition 2.5 comprises the appendix.

2. The L^1-minimization problem.

2.1. The model problem. Let Ω be a bounded polygonal domain in $\mathbb{R}^2$. We use the notation $W^{s,p}(\Omega)$, $s \geq 0$, $p \in [1,\infty]$, for Sobolev spaces, where s is the smoothness index and p is the metric index. We denote X to be the subspace of $W^{1,1}(\Omega)$ composed of the scalar-valued functions whose gradient has bounded variations,

$$X := \{v \in W^{1,1}(\Omega) \mid \nabla v \in [BV(\Omega)]^2\}. \tag{2.1}$$

For any scalar function v we define the seminorm

$$|v|_{BV} = \sup_{0 \neq \phi \in [\mathcal{C}_0^\infty(\Omega)]^2} \frac{\int_\Omega v \nabla \cdot \phi \, \mathrm{d}x}{\|\phi\|_{L^\infty}}, \tag{2.2}$$

where we use $\|\phi\|_{L^\infty} = \|(\phi_1^2 + \phi_2^2)^{\frac{1}{2}}\|_{L^\infty}$ for all $\phi \in [\mathcal{C}_0^\infty(\Omega)]^2$. For the gradient of $v \in X$ we use $|\nabla v|_{BV} = |\partial_1 v|_{BV} + |\partial_2 v|_{BV}$.

We assume that we are given a set of linear functionals acting on X; this set is denoted $d := \{d_i\}_{i \in \overline{1,I}}$. We assume also that we are given a set of measurements of a function u using these functionals; this data set is denoted $\varpi := \{\varpi_i\}_{i \in \overline{1,I}}$, where $\varpi_i := d_i(u)$, $i \in \overline{1,I}$.

Our goal is to reconstruct a smooth (eye-pleasing) approximation of u, say, u_h, such that the array $(d_1(u_h), \dots, d_I(u_h))$ is either equal to $(\varpi_1, \dots, \varpi_I)$ or a good approximation thereof in a sense yet to be defined. A simple solution to this problem could be to define a polynomial interpolant, but this technique is known to produce oscillatory reconstructions when using high-order polynomials. The approach that we follow in this paper is to define an approximation space whose dimension is a multiple of the cardinality of the data set I, and to reduce possible oscillations we select an approximant whose Hessian has minimal L^1-norm.

The idea of minimizing the norm of the Hessian in L^1 is rooted in a series of papers by Lavery and coworkers [17, 18, 28], where the approximating functions are $\mathcal{C}^1$-splines. The problem with this choice is that $\mathcal{C}^1$-continuity does not allow for sharp edges; hence, if the object to be reconstructed has edges, then the $\mathcal{C}^1$-spline technique smooths them. To overcome this unsatisfactory property of $\mathcal{C}^1$-splines, we propose a new idea which consists of using cubic $\mathcal{C}^0$ finite elements and minimizing the total variation of the gradient of the approximating function. This leads to an ℓ_1-minimization problem that we solve using an interior-point algorithm.

2.2. The finite element setting. Let $\mathcal{T}_h$ be a partition of Ω composed of open triangles and/or quadrilaterals, $\overline{\Omega} = \bigcup_{T\in\mathcal{T}_h} \overline{T}$. The mesh $\mathcal{T}_h$ is conforming in the sense that for any pair of distinct elements T, T' in $\mathcal{T}_h$, the intersection $T \cap T'$ is empty and $\overline{T} \cap \overline{T}'$ is either a common vertex or a common edge. For any element T in $\mathcal{T}_h$, we denote by h_T the diameter of T. Conformity is known to be necessary for constructing continuous finite elements. The conformity assumption could be relaxed by introducing hanging nodes which we avoid to simplify the presentation.

We introduce the discrete space X_h composed of continuous functions that are piecewise cubic on the mesh $\mathcal{T}_h$:

$$X_h = \{u \in \mathcal{C}^0(\overline{\Omega}) : u|_T \in \mathbb{P}_3 \text{ if } T \text{ is a triangle or } \\ u|_T \in \mathbb{F}_T(\mathbb{Q}_3) \text{ if } T \text{ is a quadrilateral } \forall T \in \mathcal{T}_h\}, \tag{2.3}$$

where we denote

$$\mathbb{P}_p = \left\{ \sum_{i=0}^{p}\sum_{j=0}^{p-i} c_{ij}x^iy^j : c_{ij} \in \mathbb{R} \right\}, \qquad \mathbb{Q}_{pq} = \left\{ \sum_{i=0}^{p}\sum_{j=0}^{q} c_{ij}x^iy^j : c_{ij} \in \mathbb{R} \right\},$$

and the mapping $\mathbb{F}_T$ is defined by

$$(\mathbb{F}_T\hat{q})(x) = \hat{q}(F_T^{-1}(x)) \quad \forall x \in T,\ \hat{q} \in \mathcal{C}^0([0,1]^2),$$

where F_T is the transformation that maps the reference unit square $(0,1)^2$ to the quadrilateral T. We henceforth denote $\mathbb{Q}_p := \mathbb{Q}_{pp}$. Note that X_h is a subspace of X but *not* a subspace of $W^{2,1}(\Omega)$.

The set of the vertices of the triangulation $\mathcal{T}_h$ is denoted $\mathcal{V}_h$. The set of interior edges of the partition $\mathcal{T}_h$ is denoted $\mathcal{F}_h^i$. Let $F \in \mathcal{F}_h^i$ be any of the interior edges, and let $T, T' \in \mathcal{T}_h$ be the two elements whose intersection is $F = \overline{T} \cap \overline{T}'$. Also, let $\boldsymbol{n}_{TF}$ denote the normal vector to F pointing from T to T'. We define the jump of the normal derivative of a function u to be

$$[\![\partial_{\boldsymbol{n}} u]\!]|_F = (\nabla u|_T)\cdot \boldsymbol{n}_{TF} + (\nabla u|_{T'})\cdot \boldsymbol{n}_{T'F}.$$

We now assume that we are given a family of meshes $(\mathcal{T}_h)_{h>0}$ and the corresponding spaces $(X_h)_{h>0}$. We also assume that we are given a family of functionals $(d_h := \{d_1,\ldots,d_{I_h}\})_{h>0}$ acting on $(X_h)_{h>0}$ and a family of measured data $(\varpi_h := \{\varpi_1,\ldots,\varpi_{I_h}\})_{h>0}$. Typically $I_h < \dim(X_h)$. We say that $d_h(v) = \varpi_h$ when $d_1(v) = \varpi_1, \ldots, d_{I_h}(v) = \varpi_{I_h}$.

Remark 2.1. The measurements can be point values at the vertices of the mesh, i.e., $d_i(u) = u(x_i)$, for all $x_i \in \mathcal{V}_h$. Note that the functionals d_h are bounded on X owing to the continuous embedding $X \subset \mathcal{C}^0(\overline{\Omega})$; see [23]. They can also be averages around the vertices, and again averages define bounded functionals on X.

In order to analyze the convergence properties of our reconstruction method, we assume that there exists a low-order approximation linear operator $\Pi_h : X \longrightarrow X_h$ such that the following hold for all $v \in X$:

$$\sum_{T\in\mathcal{T}_h} h_T^{j-2}|v-\Pi_h v|_{W^{j,1}(T)} \le c|\nabla v|_{\mathrm{BV}}, \qquad j=0,1, \tag{2.4}$$

$$|\nabla \Pi_h v|_{\mathrm{BV}} \le c|\nabla v|_{\mathrm{BV}}, \tag{2.5}$$

with c independent of h. We also assume that

$$d_h(\Pi_h(v)) = d_h(v) \qquad \forall v \in X, \tag{2.6}$$

$$d_h|_{\Pi_h X} \text{ is injective.} \tag{2.7}$$

Note that these two conditions imply that Π_h is a projector on $\Pi_h X \subsetneq X_h$. The existence of Π_h ensures that for any given sequence of measurements, it is possible to reconstruct a low-order stable convergent approximation. From now on, c denotes a generic constant whose generic value may change from place to place, and this constant is always independent of the mesh-size h.

Remark 2.2. In the case of point value measurements, say, $d_i(u) = u(x_i)$, $x_i \in \mathcal{V}_h$, it can be verified that the Lagrange $\mathbb{P}_1/\mathbb{Q}_1$ interpolant on the mesh $\mathcal{T}_h$ satisfies the assumptions (2.4)–(2.7). For instance, (2.4) is a well-known estimate for any $v \in W^{2,1}(\Omega)$, and by density of $W^{2,1}(\Omega)$ in X (see [23]) it follows for any $v \in X$. One can prove (2.5) for any $v \in W^{2,1}(\Omega)$ using standard scaling arguments, trace results, and the norm equivalence (2.9). The result for any $v \in X$ follows by density. The remaining properties (2.6)–(2.7) are true for the Lagrange interpolant.

2.3. The semidiscrete functional. At this point we consider two options: either we interpolate the constraints $\varpi_h = d_h(u)$ or we relax those constraints by incorporating them into the discrete problem.

2.3.1. First option: Interpolation. We introduce the affine set of functions $Y_h \subset X_h$ that interpolate the data

$$Y_h = \left\{ v_h \in X_h : d_i(v_h) = \varpi_i \ \forall i \in \overline{1, I_h} \right\}. \tag{2.8}$$

Note that when ϖ_h is such that there exists a function $u \in X$ so that $\varpi_h = d_h(u)$, then $\Pi_h(u) \in Y_h$, which means that $Y_h \neq \emptyset$. However, it may happen that $Y_h = \emptyset$ if the measurements are incompatible; for instance, specifying two different point values at the same mesh vertex is not solvable.

Our goal now is to find a function in Y_h that oscillates as little as possible. The main idea that we are pursuing is to minimize the seminorm $|\nabla \cdot|_{BV}$ over Y_h. (Note that $|\nabla \cdot|_{BV}$ is an extension to X of the L^1-norm of the Hessian for smooth functions.) It can be shown that for every function $v \in X_h$ the total measure of the Hessian of v is equivalent to

$$|\nabla v|_{\mathrm{BV}} \approx \sum_{T\in\mathcal{T}_h} \int_T (|v_{xx}| + 2|v_{xy}| + |v_{yy}|) + \sum_{F\in\mathcal{F}_h^i} \int_F |[\![\partial_{\boldsymbol{n}} v]\!]| \,. \tag{2.9}$$

In order to give ourselves more flexibility across edges, we introduce a parameter $\alpha > 0$ and consider the functional

$$\widetilde{J}_h(v) = \sum_{T\in\mathcal{T}_h} \int_T (|v_{xx}| + 2|v_{xy}| + |v_{yy}|) + \alpha \sum_{F\in\mathcal{F}_h^i} \int_F |[\![\partial_{\boldsymbol{n}} v]\!]| \,, \quad v \in X_h, \tag{2.10}$$

which induces a seminorm equivalent to the total variation of the ∇v. Note that $\widetilde{J}_h$ defines a seminorm which vanishes if and only if its argument is a linear polynomial over Ω.

2.3.2. Second option: Relaxation. Since interpolating the measurements may be impossible ($Y_h = \emptyset$) or not desirable, we relax the problem by setting

$$Y_h = X_h \tag{2.11}$$

and adding the constraints $\varpi_h = d_h(u)$ to the functional

$$\widetilde{J}_h(v) = \sum_{T\in\mathcal{T}_h} \int_T (|v_{xx}| + 2|v_{xy}| + |v_{yy}|) + \alpha \sum_{F\in\mathcal{F}_h^i} \int_F |[\![\partial_{\boldsymbol{n}} v]\!]| + \beta \sum_{i=1}^{I_h} |d_i(v) - \varpi_i|, \quad v \in X_h, \tag{2.12}$$

where the additional term in the functional is the ℓ_1-norm of the difference between $(d_1(v), \dots, d_I(v))$ and $(\varpi_1, \dots, \varpi_I)$ and the parameters $\alpha > 0$, $\beta > 0$ are user-defined.

In the rest of the paper we use definition (2.12) with either $\beta = 0$ when Y_h is defined by (2.8) or $\beta > 0$ when Y_h is defined by (2.11). Moreover, we assume that $Y_h \neq \emptyset$ when Y_h is defined by (2.8).

