Int. J. Appl. Math. Comput. Sci., 2022, Vol. 32, No. 4, 671–682
DOI: 10.34768/amcs-2022-0046

A PROXIMAL–BASED ALGORITHM FOR PIECEWISE SPARSE APPROXIMATION WITH APPLICATION TO SCATTERED DATA FITTING

YIJUN ZHONG [a,*], CHONGJUN LI [b], ZHONG LI [c], XIAOJUAN DUAN [a]

[a]Department of Mathematical Sciences
Zhejiang Sci-Tech University
No. 928, No. 2 Street, Xiasha Higher Education Park, Hangzhou, China
e-mail: zhongyijun@zstu.edu.cn

[b]School of Mathematical Sciences
Dalian University of Technology
No. 2 Linggong Road, Ganjingzi District, Dalian, China

[c]Department of Science and Technology
Huzhou University
759 Erhuan East Road, Huzhou, China

In some applications, there are signals with a piecewise structure to be recovered. In this paper, we propose a piecewise sparse approximation model and a piecewise proximal gradient method (JPGA) which aim to approximate piecewise signals. We also make an analysis of the JPGA based on differential equations, which provides another perspective on the convergence rate of the JPGA. In addition, we show that the problem of sparse representation of the fitting surface to the given scattered data can be considered as a piecewise sparse approximation. Numerical experimental results show that the JPGA can not only effectively fit the surface, but also protect the piecewise sparsity of the representation coefficient.

Keywords: piecewise sparse approximation, proximal gradient, scattered data fitting.

1. Introduction

In this paper, we consider recovering a sparse signal $\mathbf{x}^* \in \mathbb{R}^n$ from its noisy linear measurements

$$\mathbf{b} = A\mathbf{x}^* + \mathbf{e}, \tag{1}$$

where $\mathbf{b} \in \mathbb{R}^m$ is a measurement vector, $A \in \mathbb{R}^{m\times n}$ is a measurement matrix, and $\mathbf{e} \in \mathbf{N}(0, \sigma^2\mathbf{I}_n)$ is Gaussian noise. The sparse vector $\mathbf{x}^*$ has $s \leq m < n$ nonzero entries.

In some applications such as signal processing and machine learning, the signal possesses some structure, i.e., "piecewise sparse". For example, consider the decomposition of an image into texture and cartoon parts by Starck and Donoho (2005), i.e., $\mathbf{b} = A_n\mathbf{x}_n + A_t\mathbf{x}_t$ where n and t represent the cartoon and texture, respectively. It is assumed that both the parts can be represented in some given dictionaries, thus, $\mathbf{x}_n$ and $\mathbf{x}_t$ are two sparse vectors. The coefficient vector $\mathbf{x} = (\mathbf{x}_n^T, \mathbf{x}_t^T)^T$ is a "piecewise" sparse vector. Actually, the decomposition problem can be also seen as a demixing problem (McCoy and Tropp, 2014) which aims at extracting two constituents from the mixture $\mathbf{b}$.

Another example is sparse approximation of a fitting surface from scattered data (Hao *et al.*, 2018). Consider the approximation in space $H = \bigcup_{i=1}^N H_j$, where $H_j \subseteq H_{j+1}$ are principal shift invariant (PSI) spaces generated by a single compactly supported function; the fitting surface is $g = \sum_{i=1}^N g_i,\ g_i \in H_i$ with $g_i = \sum_{j=1}^{n_i} c_j^i \phi_j^i$. The coefficients $\mathbf{c} = (\mathbf{c}^1, \mathbf{c}^2, \ldots, \mathbf{c}^N)^T$ (by N pieces $\mathbf{c}^i = (c_1^i, \ldots, c_{n_i}^i)^T$) form a vector to be determined. Due to some inherent properties of PSI spaces, the coefficients are "piecewise" sparse structured, i.e., each $\mathbf{c}^i \in \mathbb{R}^{n_i}$ is a

*Corresponding author

sparse vector in H_i.

To be general, for a given vector we have

$$\mathbf{x} = (\underbrace{x_1, \ldots, x_{d_1}}_{\mathbf{x}_1^T}, \underbrace{x_{d_1+1}, \ldots, x_{d_1+d_2}}_{\mathbf{x}_2^T}, \ldots, \underbrace{x_{n-d_N+1}, \ldots, x_n}_{\mathbf{x}_N^T})^T,$$

where $n = \sum_{i=1}^{N} d_i$, and $s_i = \|\mathbf{x}_i\|_0$, $i = 1, \ldots, N$. In general, there are three types of sparsity for describing the distribution of nonzero entries in $\mathbf{x}$:

- global sparsity: $\mathbf{x}$ is called s-sparse if $\mathbf{x}$ contains $\|\mathbf{x}\|_0 \leq s$ nonzero elements;
- block sparsity: $\mathbf{x}$ is called s-block sparse if $\mathbf{x}$ contains $\|\mathbf{x}\|_{2,0} = \sum_{i=1}^{N} I(\|\mathbf{x}_i\|_2) \leq s$ blocks with nonzero entries;
- piecewise sparsity: see Definition 1.

Definition 1. (*Piecewise sparsity (Zhong and Li 2020)*) Suppose that an m-sample vector $\mathbf{b}$ is a linear superposition of N components with some additive noise,

$$\mathbf{b} = \sum_{i=1}^{N} \mathbf{b}_i + \mathbf{e}. \tag{2}$$

Furthermore, assume that each $\mathbf{b}_i$ can be sparsely represented in a basis A_i, i.e,

$$\mathbf{b}_i = A_i \mathbf{x}_i, \;\; i = 1, \ldots, N,$$

where $\mathbf{x}_i$ is a sparse vector. We define the vector $\mathbf{x} = (\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T)^T$ as a piecewise sparse vector.

Piecewise sparsity is a type of sparsity which describes a scattered distribution of nonzero elements in vectors. A brief description of the difference between block sparsity and piecewise sparsity can be found in (Zhong and Li, 2020).

According to the piecewise structure of the signal $\mathbf{x}$, the measurement matrix A is also structured as $A = [A_1, \ldots, A_N]$ where $A_i \in \mathbb{R}^{m \times n_i}$. Then the linear measurements (1) can be rewritten as

$$\mathbf{b} = \sum_{i=1}^{N} A_i \mathbf{x}_i^* + \mathbf{e}.$$

The structured information of the matrix A can be exploited to improve the sufficient conditions for successfully piecewise sparse recovery and the reliability of the greedy algorithms and BP algorithms (Li and Zhong, 2019). Moreover, the piecewise sparse structure of the signal inspires one to design a piecewise version of classical algorithm for sparse recovery in order to avoid selecting redundant false small-scaled elements (Zhong and Li, 2020).

In this paper, we study a generic piecewise sparse approximation which can be described as

$$\min_{(\mathbf{x}_1, \ldots, \mathbf{x}_N)} \frac{1}{2} \|A_1 \mathbf{x}_1 + \cdots + A_N \mathbf{x}_N - \mathbf{b}\|_2^2 \tag{3a}$$

subject to

$$R_i(\mathbf{x}_i) \leq R_i(\mathbf{x}_i^*), \quad i = 1, \ldots, N, \tag{3b}$$

where $\mathbf{b} \in \mathbb{R}^m$ is a given data vector and $A_1, \ldots, A_N$ are frames or bases. The goal is to decompose a given data vector $\mathbf{b}$ into a sum of components $\mathbf{b}_i$, each of which being sparsely represented in some frame A_i; in other words, each of which is 'small' in a sense described by the term R_i. Each function R_i aims at preserving the sparsity of sub-vector $\mathbf{x}_i$.

For piecewise sparse approximation, we note that the theory of Lagrange multipliers indicates that solving the constrained program (3) is essentially equivalent to solving the regularized problem

$$\min_{(\mathbf{x}_1, \ldots, \mathbf{x}_N)} \frac{1}{2} \|A_1 \mathbf{x}_1 + \cdots + A_N \mathbf{x}_N - \mathbf{b}\|_2^2 + \sum_{i=1}^{N} \lambda_i R_i(\mathbf{x}_i) \tag{4}$$

with the best choice of regularization parameters λ_i. Actually, (4) has strictly more optimal points than (3) (Rockafellar, 1970). In this paper, we do not restrict ourselves to specific choices of parameters λ_i; we study the design of an efficient algorithm to approximate the sub-vectors $\mathbf{x}_i$ by (3). In particular, (4) returns to basis pursuit denoising (BPDN) when $\lambda_1 = \cdots = \lambda_N$ and $R_i(\cdot) = \|\cdot\|_1$. The successive approximation bounds can be improved due to the piecewise sparse structure (see Fig.1) (Li and Zhong, 2019).