2.4. The semidiscrete minimization problem. The data reconstruction problem is formulated as follows: Find $u_h \in Y_h$ such that

$$\widetilde{J}_h(u_h) = \min_{v\in Y_h} \widetilde{J}_h(v). \tag{2.13}$$

In order to ensure solvability of this problem, we assume that the set of measurements d_h is such that

$$\forall v \in \mathbb{P}_1(\Omega) \quad (d_h(v) = 0) \Rightarrow (v = 0). \tag{2.14}$$

PROPOSITION 2.1. *The solution set of problem* (2.13) *is not empty provided that* (2.14) *holds.*

Proof. Since $Y_h \neq \emptyset$, let u_h be a fixed element in Y_h and define $S = \{v \in Y_h : \widetilde{J}_h(v) \le \widetilde{J}_h(u_h)\}$. Owing to (2.14), $\|v\|_h := |\nabla v|_{\mathrm{BV}} + |d_h(v)|_{\ell_1}$ is a norm on X_h. For all $v \in S$ we have

$$\begin{aligned}\|v\|_h &\le |\nabla v|_{\mathrm{BV}} + |d_h(v) - \varpi_h|_{\ell_1} + |\varpi_h|_{\ell_1} \\ &\le c\,\widetilde{J}_h(v) + |\varpi_h|_{\ell_1} \le c\,\widetilde{J}_h(u_h) + |\varpi_h|_{\ell_1},\end{aligned}$$

meaning that S is bounded. This implies the proposition since $\widetilde{J}_h$ is continuous. ☐

Since computing a minimizer for (2.13) may be a difficult task, we introduce the notion of almost minimizer.

DEFINITION 2.1. *We say that a sequence $\{u_h\}_{h>0}$, with $u_h \in Y_h$, is a sequence of almost minimizers if there is a constant $c_a \ge 1$, uniform with respect to h, such that*

$$\widetilde{J}_h(u_h) \le c_a \min_{v\in Y_h} \widetilde{J}_h(v) \quad \forall h > 0. \tag{2.15}$$

The following result clarifies the approximation properties of (2.13).

PROPOSITION 2.2. *Assume that Y_h is defined by* (2.8). *Assume that there is $u \in X$ such that $d_h(u) = \varpi_h$ for all $h > 0$ and that there exists Π_h satisfying* (2.4)–(2.7). *Let $\{u_h\}_{h>0}$ be a sequence of almost minimizers; then the following error estimates hold:*

$$\sum_{T\in\mathcal{T}_h} h_T^{j-2}|u-u_h|_{j,1,T} \le c|\nabla u|_{\mathrm{BV}(\Omega)}, \qquad j=0,1.$$

Proof. Since u_h is a member of Y_h, $d_h(u_h) = \varpi_h$; moreover, (2.6) implies $d_h(\Pi_h(u_h)) = d_h(u_h)$ and $d_h(\Pi_h(u)) = \varpi_h$. Since the restriction of d_h to $\Pi_h X$ is injective, we deduce that $\Pi_h(u_h) = \Pi_h(u)$. This immediately implies

$$\begin{aligned}\sum_{T\in\mathcal{T}_h} h_T^{j-2}|u-u_h|_{j,1,T} &\le \sum_{T\in\mathcal{T}_h} h_T^{j-2}(|u-\Pi_h(u)|_{j,1,T} + |\Pi_h(u)-u_h|_{j,1,T})\\ &\le c|\nabla u|_{\mathrm{BV}} + \sum_{T\in\mathcal{T}_h} h_T^{j-2}|\Pi_h(u_h)-u_h|_{j,1,T}\\ &\le c(|\nabla u|_{\mathrm{BV}} + |\nabla u_h|_{\mathrm{BV}}).\end{aligned}$$

Since u_h is an almost minimizer, we have

$$\begin{aligned}|\nabla u_h|_{\mathrm{BV}} &\le c\tilde{J}_h(u_h) \le cc_a \min_{v_h\in Y_h} \tilde{J}_h(v_h) \le cc_a\tilde{J}_h(\Pi_h(u))\\ &\le c_1|\nabla\Pi_h(u)|_{\mathrm{BV}} \le c_2|\nabla u|_{\mathrm{BV}}.\end{aligned}$$

The conclusion follows readily. ∎

Remark 2.3. When Y_h is defined using (2.11), we do not have a convergence proof yet, but we expect that convergence can be obtained by using the fact that $\sum_{i=1}^{I_h} |d_i(v) - \varpi_i|$ is bounded as $h \to 0$ combined with a Chebyshev inequality argument.

2.5. Quadratures. The evaluation of the functional $\widetilde{J}_h$ involves the computation of integrals with integrands that are absolute values. Therefore, we discretize $\widetilde{J}_h$ by replacing the integrals with quadrature rules. More specifically, the terms of the functional $\widetilde{J}_h$ are approximated using quadrature rules as follows:

$$\int_S |\mathcal{L}v| \approx \sum_{(p,\omega)\in I(S,\mathcal{L})} \omega|\mathcal{L}(v)(p)|,$$

where either $S \in \mathcal{T}_h$ and $\mathcal{L}$ is one of the linear operators $\{\partial_{xx}, 2\partial_{xy}, \partial_{yy}\}$ or $S \in \mathcal{F}_h^i$ and $\mathcal{L} = \alpha[\![\partial_{\boldsymbol{n}}]\!]$. For each pair $(S,\mathcal{L})$ the set $I(S,\mathcal{L})$ is composed of pairs (p,ω) of points $p \in \mathbb{R}^2$ and weights $\omega > 0$.

We require that the integration rules $I(S,\mathcal{L})$ satisfy the following two conditions:

1. Be exact when the sign of the integrant $\mathcal{L}v$ does not change over S.
2. Give an approximation that is equivalent to the exact integral; i.e., there are constants c_1, c_2 independent of S, $\mathcal{L}$, and h such that

$$c_1\int_S |\mathcal{L}v| \le \sum_{(p,\omega)\in I(S,\mathcal{L})} \omega|\mathcal{L}(v)(p)| \le c_2\int_S |\mathcal{L}v| \qquad \forall v \in X_h. \tag{2.16}$$

In general, the second condition requires the use of integration rules with more points than are required by the first one. For example, if T is a triangle and $\mathcal{L} = \partial_{xx}$, then v_{xx} is linear and the midpoint rule satisfies the first condition but not the second. The following proposition gives a natural construction of quadrature rules satisfying both the conditions above under an easily verifiable assumption.

PROPOSITION 2.3. *Let $\widehat{S}$ be a (closed) reference element (e.g., triangle, square, or segment), and let T be an invertible affine transformation mapping $\widehat{S}$ to S. Also, let $\widehat{\mathcal{P}}$ be a finite-dimensional subspace of $\mathcal{C}^0(\widehat{S})$ (e.g., polynomials) and $\mathcal{P} = \mathbb{T}\widehat{\mathcal{P}}$ be its image under the transformation $\mathbb{T} : \mathcal{C}^0(\widehat{S}) \longrightarrow \mathcal{C}^0(S)$ defined by*

$$v(x) := \mathbb{T}(\hat{v})(x) = \hat{v}(T^{-1}(x)) \qquad \forall x \in S.$$

Let $\hat{I} = \{(\hat{p}_i, \hat{\omega}_i)\}_{i=1}^n$ be an integration rule with positive weights on $\widehat{S}$. If $\hat{I}$ is exact for every function in $\widehat{\mathcal{P}}$ and the quadrature points are such that

$$\left[\hat{v} \in \widehat{\mathcal{P}} \quad \text{and} \quad \hat{v}(\hat{p}_i) = 0, \quad i = 1, \dots, n\right] \qquad \text{implies} \qquad \left[\hat{v}(\hat{x}) = 0 \quad \forall \hat{x} \in \widehat{S}\right], \tag{2.17}$$

then the integration rule $I = \{(p_i, \omega_i)\}_{i=1}^n$ with $p_i = T(\hat{p}_i)$ and $\omega_i = \frac{|S|}{|\widehat{S}|}\hat{\omega}_i$ (where $|\cdot|$ denotes the measure of the corresponding set) is exact for every function in $\mathcal{P}$ and

$$c_1 \int_S |v| \le \sum_{i=1}^n \omega_i |v(p_i)| \le c_2 \int_S |v| \qquad \forall v \in \mathcal{P}, \tag{2.18}$$

with constants $c_2 > c_1 > 0$ that depend on $\widehat{S}$ and $\widehat{P}$ but do not depend on the transformation T.

Proof. Since T is affine and invertible, for any $\hat{v} \in \mathcal{C}^0(\widehat{S})$ and $v = \mathbb{T}\,\hat{v}$ we have

$$\int_{\widehat{S}} \hat{v} = \frac{|\widehat{S}|}{|S|} \int_S v. \tag{2.19}$$

Thus, in particular, for any $v \in \mathcal{P}$ we have $\hat{v} = \mathbb{T}^{-1} v \in \widehat{\mathcal{P}}$ and

$$\int_S v = \frac{|S|}{|\widehat{S}|} \int_{\widehat{S}} \hat{v} = \frac{|S|}{|\widehat{S}|} \sum_{i=1}^n \hat{\omega}_i \hat{v}(\hat{p}_i) = \sum_{i=1}^n \omega_i v(p_i);$$

that is, the integration rule I is exact for v. The assumption about the quadrature points of $\hat{I}$ implies that the mapping $\hat{v} \to \sum_{i=1}^n \hat{\omega}_i |\hat{v}(\hat{p}_i)|$ is a norm on the finite-dimensional space $\widehat{\mathcal{P}}$, and therefore it is equivalent to the norm in $L^1(\widehat{S})$:

$$c_1 \int_{\widehat{S}} |\hat{v}| \le \sum_{i=1}^n \hat{\omega}_i |\hat{v}(\hat{p}_i)| \le c_2 \int_{\widehat{S}} |\hat{v}| \qquad \forall \hat{v} \in \widehat{\mathcal{P}}.$$

Multiplying these inequalities by $|S|/|\widehat{S}|$ and using (2.19) and the definitions of p_i and ω_i completes the proof. ◻

Based on the above proposition we use the following quadrature rules:

- When $S \in \mathcal{T}_h$ is a triangle and $\mathcal{L} \in \{\partial_{xx}, 2\partial_{xy}, \partial_{yy}\}$, then $\mathcal{L}(X_h|_S) = \mathbb{P}_1 = \widehat{\mathcal{P}} = \mathcal{P}$, and therefore the 3-point quadrature rule using the midpoints of the sides of the triangle satisfies the conditions of the proposition (this rule is exact for $\mathbb{P}_2$).

- When $S \in \mathcal{T}_h$ is a rectangle with sides parallel to the coordinate axes, we use three different quadrature rules for the three different second derivatives. For $\mathcal{L} = \partial_{xx}$ we have $\mathcal{L}(X_h|_S) = \mathbb{Q}_{1,3} = \widehat{\mathcal{P}} = \mathcal{P}$, and therefore we could use the 2×4 tensor product Gaussian rule; however, numerical experiments show some undesired oscillations which can be avoided by using the 3×4 tensor product Gaussian rule. For $\mathcal{L} = 2\partial_{xy}$ we have $\mathcal{L}(X_h|_S) = \mathbb{Q}_{2,2} = \widehat{\mathcal{P}} = \mathcal{P}$, and we use the 3×3 tensor product Gaussian rule. For $\mathcal{L} = \partial_{yy}$, $\mathcal{L}(X_h|_S) = \mathbb{Q}_{3,1} = \widehat{\mathcal{P}} = \mathcal{P}$, and we use the 4×3 tensor product Gaussian rule.
- When $S \in \mathcal{T}_h$ is not a rectangle with sides parallel to the coordinate axes, we have $\widehat{\mathcal{P}} \neq \mathcal{P}$, and it is more convenient to replace the second derivatives in $\widetilde{J}_h$ by second derivatives in directions parallel to the sides of S. This case is not considered in the numerical experiments reported in this paper.
- When $S \in \mathcal{F}_h^i$ and $\mathcal{L} = \alpha[\![\partial_{\boldsymbol{n}}]\!]$ we have two cases: (1) S is the edge of two triangles, and (2) S is a side in a quadrilateral. $\mathcal{P}$ and $\widehat{\mathcal{P}}$ are composed of one-dimensional quadratic polynomials in the first case and cubic polynomials in the second case. Therefore, we use the 3-point Gaussian rule in the first case and the 4-point Gaussian rule in the second.