It is noted that Fig. 1 differs from the phase transition diagrams of the sparse approximation algorithm, which shows the empirical probability of successful recovery or approximation of algorithm. Figure 1 displays the theoretical performance guarantee for BPDN and (4) with $\lambda_1 = \cdots = \lambda_N$ and $R_i(\cdot) = \|\cdot\|_1$. The upper left curve shows that the piecewise structured information of both vector $\mathbf{x}$ and matrix A results in improved bounds, i.e., the upper bounds on the number of nonzero entries (4) can be recovered.

It is to be noted that the piecewise sparse vector $\mathbf{x}$ is also a global s-sparse vector. Large amounts of literature on approximating global sparse signals have been produced (Beck and Teboulle, 2009; Cai *et al.*, 2009; Zhang *et al.*, 2011; Pięta and Szmuc, 2021). Though algorithms for global sparse approximation can be used directly for solving (3), they may fail to preserve the piecewise sparse structure defined by R_i (see Example 4.1

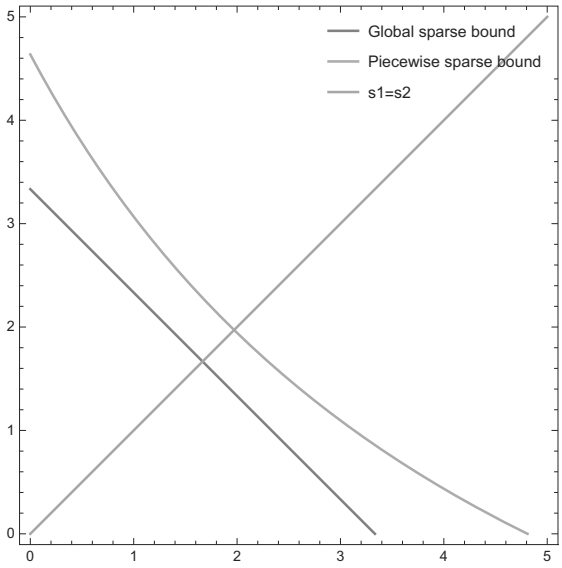


Fig. 1. Performance guarantee for BPDN with global sparsity and piecewise sparsity: the x- and y-axes represent the sparsities of $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively, or the piecewise sparsity (s_1, s_2) of $\mathbf{x} = (\mathbf{x}_1^T, \mathbf{x}_2^T)^T$.

of Zhong and Li (2020)). The majority of the algorithms for solving (3) are distributed algorithms, such as lots of extensions of the Bregman iteration (Sun *et al.*, 2015; Wei *et al.*, 2017; Wang *et al.*, 2018; Zhang and Luo, 2020). Indeed, a number of the well-known algorithms (e.g., iterative thresholding, projected Landweber, projected gradient, alternating projections, ADMM, and algorithms mentioned above) can be considered as proximal-based algorithms. As described by Parikh and Boyd (2014), proximal methods sit at a higher level of abstraction than classical optimization algorithms like Newton's method. Proximal-based algorithms enjoy the merits of easy computing by solving small convex optimization problems.

In this paper, we use a parallel and distributed extension of a proximal gradient algorithm (a Jacobi-type PGA) to solve (3). The piecewise version of the PGA (or the Jacobi-type PGA) is designed to approximate each sub-vector $\mathbf{x}_i$ from (3) separately and simultaneously. We show that solving (3) by the JPGA is in fact computationally easier and preserves better piecewise sparsity than classical algorithms, e.g., can be applied to approximation problems with a piecewise structure. Moreover, we show that the JPGA inherits a merit of the proximal algorithm, i.e., the $O(1/k)$ convergence rate, from a perspective of a differential equation.

In particular, we apply the JPGA to obtain a piecewise sparse representation of scattered data fitting. In fact, only a few papers have considered sparse approximation of scattered data fitting as a piecewise sparse problem. Large amounts of literature have made significant progress in solving interpolation and approximation problems by using multilevel B-splines, such as the quasi-interpolation algorithm based on hierarchical B-splines (Kraft, 1997), multilevel B-splines for scattered data interpolation (Lee *et al.*, 1997), multilevel regularization of wavelet based fitting of scattered data (Castaño and Kunoth, 2005), PHT-splines (polynomial splines over hierarchical T-meshes) (Deng *et al.*, 2008), which can be seen as a generalization of B-splines over hierarchical T-meshes, THB-splines (the truncated basis for hierarchical splines) (Giannelli *et al.*, 2012), hierarchical MK splines (Cai *et al.*, 2016), a modified multilevel B-spline approximation method (Moon and Ko, 2018), modified PHT-splines (Ni *et al.*, 2019), and other applications in control theory (Bingi *et al.*, 2019). We focus on a piecewise sparse approximation approach using uniform B-splines as basis functions for scattered data fitting. We mention that the use of B-splines as basis functions is not new, and we mainly show the piecewise sparse perspective to solve approximation problems. We try to make a link between the sparse representation in the PSI space and structured sparsity, which will provide interesting insights into the application of sparse techniques in function approximation.