Using the above quadrature rules, we obtain the approximate functional

$$J_h(u) = \sum_{\substack{T\in\mathcal{T}_h \\ \mathcal{L}\in\{\partial_{xx},2\partial_{xy},\partial_{yy}\}}} \sum_{(p,\omega)\in I(T,\mathcal{L})} \omega|\mathcal{L}(u)(p)| + \alpha \sum_{F\in\mathcal{F}_h^i} \sum_{(p,\omega)\in I(F,[\![\partial_{\boldsymbol{n}}]\!])} \omega|[\![\partial_{\boldsymbol{n}} u]\!](p)| + \beta \sum_{i=1}^{I_h} |d_i(u) - \varpi_i|. \tag{2.20}$$

Note that J_h defines a seminorm on X_h which is equivalent to that induced by $\widetilde{J}_h$ with constants independent of h.

The fully discretized version of problem (2.13) is the following: Find $u_h \in Y_h$ such that

$$J_h(u_h) = \min_{v_h\in Y_h} J_h(v_h). \tag{2.21}$$

PROPOSITION 2.4. *Assume that Y_h is defined by* (2.8). *Assume that there is $u \in X$ such that $d_h(u) = \varpi_h$ for all $h > 0$ and that there exists Π_h satisfying* (2.4)–(2.7). *Let $(u_h)_{h>0}$ be a sequence of almost minimizers of problem* (2.21)*; then the following error estimates hold:*

$$\sum_{T\in\mathcal{T}_h} h_T^{j-2}|u-u_h|_{j,1,T} \leq c|\nabla u|_{\mathrm{BV}(\Omega)}, \qquad j = 0, 1.$$

Proof. The proof is similar to that of Proposition 2.2. The only technicality consists of controlling $|\nabla u_h|_{\mathrm{BV}}$. Using successively the lower and the upper bounds in (2.16), we infer that

$$\begin{aligned} c_1|\nabla u_h|_{\mathrm{BV}} &\leq c\, J_h(u_h) \leq c\, c_a \min_{v_h\in Y_h} J_h(v_h) \leq c\, c_a\, J_h(\Pi_h(u)) \\ &\leq c\,|\nabla \Pi_h(u)|_{\mathrm{BV}} \leq c\,|\nabla u|_{\mathrm{BV}}. \end{aligned}$$

This concludes the proof. □

PROPOSITION 2.5. *Assume that all elements of the mesh $\mathcal{T}_h$ are quadrilaterals and the functionals d_h are point evaluations at the vertices of the mesh.*

1. *If Y_h is defined by* (2.8), *then there exists $\bar{\alpha} > 0$ such that for all $\alpha > \bar{\alpha}$, every solution u_h to* (2.21) *is in $\mathcal{C}^1(\overline{\Omega})$.*
2. *If $Y_h = X_h$, $\beta > 0$, then we have the following:*
   (a) *For fixed α, there is a threshold value $\bar{\beta} = \bar{\beta}(\alpha)$ such that for all $\beta > \bar{\beta}$, every solution u_h to* (2.21) *interpolates the data.*
   (b) *For fixed β, there is a threshold value $\bar{\alpha} = \bar{\alpha}(\beta)$ such that for all $\alpha > \bar{\alpha}$, every solution u_h to* (2.21) *is in $\mathcal{C}^1(\overline{\Omega})$.*
   (c) *There exists $\gamma > 0$ such that for all α, β satisfying* $\min(\alpha, \beta) > \gamma$, *every solution u_h to* (2.21) *is in $\mathcal{C}^1(\overline{\Omega})$ and interpolates the data.*

In all cases, the threshold values may be chosen to be independent of the data, $\{\varpi_i\}$.

Proof. See the appendix. ∎

This proposition shows that by increasing the values of the parameters α and β, one can reconstruct $\mathcal{C}^1$ interpolants similar to the cubic L^1-splines developed in [13, 17, 29].

Remark 2.4. The value of $\bar{\alpha}$ in Proposition 2.5, part 1, is not a priori uniform with respect to the typical mesh-size h. However, numerical tests indicate that using $\alpha = 5$ guarantees $\mathcal{C}^1$-smoothness independently of h.

Remark 2.5. For triangular meshes, the statements of Proposition 2.5, parts 1 and 2(c), hold if and only if the data and the mesh are compatible in the sense that X_h contains at least one $\mathcal{C}^1$ interpolant of the data; parts 2(a) and 2(b) hold with no restrictions.

3. The interior-point algorithm.

In this section we focus on the algebraic formulation of the minimization problem (2.21), establish some properties of the minimizer, and describe an interior-point technique for computing an almost minimizer.

3.1. Matrix formulation.

Let $\{\phi_i\}_{i=1}^{\hat{n}}$ be a basis for X_h, let $\{(p_i, \omega_i)\}_{i=1}^{m}$ be an enumeration of all the quadrature points (and weights) in all the quadrature rules used in the discretization of $\widetilde{J}_h$, and let $\{\mathcal{L}_i\}_{i=1}^{m}$ be the collection of linear operators corresponding to the quadrature rule. Thus, the total number of quadrature points is given by

$$m = \sum_{\substack{T \in \mathcal{T}_h \\ \mathcal{L} \in \{\partial_{xx}, 2\partial_{xy}, \partial_{yy}\}}} \#\left(I(T, \mathcal{L})\right) + \sum_{F \in \mathcal{F}_h^i} \#\left(I(F, [\![\partial_{\boldsymbol{n}}]\!])\right), \tag{3.1}$$

where $\#(\cdot)$ denotes the cardinal number function. We denote

$$I(\mathcal{T}_h) := \{i : \mathcal{L}_i \in \{\partial_{xx}, 2\partial_{xy}, \partial_{yy}\}\} \quad \text{and} \quad I(\mathcal{F}_h^i) := \{i : \mathcal{L}_i = [\![\partial_{\boldsymbol{n}}]\!]\}. \tag{3.2}$$

3.1.1. Interpolation.

Let us assume first that $\beta = 0$; i.e., Y_h is defined by (2.8). The functional J_h can then be rewritten as

$$J_h(v) = |\widehat{A}x|_1 \qquad \text{where} \qquad x \in \mathbb{R}^{\hat{n}} : v = \sum_{i=1}^{\hat{n}} x_i \phi_i, \tag{3.3}$$

and the entries of the matrix $\widehat{A}$ are given by

$$\widehat{A}_{ij} = \begin{cases} \omega_i \mathcal{L}_i(\phi_j)(p_i), & i \in I(\mathcal{T}_h), \\ \alpha\omega_i \mathcal{L}_i(\phi_j)(p_i), & i \in I(\mathcal{F}_h^i), \end{cases} \qquad j = 1, \dots, \hat{n}. \tag{3.4}$$

Since the minimization is done in Y_h, the total number of degrees of freedom is not $\hat{n}$ but $n := \hat{n} - I_h$, provided that the measurements are linearly independent. We now explain how to reduce the constrained minimization problem into an unconstrained one similar to (3.5). The linear independence of the measurements implies that one can construct a basis $\{\phi_j\}_{j=1}^{\hat{n}}$ for X_h so that $d_i(\phi_j) = \delta_{ij}$ for all $i \in \overline{1, I_h}$ and $j \in \overline{1, \hat{n}}$, where δ_{ij} is the Kronecker symbol. This construction is trivial if $\{\phi_i\}_{i=1}^{\hat{n}}$ is a nodal basis, and the measurements assign point values at the vertices in $\mathcal{V}_h$. Let $\varpi \in \mathbb{R}^{I_h}$ be the vector representing the measurements $(\varpi_1, \dots, \varpi_{I_h})$. Then the coordinate vector of any function in Y_h is of the form $x = (\varpi, y)$ and $y \in \mathbb{R}^n$. The matrix $\widehat{A}$ can be written in 1×2 block form $\widehat{A} = (\widehat{A}_1 A)$, where $\widehat{A}_1$ is $m \times I_h$ and A is $m \times n$. Let $b = -\widehat{A}_1 \varpi$; then the discrete problem (2.21) simplifies as follows: Find $x \in \mathbb{R}^n$ such that

$$|Ax - b|_1 = \min_{y \in \mathbb{R}^n} |Ay - b|_1. \tag{3.5}$$

It can be shown that A is full rank owing to (2.14).

3.1.2. Relaxation. If β is not zero, then the minimization problem is unconstrained and has the same form as (3.5). In particular, we have $n = \hat{n}$ and

$$\widehat{A}_{ij} = \begin{cases} \omega_i \mathcal{L}_i(\phi_j)(p_i), & i \in I(\mathcal{T}_h), \\ \alpha\omega_i \mathcal{L}_i(\phi_j)(p_i), & i \in I(\mathcal{F}_h^i), \\ \beta d_{i-m}(\phi_j), & i = m+1, \dots, m+I_h, \end{cases} \qquad j = 1, \dots, \hat{n}, \tag{3.6}$$

and

$$b_i = \begin{cases} 0, & i = 0, \dots, m, \\ \beta\varpi_{i-m}, & i = m+1, \dots, m+I_h. \end{cases} \tag{3.7}$$

3.2. Discrete problem. In this section we study properties of ℓ_1-minimization problems of generic form (3.5). Let A be an $m \times n$ real matrix ($m > n$) and $b \in \mathbb{R}^m$. We define the Lagrangian

$$L(x, \lambda) = (b - Ax)^t \lambda, \qquad x \in \mathbb{R}^n, \ \lambda \in \mathbb{R}^m,$$

and the primal and dual functions f and g, respectively,

$$f(x) = \max_{\substack{\lambda \in \mathbb{R}^m \\ |\lambda|_\infty \le 1}} L(x, \lambda) = |b - Ax|_1,$$

$$g(\lambda) = \min_{x \in \mathbb{R}^n} L(x, \lambda) = \begin{cases} b^t \lambda, & A^t \lambda = 0, \\ -\infty, & A^t \lambda \neq 0. \end{cases}$$

It is clear that for all $x \in \mathbb{R}^n$ and all $\lambda \in \mathbb{R}^m$, $|\lambda|_\infty \le 1$ we have

$$f(x) \ge L(x, \lambda) \ge g(\lambda). \tag{3.8}$$

The primal problem is defined to be

$$\text{minimize } \ f(x) = |b - Ax|_1, \qquad x \in \mathbb{R}^n, \tag{3.9}$$

and the dual problem is defined to be

$$\begin{aligned} &\text{maximize } \ g(\lambda) = b^t \lambda, \\ &\text{subject to } \ \lambda \in \Lambda := \{\lambda \in \mathbb{R}^m : A^t \lambda = 0, \ |\lambda|_\infty \le 1\}. \end{aligned} \tag{3.10}$$

The set Λ is referred to as the dual feasible set.

3.3. Primal-dual interior-point method. We now describe an approach for solving the minimization problem (3.5). First, we reformulate (3.5) as a linear programming problem: Find $y \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$ such that

$$\text{minimize } y^t 1 = \sum_{i=1}^{m} y_i, \qquad \text{subject to } \begin{cases} y \geq b - Ax, \\ y \geq Ax - b. \end{cases} \tag{3.11}$$

We apply the primal-dual interior-point method described in [1, section 11.8.2] (see also [30]). Upon setting

$$\widetilde{A} = \begin{pmatrix} A & -I \\ -A & -I \end{pmatrix}, \qquad \tilde{b} = \begin{pmatrix} b \\ -b \end{pmatrix}, \qquad \tilde{x} = \begin{pmatrix} x \\ y \end{pmatrix}, \qquad z = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{matrix} \}\, n, \\ \}\, m, \end{matrix}$$

we rewrite (3.11) in a more compact form:

$$\text{minimize } \; z^t \tilde{x}, \qquad \text{subject to } \; \widetilde{A}\tilde{x} \leq \tilde{b}.$$

The dual of this problem can be formulated as follows: find $\tilde{\lambda} \in \mathbb{R}^{2m}$ solving

$$\text{maximize } \; -\tilde{b}^t \tilde{\lambda}, \qquad \text{subject to } \; \widetilde{A}^t \tilde{\lambda} + z = 0, \; \tilde{\lambda} \geq 0.$$

We observe that this dual problem is equivalent to (3.10) in the sense that λ solves (3.10) if and only if $\tilde{\lambda} := (\frac{1-\lambda}{2}, \frac{1+\lambda}{2})$ solves the problem above.

We now describe the primal-dual interior-point algorithm that we use to solve (3.11) (see Algorithm 1).

ALGORITHM 1. Interior-point method.

```
 1. input: $A$, $b$, $x$, $\lambda$; $\mu$, $\epsilon$;
 2. $r = b - Ax$;
 3. $a = (|r|_1 - r^t\lambda)/m$;   $y_i = |r_i| + a,\ i = 1, \dots, m$;
 4. while $(|r|_1 > (1+\epsilon)\, r^t\lambda)$ do
 5.     $t^{-1} = (y^t 1 - r^t\lambda)/(2m\mu)$;
 6.     $s_1 = y + r$;   $s_2 = y - r$;
 7.     $d_1 = (1-\lambda)/(2s_1)$;   $d_2 = (1+\lambda)/(2s_2)$;
 8.     $d = 4d_1 d_2/(d_1 + d_2)$;
 9.     $v = t^{-1}(s_2^{-1} - s_1^{-1}) + (d_2 - d_1)/(d_1 + d_2)[1 - t^{-1}(s_1^{-1} + s_2^{-1})]$;
10.     $w = A^t v$;
11.     $\Delta x = (A^t \operatorname{diag}(d) A)^{-1} w$;
12.     $v = A\Delta x$;
13.     $\Delta y = [-1 + t^{-1}(s_1^{-1} + s_2^{-1}) + (d_1 - d_2)v]/(d_1 + d_2)$;
14.     $\Delta\lambda = -\lambda + t^{-1}(s_2^{-1} - s_1^{-1}) - (d_1 + d_2)v + (d_1 - d_2)\Delta y$;
15.     $\sigma = \max\{\tau \in (0, 2] : -1 \leq \lambda + \tau\Delta\lambda \leq 1,$
                    $y + \tau\Delta y \geq r - \tau v,\ y + \tau\Delta y \geq -r + \tau v\}$
16.     $\sigma = \min\{1,\ 0.99\sigma\}$;
17.     $x = x + \sigma\Delta x$;   $y = y + \sigma\Delta y$;   $r = r - \sigma v$;   $\lambda = \lambda + \sigma\Delta\lambda$;
18. end while
19. output: $x$, $\lambda$;
```

The input parameter μ is a positive real number (we use $\mu = 10$), and ϵ is a given tolerance. The initial input value of the dual variable λ is assumed to be strictly dual feasible; that is, $A^t\lambda = 0$ and $|\lambda|_\infty < 1$ (we use $\lambda = 0$). In the algorithm, a, t, and σ are scalar variables; $r, y, d, v, \Delta y, \Delta\lambda \in \mathbb{R}^m$; $w, \Delta x \in \mathbb{R}^n$; and the vectors s_1, s_2, d_1, d_2 do not need to be stored since their components can be evaluated one by one when needed (one time when computing d and v, and another time when computing Δy and $\Delta\lambda$). All operations in the definitions of $d_1, d_2, d, v, \Delta y$, and $\Delta\lambda$ are componentwise.

This algorithm is a specialization of [1, Alg. 11.2] to our setting, and we now present the key steps in its derivation. The critical part of this iterative algorithm consists of computing the next primal-dual search direction, and this is done by considering the modified KKT equations for (3.11),

$$\begin{cases} r_{\text{dual}}(\tilde\lambda) := z + \widetilde{A}^t\tilde\lambda = 0, \\ r_{\text{cent}}(\tilde x, \tilde\lambda, t) := \operatorname{diag}(\tilde\lambda)(\tilde b - \widetilde{A}\tilde x) - t^{-1}1 = 0, \end{cases} \tag{3.12}$$

where $t > 0$ is a regularization parameter (see [1, section 11.2]). Then, given a strictly feasible pair $(\tilde x, \tilde\lambda)$ (i.e., $\tilde b - \widetilde{A}\tilde x > 0$ and $\tilde\lambda > 0$) and $t > 0$, the next primal-dual search direction $(\Delta\tilde x, \Delta\tilde\lambda)$ is obtained by applying one Newton step to (3.12):

$$\begin{pmatrix} 0 & \widetilde{A}^t \\ -\operatorname{diag}(\tilde\lambda)\widetilde{A} & \operatorname{diag}(\tilde b - \widetilde{A}\tilde x) \end{pmatrix} \begin{pmatrix} \Delta\tilde x \\ \Delta\tilde\lambda \end{pmatrix} = -\begin{pmatrix} r_{\text{dual}} \\ r_{\text{cent}} \end{pmatrix}. \tag{3.13}$$

To simplify the notation and when the context is such that no confusion is possible, instead of using $\operatorname{diag}(a)$ we simply use a. Since $\tilde x$ is strictly feasible (i.e., $\tilde b - \widetilde{A}\tilde x > 0$), we can express $\Delta\tilde\lambda$ from the second equation of (3.13) as

$$\Delta\tilde\lambda = s^{-1}(-r_{\text{cent}} + \tilde\lambda\widetilde{A}\Delta\tilde x),$$

where we have set $s = \tilde b - \widetilde{A}\tilde x$. Note that s can be written as $s = (s_1, s_2) = (r + y, -r + y)$ with $r = b - Ax$. Substituting $\Delta\tilde\lambda$ into the first equation of (3.13) and simplifying the right-hand side gives

$$\begin{aligned} \widetilde{A}^t s^{-1}\tilde\lambda\widetilde{A}\Delta\tilde x &= -r_{\text{dual}} + \widetilde{A}^t s^{-1} r_{\text{cent}} = -z - \widetilde{A}^t\tilde\lambda + \widetilde{A}^t s^{-1}(\tilde\lambda s - t^{-1}1) \\ &= -z - t^{-1}\widetilde{A}^t s^{-1}1. \end{aligned} \tag{3.14}$$

Now we want to express the two components of $\Delta\tilde x := (\Delta x, \Delta y)$. For this purpose we denote $d_i = s_i^{-1}\tilde\lambda_i$, $i = 1, 2$, $e_1 = d_1 + d_2$, and $e_2 = d_2 - d_1$ (recall that $\tilde\lambda_1 = \frac{1-\lambda}{2}$ and $\tilde\lambda_2 = \frac{1+\lambda}{2}$; see line 7 in Algorithm 1). Then the matrix $\widetilde{A}^t s^{-1}\tilde\lambda\widetilde{A}$ has the following block structure:

$$\widetilde{A}^t s^{-1}\tilde\lambda\widetilde{A} = \begin{pmatrix} A^t & -A^t \\ -I & -I \end{pmatrix}\begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix}\begin{pmatrix} A & -I \\ -A & -I \end{pmatrix} = \begin{pmatrix} A^t e_1 A & A^t e_2 \\ e_2 A & e_1 \end{pmatrix},$$

and the right-hand side in (3.14) can be rewritten as follows:

$$-z - t^{-1}\widetilde{A}^t s^{-1}1 = \begin{pmatrix} -t^{-1}A^t(s_1^{-1} - s_2^{-1})1 \\ -1 + t^{-1}(s_1^{-1} + s_2^{-1})1 \end{pmatrix} =: \begin{pmatrix} r_x \\ r_y \end{pmatrix}.$$

Then (3.14) can be recast into

$$\begin{pmatrix} A^t e_1 A & A^t e_2 \\ e_2 A & e_1 \end{pmatrix}\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_x \\ r_y \end{pmatrix}.$$

Since s and $\tilde{\lambda}$ are positive, the diagonal matrix $\operatorname{diag}(e_1)$ is invertible, and we infer that Δx and Δy are given by

$$A^t(e_1 - e_2 e_1^{-1} e_2) A \Delta x = r_x - A^t e_2 e_1^{-1} r_y, \tag{3.15}$$

$$\Delta y = e_1^{-1}(r_y - e_2 A \Delta x); \tag{3.16}$$

see lines 11 and 13 in Algorithm 1. Using the definitions above, we have

$$e_1 - e_2 e_1^{-1} e_2 = (e_1^2 - e_2^2) e_1^{-1} = 4 d_1 d_2 (d_1 + d_2)^{-1} =: d.$$

The right-hand side of the equation for Δx, (3.15), can be simplified to require only one multiplication by A^t:

$$\begin{aligned} r_x - A^t e_2 e_1^{-1} r_y = A^t \big[& t^{-1}(s_2^{-1} - s_1^{-1}) 1 \\ & + (d_2 - d_1)(d_1 + d_2)^{-1}\{1 - t^{-1}(s_1^{-1} + s_2^{-1}) 1\} \big] =: w. \end{aligned}$$

Finally, we can compute the increment $\Delta\tilde{\lambda}$:

$$\begin{aligned} \Delta\tilde{\lambda} &= s^{-1}(-r_{\text{cent}} + \tilde{\lambda}\widetilde{A}\Delta\tilde{x}) = s^{-1}(t^{-1}1 - \tilde{\lambda} s + \tilde{\lambda}\widetilde{A}\Delta\tilde{x}) \\ &= \begin{pmatrix} -\tilde{\lambda}_1 + t^{-1} s_1^{-1} 1 + d_1(A\Delta x - \Delta y) \\ -\tilde{\lambda}_2 + t^{-1} s_2^{-1} 1 - d_2(A\Delta x + \Delta y) \end{pmatrix} =: \begin{pmatrix} \Delta\tilde{\lambda}_1 \\ \Delta\tilde{\lambda}_2 \end{pmatrix}. \end{aligned}$$

Formally, the algorithm could stop here since we have defined the increments $\Delta\tilde{x}$ and $\Delta\tilde{\lambda}$. We further simplify the algorithm by using the property

$$\begin{aligned} \Delta\tilde{\lambda}_1 + \Delta\tilde{\lambda}_2 &= -\tilde{\lambda}_1 - \tilde{\lambda}_2 + t^{-1}(s_1^{-1} + s_2^{-1}) 1 + (d_1 - d_2) A\Delta x - (d_1 + d_2)\Delta y \\ &= -\tilde{\lambda}_1 - \tilde{\lambda}_2 + t^{-1}(s_1^{-1} + s_2^{-1}) 1 + (d_1 - d_2) A\Delta x \\ &\quad - [-1 + t^{-1}(s_1^{-1} + s_2^{-1}) 1 - (d_2 - d_1) A\Delta x] = 1 - \tilde{\lambda}_1 - \tilde{\lambda}_2, \end{aligned}$$

which holds even if Δx is not the exact solution of (3.15), provided that Δy is defined by (3.16). The equality $\Delta\tilde{\lambda}_1 + \Delta\tilde{\lambda}_2 = 1 - \tilde{\lambda}_1 - \tilde{\lambda}_2$ implies that if the initialization for $\tilde{\lambda}$ is such that $\tilde{\lambda}_1 + \tilde{\lambda}_2 = 1$, which is the second block equality in the constraint $\widetilde{A}^t\tilde{\lambda} + z = 0$, then it will still hold after the update $\tilde{\lambda} \leftarrow \tilde{\lambda} + \sigma\Delta\tilde{\lambda}$. As a result, instead of working with the pair $(\tilde{\lambda}_1, \tilde{\lambda}_2)$, we simplify the algorithm by working with $\lambda = \tilde{\lambda}_2 - \tilde{\lambda}_1$ only. The increment $\Delta\lambda$ is given by $\Delta\lambda := \Delta\tilde{\lambda}_2 - \Delta\tilde{\lambda}_1$ (see line 14 in Algorithm 1).

Note that the strict feasibility condition $\tilde{\lambda} > 0$ is equivalent to $|\lambda|_\infty < 1$ and the constraint $\widetilde{A}^t\tilde{\lambda} + z = 0$ is equivalent to $A^t\lambda = 0$. All the vectors λ and y generated by Algorithm 1 are strictly dual feasible ($A^t\lambda = 0$ and $|\lambda|_\infty < 1$, $|r| < y$) provided that the input λ is strictly dual feasible. To see this, observe that line 15 in Algorithm 1 implies that the new λ is such that $|\lambda|_\infty \le 1$ and line 16 implies that $|\lambda|_\infty < 1$. The same comment applies to y; i.e., the new vector y is larger than the absolute value of the new residual. Moreover, by construction one can verify from the first equation in (3.13) that $\Delta\lambda$ and λ are related by $A^t\Delta\lambda = -A^t\lambda$; as a result, if $A^t\lambda = 0$, then $A^t(\lambda + \sigma\Delta\lambda) = 0$.