The remaining part of the paper proceeds as follows: Preliminaries on proximal gradient algorithms are shown in Section 2. Section 3 is concerned with the methodology used for this study. We examine our method in Section 4.

2. Preliminaries

Definition 2. (*Proximal operator*) The proximal operator of a function f is defined by (Parikh and Boyd, 2014)

$$\text{prox}_f(\mathbf{v}) = \arg\min_{\mathbf{x}}(f(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|_2^2), \qquad (5)$$

where $\|\cdot\|_2$ is the Euclidean norm.

For a convex optimization problem

$$\min_{\mathbf{x}} F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}),$$

where $f(\cdot)$ is differentiable and $g(\cdot)$ is convex (possibly nondifferentiable), the proximal gradient algorithm iterates by

$$\mathbf{x}^{k+1} = \text{prox}_{\alpha^k g}(\mathbf{x}^k - \alpha^k \nabla f(\mathbf{x}^k)) \qquad (6)$$

where $\alpha^k > 0$ is a step size. It is known that when ∇f is Lipschitz continuous with constant L, this iteration (6) with a fixed step size $\alpha^k = \alpha \in (0, 1/L]$ converges with rate $O(1/k)$ (Parikh and Boyd, 2014).

3. Piecewise proximal gradient method

In an attempt to solve (3), we use the l^1 norm to keep the piecewise sparse structure, i.e., $R_i = \|\mathbf{x}_i\|_1$. We use a parallel and distributed extension to the proximal gradient algorithm, which is a Jacobi-type algorithm: for each "piece," we approximate subvector $\mathbf{x}_i$ by solving minimization using the iteration

$$\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i} \|\mathbf{x}_i\|_1 + \frac{1}{2}\|A_i\mathbf{x}_i + \sum_{j\neq i} A_j\mathbf{x}_j^k - \mathbf{b}^\sigma\|_2^2, \quad (7)$$

where $\mathbf{x}_j^k$ is the k-th update of subvector $\mathbf{x}_j$. Observe that (7) is similar to LASSO. However, in contrast to the traditional LASSO, we do not use the regularization parameter λ to balance the residual and regularization terms. The minimization (7) is not a standard LASSO since $\sum_{j\neq i} A_j\mathbf{x}_j^k$ is fixed based on the previous iteration and we do not seek a balance between $\|\mathbf{x}_i\|_1$ and $\|A_i\mathbf{x}_i + (\sum_{j\neq i} A_j\mathbf{x}_j^k - \mathbf{b}^\sigma)\|_2$. Actually, there are two "balances" in the JPGA:

- The balance between the residual term $\|\sum_{i=1} A_i\mathbf{x}_i - \mathbf{b}^\sigma\|_2$ and $\|\mathbf{x}_i\|_1$: we want to make the residual norm small enough in order to obtain an appropriate fitting with sparse representation (small l^1 norm). Thus a regularization parameter λ ($\lambda\|\mathbf{x}_i\|_1 + \frac{1}{2}\|A_i\mathbf{x}_i + \sum_{j\neq i} A_j\mathbf{x}_j^k - \mathbf{b}^\sigma\|_2^2$) does not yield the balance we need. Actually, the iteration in parallel ($i = 1,\ldots,N$) with a stopping rule on the residual norm automatically approximates the balance.
- The balance among $\|\mathbf{x}_i\|_1$ for $i = 1,\ldots,N$: we need a piecewise sparse structure of the determined vector $\mathbf{x}$. We obtain this balance by setting a proper iteration step and step size.