The surrogate duality gap defined by $\hat{\eta} := (\tilde{b} - \widetilde{A}\tilde{x})^t\tilde{\lambda}$ should be zero at convergence. This quantity can also be rewritten as follows:

$$\hat{\eta} = s^t\tilde{\lambda} = s_1^t\tilde{\lambda}_1 + s_2^t\tilde{\lambda}_2 = y^t 1 - r^t\lambda.$$

This expression justifies the definition of the regularization parameter t in line 5 in Algorithm 1.

PROPOSITION 3.1. *The output vector x generated by Algorithm 1 is an almost minimizer for (3.5) with a tolerance $1 + \epsilon$.*

Proof. The stopping criterion used in the algorithm guarantees that

$$f(x) - f(x^*) \le f(x) - g(\lambda) = |r|_1 - r^t\lambda \le \epsilon\, r^t\lambda \le \epsilon f(x^*),$$

which proves the statement. $\square$

The most expensive step at each iteration of the **while** loop is the solution of (3.15) for Δx. Since direct solution methods are not practical for large n, we use an iterative method to solve the linear system approximately. In the resulting algorithm the vectors λ do not satisfy $A^t\lambda = 0$. However, numerically we observe that solving iteratively with relative tolerance $\epsilon/10$ produces results that are very similar to the results obtained by solving almost exactly. The iterative method we use is the preconditioned conjugate gradient (PCG) method with a simple symmetric Gauss–Seidel preconditioner.

Remark 3.1. It is not guaranteed that Algorithm 1 will produce an output in all cases in a reasonable amount of time. We observe that 20 to 50 interior-point steps are sufficient in our numerical examples for $\epsilon = 10^{-2}$, the linear system being solved with relative tolerance $\epsilon/10$, but to the best of our knowledge there is no theoretical result limiting the number of steps in general when the linear system (3.15) is not solved exactly. If the linear system (3.15) is solved exactly, using a direct method, we expect that Algorithm 1 will terminate in approximately $\sqrt{n}$ interior-point steps; see [20, 25] for convergence and complexity analysis of similar interior-point methods.

4. Surface reconstruction.

We illustrate our data reconstruction technique in this section. In all our numerical experiments, Ω is a square, and we use a uniform rectangular mesh with equal step size in both x and y directions. The tolerance in the interior-point (IP) method is $\epsilon = 10^{-2}$, and the linear systems for Δx are solved with relative tolerance 10^{-3}.

4.1. Piecewise smooth data.

The data for this set of experiments is obtained from point values of the function

$$u(x, y) = f(\max\{|x - 1/2|, |y - 1/2|\}), \qquad (x, y) \in \Omega := [0, 1]^2, \tag{4.1}$$

where

$$f(r) = \begin{cases} 5/3, & r \in [0, 1/8], \\ 1, & r \in (1/8, 5/16], \\ 16(1/2 - r)/3, & r \in (5/16, 1/2]. \end{cases} \tag{4.2}$$

Note that $u(x, y)$ is discontinuous at $\Gamma_1 = \{r = 1/8\}$, and its gradient also has jumps across $\Gamma_2 = \{r = 5/16\}$ and $\Gamma_3 = (\{x = y\} \cup \{x + y = 1\}) \cap \{5/16 \le r \le 1/2\}$. Away from those discontinuities the function is linear. The graph of u looks like an Aztec pyramid (see Figure 2).

We construct a uniform Cartesian mesh composed of $1/h \times 1/h$ square cells, and we define $\mathcal{V}_h = \{x_1, \ldots\}$ to be the $\mathbb{Q}_1$ vertices of this mesh. Our goal is to reconstruct a nonoscillatory approximation of u from pointwise values; i.e., we set $d_i(v) := v(x_i)$ for all $x_i \in \mathcal{V}_h$.

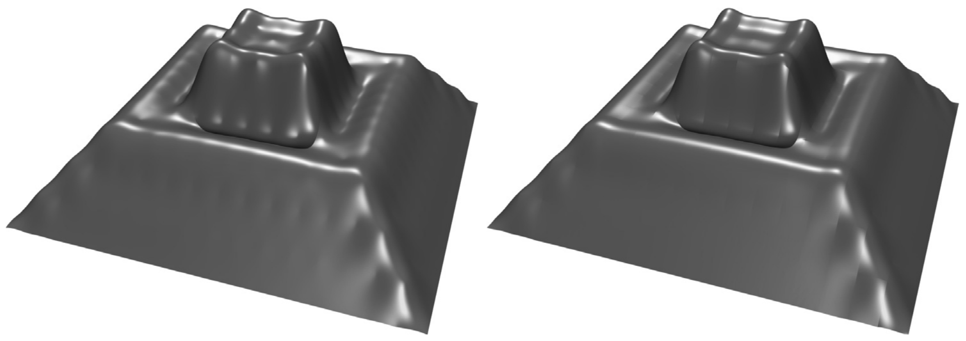

FIG. 1. *The pyramid test case, $h = 1/16$. Thin plate spline reconstruction (left) and ℓ_2-reconstruction (right).*

In the left panel of Figure 1 we show the graph obtained using the thin plate spline reconstruction [8, 27] (the thin plate spline technique is based on radial basis functions). In the right panel of Figure 1 we show the graph of the approximate solution obtained by using the discrete ℓ_2-norm instead of the discrete ℓ_1-norm, i.e., $u_h \leftarrow \mathrm{Argmin}_{y\in\mathbb{R}^n}|Ay-b|_{\ell_2}$, where A and b have been defined in section 3.1. The graph of the thin plate spline approximation and that obtained from the ℓ_2-minimization process are very similar and obviously oscillatory. This type of result motivates our interest in L^1-minimization techniques.

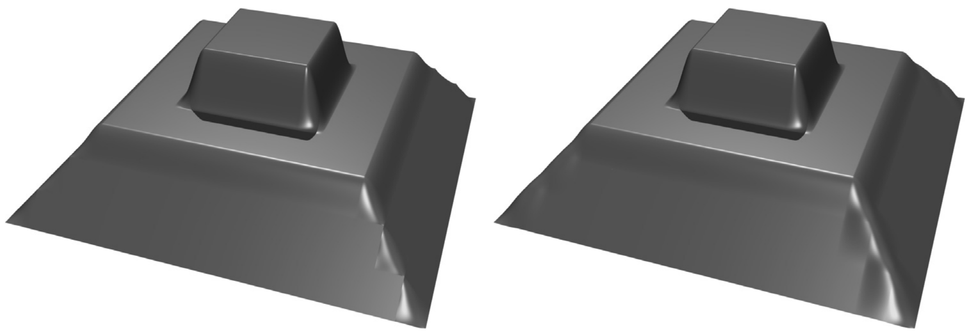

FIG. 2. *L^1-reconstruction of the graph of the pyramid test case, $h = 1/16$; $\alpha = 3$ (left) and $\alpha = 5$ (right).*

Figure 2 shows two reconstructed surfaces obtained by solving (2.21) with exact interpolation ($\beta = 0$) using $\alpha = 3$ (left panel) and $\alpha = 5$ (right panel) on the 16×16 mesh. The solution obtained with $\alpha = 5$ is $\mathcal{C}^1$ everywhere, and that obtained with $\alpha = 3$ is $\mathcal{C}^1$ almost everywhere but around the edges defined by Γ_3. The improvement over the results shown in Figure 1 is clear.

Figure 3 shows the isocontours of the two L^1-reconstructed solutions shown in Figure 2. The result in the left panel has been computed using $\alpha = 3$. The solution is not $\mathcal{C}^1$, but the level sets seem to be convex. The result shown in the right panel has been computed using $\alpha = 5$. We have checked that the solution is $\mathcal{C}^1$ by verifying that the constraints on the jump of the normal derivative across all the mesh cells are satisfied. We observe that the price to pay for $\mathcal{C}^1$-continuity is the introduction of oscillations and the loss of convexity of the level sets.

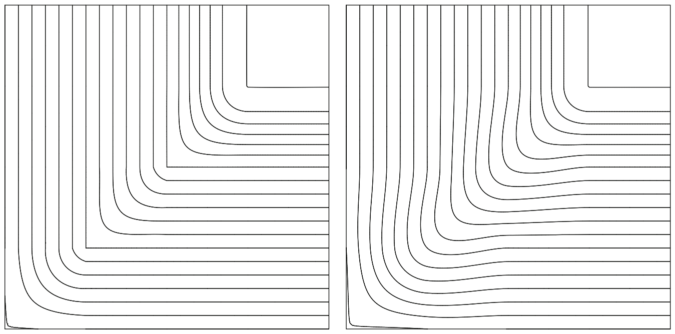

FIG. 3. *Isocontours of the L^1-reconstruction of the pyramid test case, $h = 1/16$; zoom on the bottom left corner; $\alpha = 3$ (left) and $\alpha = 5$ (right).*

In order to recover some convexity on the isocontours, we now relax the interpolation constraint by solving (2.21) with $Y_h = X_h$ and using $\beta = 7$ in (2.20). The results are shown in Figure 4. The graph of the approximate solution is shown in the left panel, and the isocontours are shown in the right panel. We observe that the isocontours are now approximately convex. We have verified numerically that the solution is $\mathcal{C}^1$ everywhere; increasing the parameter α does not change the output since the solution is $\mathcal{C}^1$.

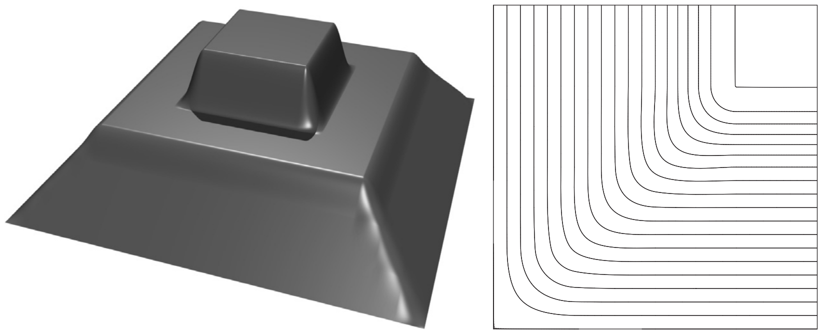

FIG. 4. *L^1-reconstruction of the graph of the pyramid test case with the interpolation constraint relaxed, using $\alpha = 3$ and $\beta = 7$, $h = 1/16$; 3D view (left) and isocontours zoom on the bottom left corner (right).*

In Table 1 (left side), we present results illustrating the convergence properties of the IP method as the mesh is refined. We performed tests with mesh-sizes $h = 1/16$, $1/32$, $1/64$, $1/128$, and $1/256$ using exact interpolation at all mesh vertices ($\beta = 0$). For each mesh we report the number of IP iterations to reach the tolerance $\epsilon = 10^{-2}$. We observe that the number of IP iterations increases like $\ln(1/h)$. We also report in this table the total number of PCG iterations, the linear system being solved with relative tolerance 10^{-3}. For two consecutive meshes we compute the ratio of the number of PCG iterations from one level to the next, and we multiply this ratio by 4, which roughly measures the increase in the computational cost per level. When we compare these ratios with the actual increase in computing time (last line in the table), we observe that both ratios are fairly close. These numbers show that the computational complexity of the algorithm (and the CPU time) scales like $\mathcal{O}(n^\gamma)$ with $\gamma = \ln(6)/\ln(4) \approx 1.29$.

TABLE 1
Convergence tests for the pyramid test case with $\alpha = 3$.

$1/h$	16	32	64	128	256
IP iter.	15	16	17	18	19
$\hat{n}$	2 401	9 409	37 249	148 225	591 361
m	10 368	41 728	167 424	670 720	2 684 928
PCG iter.	1 512	1 599	2 437	3 628	4 751
Ratio×4	—	4.23	6.10	5.95	5.24
Time, sec.	4.62	21.22	126.68	754.37	3 908.27
Ratio	—	4.59	5.97	5.95	5.18

4.2. Real terrain data. Next, we present results for two data sets coming from elevation maps of real terrains.

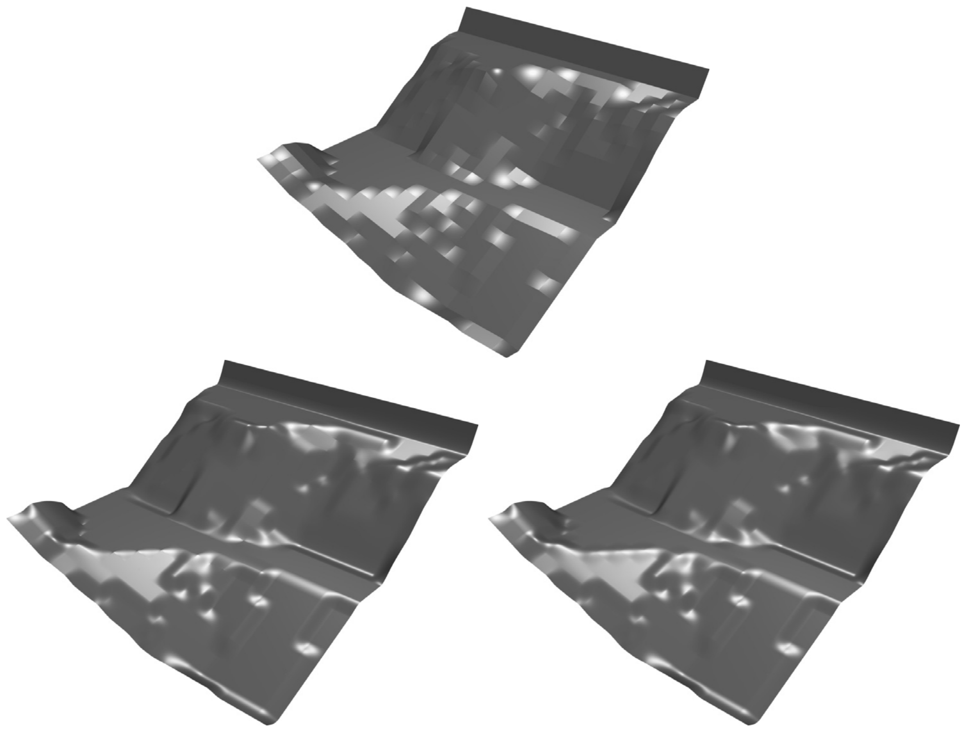

FIG. 5. *Small terrain data set. $\mathbb{Q}_1$ interpolant (top); L^1-reconstruction using $\alpha = 3$, $\beta = 0$ (left) and $\alpha = 5$, $\beta = 0$ (right).*

The first test case (hereafter called the small terrain data set) is a reference test from [28]. The domain is the unit square, $\Omega = (0,1)^2$, and terrain elevations are given at the vertices of a uniform 20×20 Cartesian grid covering Ω. Denoting by $\mathcal{V}_h$ the $\mathbb{Q}_1$ vertices of the mesh, our objective is to reconstruct the terrain topography using the interpolation constraints $d_i(v) = v(x_i)$ for all x_i in $\mathcal{V}_h$. The $\mathbb{Q}_1$ interpolant of the data set is shown in Figure 5.

The results using penalty parameters $\alpha = 3$ and $\alpha = 5$ (and $\beta = 0$ for both) are shown in Figure 5. It is clear that both reconstructions are significantly smoother than the $\mathbb{Q}_1$ interpolant. We have verified numerically that the solution computed with $\alpha = 5$ is $\mathcal{C}^1$. By looking at fine details it seems that the solution computed with $\alpha = 3$ is slightly more monotone than the $\mathcal{C}^1$ solution and is thus more eye-pleasing.

For the second test case we use the elevation map of a 3km×3km terrain near Barton Creek in Austin, Texas. The data comes from the Digital Elevation Models (DEM) data files produced by the U.S. Geological Survey (USGS).[1] The data set is sampled on a 100×100 uniform Cartesian mesh. Denoting again by $\mathcal{V}_h$ the $\mathbb{Q}_1$ vertices of the mesh, we reconstruct the topography using the interpolation constraints $d_i(v) = v(x_i)$ for all x_i in $\mathcal{V}_h$. The $\mathbb{Q}_1$ interpolant of the data and the reconstructed $\mathbb{Q}_3$ solution using $\alpha = 3$ (and $\beta = 0$) are shown in Figure 6 (two top panels). So that the reader may better appreciate the quality of the reconstruction, we show in the two bottom panels of Figure 6 a zoom of a small region.

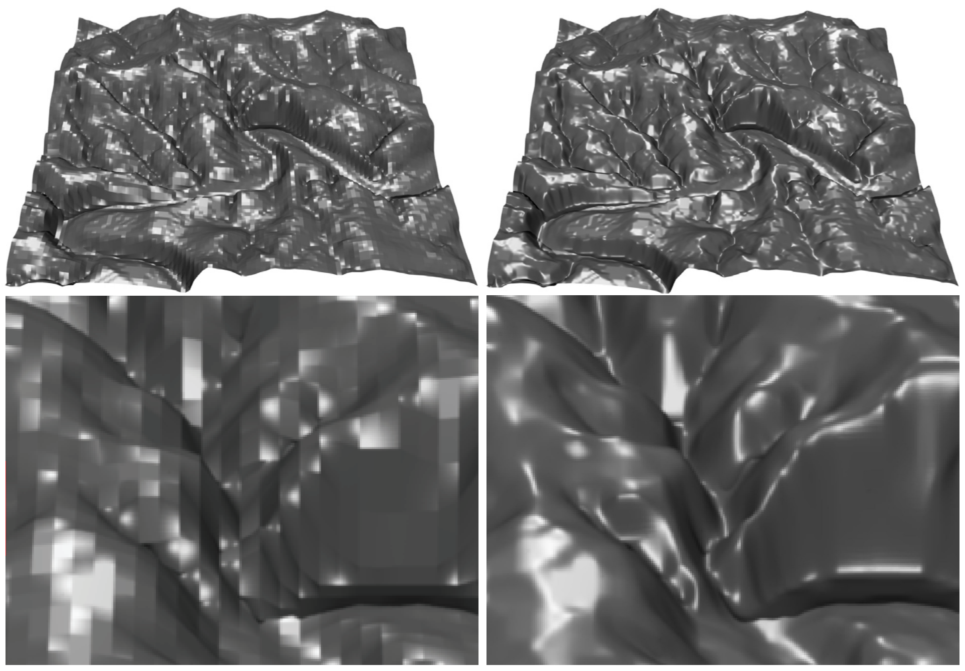

FIG. 6. *Barton Creek data set.* $\mathbb{Q}_1$ *interpolant (top left);* L^1*-reconstruction using* $\alpha = 3$ *(top right); zoom of the* $\mathbb{Q}_1$ *interpolant (bottom left); zoom of the* L^1*-reconstruction using* $\alpha = 3$ *(bottom right).*

In Table 2, we present the computational characteristics of the IP method applied to the small terrain data set and the Barton Creek case. The computation has been done on a desktop, and no particular attention has been given to the optimization of the code. The complexity we report in Tables 1 and 2 seems to outperform that reported in Table 1 in [26]. The complexity of the method seems to be $\mathcal{O}(n^\gamma)$, with $\gamma \approx 1.29$.

5. Image enhancing. We now apply the reconstruction technique to image processing. Given a gray-scale image, our goal is to enhance the resolution of under-resolved or aliased gray-scale images.

[1]DEM data is available at http://www.webgis.com/terraindata.html.

Table 2
Convergence tests for the small terrain (20×20) *and Barton creek* (100×100) *tests with* $\alpha = 3$.

$1/h$	20×20	100×100
IP iter.	21	28
$\hat{n}$	3 721	90 601
m	16 240	409 200
PCG iter.	4 656	3 923
Time, sec.	21.78	498.55

5.1. The principle. We represent digital gray-scale pictures on uniform Cartesian grids as piecewise constant functions, each square in the mesh representing a pixel and the value therein ϖ_i being the light intensity.

To reconstruct a better image, we can define the constraint functionals d_h in various ways. One possible way, which we call the $s0$ technique, consists of setting $d_i(v) = v(x_i)$ for all $x_i \in \mathcal{B}_h$, where $\mathcal{B}_h$ is the set of barycenters of the mesh elements. These constraints impose the original light intensity at the barycenter of each pixel for the reconstructed image. Another possible way, which we call the $s1$ technique, consists of setting $d_i(v) = \frac{1}{|T|}\int_T v(x)\,\mathrm{d}x$ for all $T \in \mathcal{T}_h$. With these constraints we impose the average light intensity over each pixel of the reconstructed image to be equal to that of the original image.

5.2. Target test case. We start with a test from [26] which we call the "target test." In this test, we take a low resolution aliased image (see the top panel in Figure 7) and use the $s0$ and $s1$ reconstruction techniques to enhance the image. The results are shown in Figure 7.

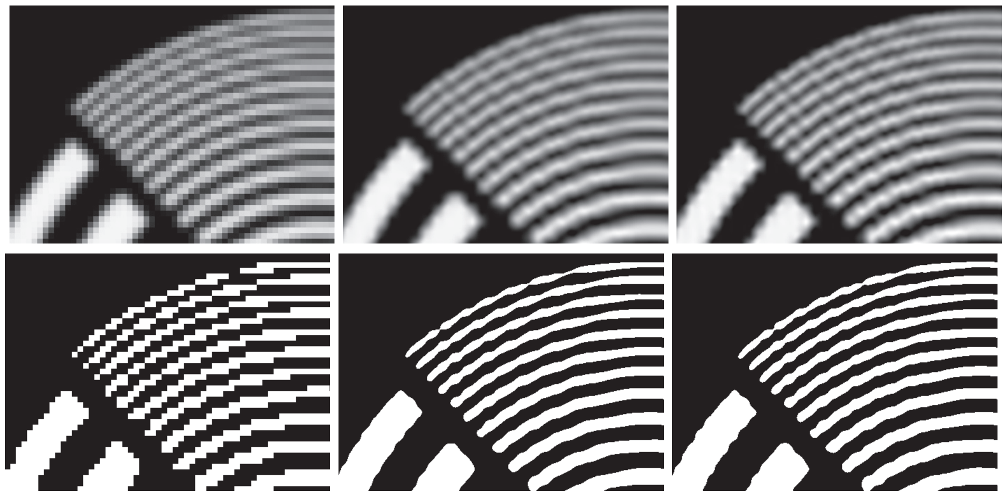

Fig. 7. *Target test: Original image (top left);* $s0$ *reconstruction (top center);* $s1$ *reconstruction (top right). BW thresholding: Original image (bottom left);* $s0$ *reconstruction (bottom center);* $s1$ *reconstruction (bottom right).*

We observe that both the $s0$ and $s1$ reconstructions reduce the aliasing effect with maybe the $s1$ technique producing slightly sharper edges. For the reader who is familiar with superresolution techniques let us stress that, contrary to what is done in the superresolution community, we use only one original image. Our technique cannot be compared to other superresolution techniques which use ten or more samples of the same image to remove the aliasing.

In the second row of Figure 7 we show all images in black and white. The gray-scale goes from 0 to 255, and we have set the black/white threshold at 127. These pictures show that our technique reconstructs the level sets well.

5.3. Text test case. Our second image enhancing test case comes from [21].[2] It is an aliased picture of the word "ČESKOSLOVENSKO." The original image is shown in the top panel in Figure 8. Again, our approach consists of enhancing the image using the reconstruction techniques described above.

The results obtained using the $s0$ and $s1$ reconstructions are shown in the panels in the second row in Figure 8. The improvement is clear, the $s1$ reconstruction being slightly sharper than the $s0$ reconstruction. The dealiasing effect is a little less dramatic than in [21], but recall again that we are using only one original image for the reconstruction.

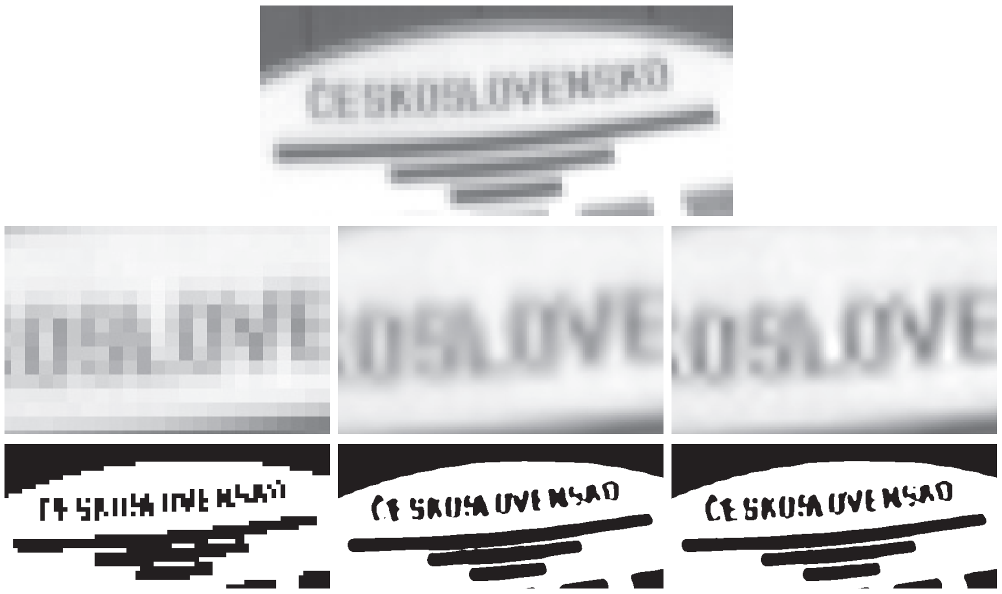

FIG. 8. *ČESKOSLOVENSKO test case. Original picture (top). Detailed region of the central letters: Original picture (second row left); $s0$ reconstruction (second row center); $s1$ reconstruction (second row right). BW thresholding: Original picture (third row left); $s0$ reconstruction (third row center); $s1$ reconstruction (third row right). The original image was first published in The Computer Journal (L. C. Pickup, D. P. Capel, S. J. Roberts, and A. Zisserman, "Bayesian methods for image super-resolution," The Computer Journal,* 52 (1) (2009)*) and appears here with permission of Oxford University Press.*

In the third row of Figure 8 we show all images in black and white. The gray-scale goes from 0 to 255, and we have set the black/white threshold at 186 for the three pictures. These images show that our technique reconstructs the level sets well, with the $s1$ reconstruction being the best again.

5.4. Pepper test case. We now show how our L^1-reconstruction method works on a pepper image. The original image is shown in the top panel and bottom left

[2]A full sequence of superresolution tests is available at http://www.robots.ox.ac.uk/~vgg/data.

panel in Figure 9. The result obtained using the $s1$ reconstruction is shown in the right bottom panels in Figure 9. The improvement is clear again.

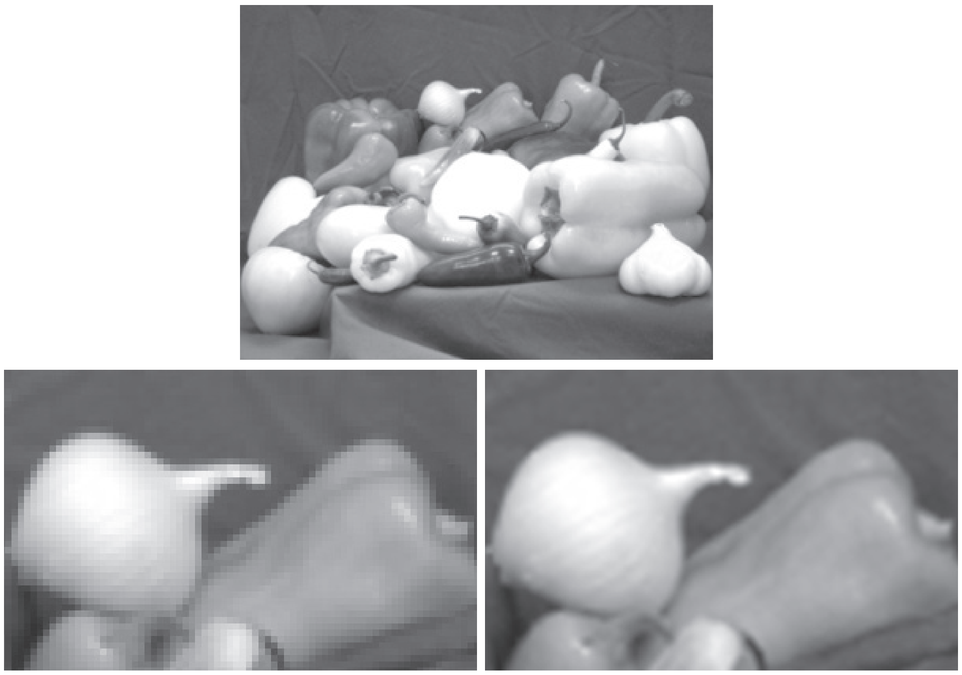

FIG. 9. *Pepper test case. Original picture (top and bottom left);* $s1$ *reconstruction (bottom right).*

5.5. Lena test. We finish this series on image enhancing by reconstructing a down-sampled version of the standard test image of Lena. Similarly to [19], we down-sample the 512×512 gray-scale original image to a 128×128 image by averaging 4×4 pixel blocks. The result obtained using the $s1$ reconstruction is shown in Figure 10. Compared to the results in [19, section 5.1], our reconstruction does not have the same sharpness at the edges; however, the staircase effect is substantially reduced as a result of the algorithm smoothing the boundary of the level sets. The staircase effect is a typical problem of the BV-regularized reconstruction methods. Regularizing the BV-norm of the gradient, as advocated in the present paper, is a way to avoid this problem.

6. Conclusion. As claimed by Lavery [16, 18], we have observed that the L^1-metric is suitable for nonoscillatory surface reconstruction. We have proposed a finite element technique which is more flexible than L^1-cubic splines for this purpose. We have proposed a preconditioned interior-point technique for solving the minimization problem whose complexity is $\mathcal{O}(n^\gamma)$, with $\gamma \approx 1.29$ in our numerical experiments. In general, one should expect that the number of interior-point steps scales like $\sqrt{n}$ (see [20, 25] for details), but in our examples we observe only a logarithmic growth. The observed cost reduction is based on the use of the PCG method when solving the linear system (3.15) instead of a direct method. For instance, the method in [28] scales like n^2 per interior-point step, and the method in [13] scales like n^3 per interior-point step. We have also demonstrated that the method can be useful for enhancing low resolution images, creating oscillation-free surfaces, and reducing aliasing effects.

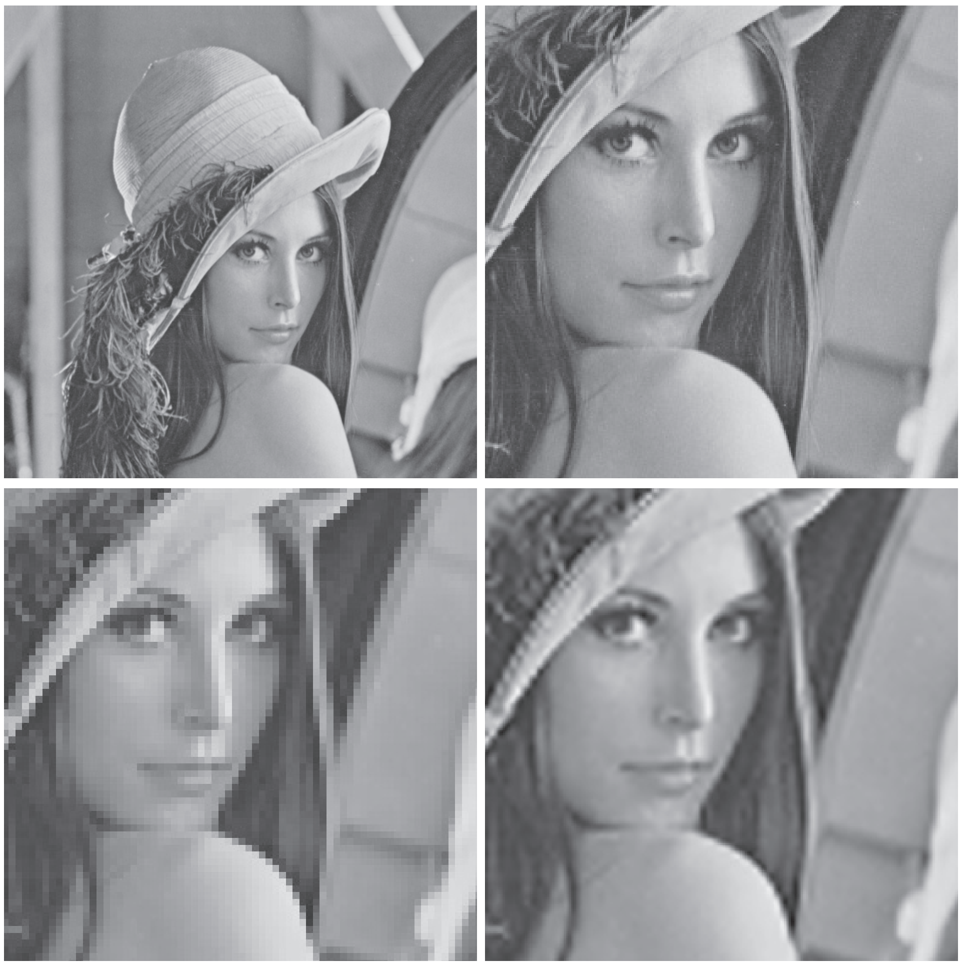

FIG. 10. *Lena test: Original image and zoomed part (top); downscaled (bottom left); and s*1 *reconstruction (bottom right).*

Appendix. Proof of Proposition 2.5. The proof is split into a series of lemmas.

LEMMA A.1 (strong duality). *For any $x^* \in \mathbb{R}^n$ and $\lambda^* \in \Lambda$, the pair (x^*, λ^*) solves* (3.9)–(3.10) *if and only if $f(x^*) = g(\lambda^*)$.*

Proof. Let us denote the open ℓ_1-ball in $\mathbb{R}^m$ of radius r by

$$B_r = \{z \in \mathbb{R}^m : |z|_1 < r\}$$

and then define the number

$$\rho = \inf\left\{r \in \mathbb{R} : \overline{B}_r \cap (b + \operatorname{Im} A) \neq \emptyset\right\}.$$

If $\rho = 0$, then $b \in \operatorname{Im} A$, and we have

$$0 = f(x^*) \geq g(\lambda^*) \geq g(0) = 0.$$

When $\rho > 0$, the set $\overline{B}_\rho \cap (b + \operatorname{Im} A)$ is nonempty (which implies existence of solutions of (3.9)), and it is a subset of the boundary of B_ρ. In particular, the convex sets B_ρ

and $b+\operatorname{Im} A$ are disjoint and can be separated by a hyperplane; i.e., there are $\mu \in \mathbb{R}^m$ and $\chi \in \mathbb{R}$ such that

$$\mu^t z \le \chi \quad \forall z \in B_\rho \qquad \text{and} \qquad \mu^t z \ge \chi \quad \forall z \in b + \operatorname{Im} A,$$

where we can take μ such that $|\mu|_\infty = 1$. By taking a supremum over z in the first inequality, we find

$$\sup_{|z|_1<\rho} \mu^t z = \rho|\mu|_\infty = \rho \le \chi.$$

The second inequality is equivalent to

$$\mu^t(b - Ax) \ge \chi \quad \forall x \in \mathbb{R}^n,$$

which implies that $A^t\mu = 0$, and therefore we have

$$\mu^t b \ge \chi \ge \rho.$$

Thus, we showed that μ is dual feasible, and for any pair of solutions x^* and λ^* of (3.9) and (3.10) we have

$$g(\mu) = \mu^t b \ge \rho = f(x^*) \ge g(\lambda^*) \ge g(\mu).$$

The converse is evident from (3.8). Note that the lemma can also be deduced from [22, Prop. 6.2]. ∎

LEMMA A.2. *For any $x^* \in \mathbb{R}^n$ and $\lambda^* \in \Lambda$, the pair (x^*, λ^*) solves* (3.9)–(3.10) *if and only if $(b - Ax^*)_i\lambda_i^* = |(b - Ax^*)_i|$ for all $i = 1, \dots, m$. In particular, if λ^* is a solution of* (3.10) *and $|\lambda_i^*| < 1$, then for every solution x^* of* (3.9) *we have $(b - Ax^*)_i = 0$.*

Proof. For any $\lambda^* \in \Lambda$ we have $g(\lambda^*) = (b - Ax^*)^t\lambda^* \le |b - Ax^*|_1 = f(x^*)$, and thus $g(\lambda^*) = f(x^*)$ is equivalent to $(b - Ax^*)_i\lambda_i^* = |(b - Ax^*)_i|$ for all $i = 1, \dots, m$, owing to Lemma A.1. ∎

We now assume that the matrix A and the vector b have the block structure

$$A = \begin{pmatrix} A_1 \\ \alpha A_2 \end{pmatrix}, \qquad b = \begin{pmatrix} b_1 \\ \alpha b_2 \end{pmatrix},$$

which is exactly the structure they have in problem (3.5), where A_2 and b_2 correspond to the rows generated by the terms $\int_F |[\![\partial_{\boldsymbol{n}} u_h]\!]|$, $F \in \mathcal{F}_h^i$, plus the rows generated by terms $\frac{\beta}{\alpha}|d_i(u_h) - \varpi_i|$, $i = 1, \dots, I_h$, if $\beta \ne 0$. (Recall that (3.5) is the matrix version of (2.21).) The primal function has the form

$$f(x) = |b - Ax|_1 = |b_1 - A_1x|_1 + \alpha|b_2 - A_2x|_1.$$

LEMMA A.3. *Assume that the rows of A_2 are linearly independent. Then there exists a number $\bar{\alpha}$ such that when $\alpha > \bar{\alpha}$ every solution x^* of* (3.9) *satisfies*

$$b_2 - A_2x^* = 0.$$

Proof. We will show that when α is large enough, the feasible set of the dual problem (3.10) (and therefore any solution) satisfies $|\lambda_2|_\infty < 1$, which, in view of Lemma A.2, implies the proposition. Indeed, if λ is dual feasible, then we have

$$0 = A^t\lambda = A_1^t\lambda_1 + \alpha A_2^t\lambda_2.$$

The assumption on A_2 implies the existence of right inverse R of A_2,

$$A_2R = I \qquad \text{or} \qquad R^tA_2^t = I,$$

and thus we have $\lambda_2 = -\frac{1}{\alpha}R^tA_1^t\lambda_1$. Now, if we define $\bar{\alpha} = |R^tA_1^t|_\infty$ and take $\alpha > \bar{\alpha}$, then we get

$$|\lambda_2|_\infty = \frac{1}{\alpha}|R^tA_1^t\lambda_1|_\infty \leq \frac{1}{\alpha}|R^tA_1^t|_\infty|\lambda_1|_\infty < 1,$$

which combined with Lemma A.2 concludes the proof. ☐

LEMMA A.4. *There exists a number $\bar{\alpha}$ so that for all $b_2 \in \mathrm{Im}A_2$ every solution x^* of* (3.9) *satisfies $b_2 - A_2x^* = 0$ for all $\alpha > \bar{\alpha}$.*

Proof. Let $\widetilde{A}_2$ denote the matrix whose rows are a maximal linearly independent set of rows of A_2. Without loss of generality we can write

$$A_2 = \begin{pmatrix} \widetilde{A}_2 \\ A_3 \end{pmatrix}, \qquad b_2 = \begin{pmatrix} \tilde{b}_2 \\ b_3 \end{pmatrix}.$$

We have the following property: if $b_2 \in \mathrm{Im}A_2$ and $\widetilde{A}_2x = \tilde{b}_2$, then $A_3x = b_3$. Let us now define

$$\widetilde{A} = \begin{pmatrix} A_1 \\ \alpha\widetilde{A}_2 \end{pmatrix}, \qquad \tilde{b} = \begin{pmatrix} b_1 \\ \alpha\tilde{b}_2 \end{pmatrix}$$

and consider the reduced minimization problem

$$\text{minimize } \tilde{f}(x) = |\tilde{b} - \widetilde{A}x|_1 = |b_1 - A_1x|_1 + \alpha|\tilde{b}_2 - \widetilde{A}_2x|_1, \tag{A.1}$$

obtained from (3.9) by replacing A and b with $\widetilde{A}$ and $\tilde{b}$, respectively. Since the rows of $\widetilde{A}_2$ are linearly independent, we apply Lemma A.3 to this problem and conclude that for $\alpha > \bar{\alpha}$ every solution $\tilde{x}$ of (A.1) satisfies $\widetilde{A}_2\tilde{x} = \tilde{b}_2$. We now assume that $\alpha > \bar{\alpha}$ and $b_2 \in \mathrm{Im}A_2$, and we want to show that problems (3.9) and (A.1) are equivalent. First we note that for all $x \in \mathbb{R}^n$

$$\begin{aligned}\tilde{f}(x) &= |b_1 - A_1x|_1 + \alpha|\tilde{b}_2 - \widetilde{A}_2x|_1 \\ &\leq |b_1 - A_1x|_1 + \alpha|\tilde{b}_2 - \widetilde{A}_2x|_1 + \alpha|b_3 - A_3x|_1 = f(x),\end{aligned}$$

and therefore for any two solutions x^* and $\tilde{x}$ of (3.9) and (A.1), respectively, we have $\tilde{f}(\tilde{x}) \leq f(x^*)$. Since $\widetilde{A}_2\tilde{x} = \tilde{b}_2$ and we assumed that $b_2 \in \mathrm{Im}A_2$, we conclude that $A_3\tilde{x} = b_3$, and therefore

$$f(\tilde{x}) = \tilde{f}(\tilde{x}) \leq f(x^*) \leq f(\tilde{x}),$$

which shows that $\tilde{f}(\tilde{x}) = f(x^*)$ and $\tilde{x}$ is a solution to (3.9). For $\tilde{f}(x^*)$ we have

$$\tilde{f}(x^*) \leq f(x^*) = \tilde{f}(\tilde{x}) \leq \tilde{f}(x^*),$$

which shows that $\tilde{f}(x^*) = \tilde{f}(\tilde{x})$, and therefore x^* is a solution to (A.1). Since we already know that $\widetilde{A}_2\tilde{x} = \tilde{b}_2$ and $A_3\tilde{x} = b_3$, i.e., $A_2\tilde{x} = b_2$, and since $\tilde{x}$ is an arbitrary solution to (A.1) (or, as we just proved, to (3.9)), this completes the proof. ☐

Let us now finish the proof of Proposition 2.5. We assume that Y_h is defined by (2.8); the proof of the second case is analogous. Since the mesh is composed of quadrilateral elements, it is possible to construct a function $v_h \in Y_h \cap \mathcal{C}^1(\overline{\Omega})$ using the Bogner–Fox–Schmit-type interpolation (see, e.g., [2, p. 72]), for which one prescribes the values of the function, its gradient, and its mixed second derivatives at the vertices of the mesh. The functional (2.21) can be put in the form

$$J_h(w_h) = |Ax - b_1|_1 + \alpha|A_2x - b_2|_1,$$

where A_2 and b_2 correspond to the rows generated by the terms $\int_F |[\![\partial_{\boldsymbol{n}} u_h]\!]|$ (the pointwise constraint $d_h(w_h) = \varpi_h$ is satisfied for all $w_h \in Y_h$ by the definition of Y_h). Moreover, $v_h \in \mathcal{C}^1(\overline{\Omega})$ implies that $\int_F |[\![\partial_{\boldsymbol{n}} u_h]\!]| = 0$ for all $F \in \mathcal{F}_h^i$, and because $|A_2x - b_2|_1$ is an equivalent discretization (see (2.16)) of $\sum_{F\in\mathcal{F}_h^i} \int_F |[\![\partial_{\boldsymbol{n}} u_h]\!]|$, we conclude that

$$|A_2x - b_2|_1 = 0,$$

which is equivalent to $b_2 \in \mathrm{Im}A_2$. Now, we apply Lemma A.4, and this completes the proof of Proposition 2.5. $\square$

REFERENCES

[1] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[2] D. Braess, *Finite Elements. Theory, Fast Solvers, and Applications in Solid Mechanics*, 2nd ed., Cambridge University Press, Cambridge, UK, 1997.

[3] E. J. Candès, J. K. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Comm. Pure Appl. Math., 59 (2006), pp. 1207–1223.

[4] E. J. Candes and T. Tao, *Decoding by linear programming*, IEEE Trans. Inform. Theory, 51 (2005), pp. 4203–4215.

[5] A. Chambolle and P.-L. Lions, *Image recovery via total variation minimization and related problems*, Numer. Math., 76 (1997), pp. 167–188.

[6] T. F. Chan and S. Esedoḡlu, *Aspects of total variation regularized L^1 function approximation*, SIAM J. Appl. Math., 65 (2005), pp. 1817–1837.

[7] J. Darbon and M. Sigelle, *Image restoration with discrete constrained total variation.* I. *Fast and exact optimization*, J. Math. Imaging Vision, 26 (2006), pp. 261–276.

[8] J. Duchon, *Splines minimizing rotation-invariant semi-norms in Sobolev spaces*, in Constructive Theory of Functions of Several Variables, Lecture Notes in Math. 571, Springer, Berlin, 1977, pp. 85–100.

[9] J. L. Guermond, *A finite element technique for solving first-order PDEs in L^P*, SIAM J. Numer. Anal., 42 (2004), pp. 714–737.

[10] J.-.L. Guermond and B. Popov, *Linear advection with ill-posed boundary conditions via L^1 minimization*, Int. J. Numer. Anal. Model., 4 (2007), pp. 39–47.

[11] J.-L. Guermond and B. Popov, *L^1-approximation of Hamilton–Jacobi equations*, SIAM J. Numer. Anal., 47 (2008), pp. 339–362.

[12] J.-L. Guermond and B. Popov, *L^1-minimization methods for Hamilton-Jacobi equations: The one-dimensional case*, Numer. Math., 109 (2008), pp. 269–284.

[13] M.-J. Lai and P. Wenston, *L_1 spline methods for scattered data interpolation and approximation*, Adv. Comput. Math., 21 (2004), pp. 293–315.

[14] J. E. Lavery, *Solution of steady-state one-dimensional conservation laws by mathematical programming*, SIAM J. Numer. Anal., 26 (1989), pp. 1081–1089.

[15] J. E. Lavery, *Solution of steady-state, two-dimensional conservation laws by mathematical programming*, SIAM J. Numer. Anal., 28 (1991), pp. 141–155.

[16] J. E. Lavery, *Univariate cubic L_p splines and shape-preserving, multiscale interpolation by univariate cubic L_1 splines*, Comput. Aided Geom. Design, 17 (2000), pp. 319–336.

[17] J. E. Lavery, *Shape-preserving, multiscale interpolation by bi- and multivariate cubic L_1 splines*, Comput. Aided Geom. Design, 18 (2001), pp. 321–343.

[18] J. E. Lavery, *Shape-preserving interpolation of irregular data by bivariate curvature-based cubic L_1 splines in spherical coordinates*, Comput. Aided Geom. Design, 22 (2005), pp. 818–837.

[19] A. Marquina and S. J. Osher, *Image super-resolution by TV-regularization and Bregman iteration*, J. Sci. Comput., 37 (2008), pp. 367–382.

[20] R. D. C. Monteiro and I. Adler, *Interior path following primal-dual algorithms.* I. *Linear programming*, Math. Programming, 44 (1989), pp. 27–41.

[21] L. C. Pickup, D. P. Capel, S. J. Roberts, and A. Zisserman, *Bayesian methods for image super-resolution*, The Computer Journal, 52 (2009), pp. 101–113.

[22] A. M. Pinkus, *On L^1-approximation*, Cambridge Tracts in Math. 93, Cambridge University Press, Cambridge, UK, 1989.

[23] A. C. Ponce and J. Van Schaftingen, *The continuity of functions with N-th derivative measure*, Houston J. Math., 33 (2007), pp. 927–939.

[24] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.

[25] D. F. Shanno and A. Bagchi, *A unified view of interior point methods for linear programming*, Ann. Oper. Res., 22 (1990), pp. 55–70.

[26] P. Vandewalle, S. Süsstrunk, and M. Vetterli, *A frequency domain approach to registration of aliased images with application to super-resolution*, EURASIP J. Adv. Signal Process., 2006 (2006), pp. 1–14.

[27] G. Wahba, *Spline Models for Observational Data*, CBMS-NSF Regional Conf. Ser. in Appl. Math. 59, SIAM, Philadelphia, 1990.

[28] Y. Wang, S.-C. Fang, and J. E. Lavery, *A compressed primal-dual method for generating bivariate cubic L_1 splines*, J. Comput. Appl. Math., 201 (2007), pp. 69–87.

[29] Y. Wang, S.-C. Fang, J. E. Lavery, and H. Cheng, *A geometric programming approach for bivariate cubic L_1 splines*, Comput. Math. Appl., 49 (2005), pp. 481–514.

[30] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.