Write $R_i(\cdot) = \|\cdot\|_1$ and $f(\cdot) = \frac{1}{2}\|A\cdot - \mathbf{b}^\sigma\|_2^2$. We propose a piecewise proximal gradient algorithm which is essentially a Jacobi-type algorithm, cf. Algorithm 1. Notice that in the JPGA, we have

$$\nabla_{\mathbf{x}_i} f(\mathbf{x}^k) = A_i^T(\sum_{i=1}^N A_i\mathbf{x}_i^k - \mathbf{b}^\sigma).$$

3.1. JPGA as a continuous dynamical system.

Consider the JPGA iteration

$$\begin{cases} \mathbf{x}_1^{k+1} = \mathrm{prox}_{\alpha_1^k R_1}(\mathbf{x}_1^k - \alpha_1^k\nabla_{\mathbf{x}_1} f(\mathbf{x}^k)), \\ \quad\vdots \\ \mathbf{x}_N^{k+1} = \mathrm{prox}_{\alpha_N^k R_N}(\mathbf{x}_N^k - \alpha_N^k\nabla_{\mathbf{x}_N} f(\mathbf{x}^k)) \end{cases}$$

Algorithm 1. JPGA for piecewise sparse approximation.

Require: matrix $A = [A_1,\ldots,A_N]$, noisy observation function $\mathbf{b}^\sigma$;
Ensure: piecewise sparse vector $\mathbf{x} = (\mathbf{x}_1^T,\ldots,\mathbf{x}_N^T)^T$;
1: Given $\alpha_i^k (i = 1,\ldots,N)$
2: Let $\alpha_i^k = \alpha_i$
3: $i = 1,\ldots,N$ **repeat** solve $\mathbf{z}_i = \mathrm{prox}_{\alpha_i^k R_i}(\mathbf{x}_i^k - \alpha_i^k\nabla_{\mathbf{x}_i} f(\mathbf{x}^k))$
4: **break if** stopping rule holds
5: **return** $\alpha_i^{k+1} = \alpha_i$, $\mathbf{x}_i^{k+1} = \mathbf{z}_i$

which solves the following optimization problems:

$$\begin{cases} \mathbf{x}_1^{k+1} = \arg\min_{\mathbf{x}_1}\Big\{\|\mathbf{x}_1\|_1 \\ \qquad + \frac{L_1}{2}\|\mathbf{x}_1 - (\mathbf{x}_1^k - \frac{1}{L_1}\nabla_{\mathbf{x}_1} f(\mathbf{x}^k))\|_2^2\Big\}, \\ \quad\vdots \\ \mathbf{x}_N^{k+1} = \arg\min_{\mathbf{x}_N}\Big\{\|\mathbf{x}_N\|_1 \\ \qquad + \frac{L_N}{2}\|\mathbf{x}_N - (\mathbf{x}_N^k - \frac{1}{L_N}\nabla_{\mathbf{x}_N} f(\mathbf{x}^k))\|_2^2\Big\}, \end{cases} \quad (8)$$

where L_i ($i = 1,\ldots,N$) are Lipschitz constants with respect to $\nabla_{\mathbf{x}_i} f(\mathbf{x})$. The first-order optimality condition of (8) asserts that there exist $\mathbf{p}_i^{k+1} \in \partial\|\mathbf{x}_i^{k+1}\|_1$ that satisfy

$$\begin{cases} -\mathbf{p}_1^{k+1} = L_1(\mathbf{x}_1^{k+1} - \mathbf{x}_1^k) + A_1^T\mathbf{h}, \\ \quad\vdots \\ -\mathbf{p}_N^{k+1} = L_N(\mathbf{x}_N^{k+1} - \mathbf{x}_N^k) + A_N^T\mathbf{h}, \end{cases} \quad (9)$$

where

$$\mathbf{h} = \sum_{i=1}^N A_i\mathbf{x}_i^k - \mathbf{b}^\sigma.$$

Consider $\mathbf{x}_i^k = \mathbf{x}_i(t)$ and $\mathbf{x}_i^{k+1} = \mathbf{x}_i(t+\delta)$ for $\delta > 0$ small enough. By the mean value theorem, $\mathbf{x}_i(t+\delta) = \mathbf{x}_i(t) + \delta\dot{\mathbf{x}}_i(t+\lambda\delta)$ with $\lambda \in (0,1)$; then $L_i(\mathbf{x}_i^{k+1} - \mathbf{x}_i^k) \to \dot{\mathbf{x}}_i(t)$ as $\delta \to 0$ and

$$\lim_{\delta\to 0}\frac{\mathbf{x}_i^{k+1} - \mathbf{x}_i^k}{\delta} = \lim_{\delta\to 0}\frac{\mathbf{x}_i(t+\lambda\delta) - \mathbf{x}_i(t)}{\delta} = \dot{\mathbf{x}}_i(t)$$

while setting $L_i = 1/\delta$ ($i = 1,\ldots,N$). Moreover, the variables $\mathbf{p}_i^{k+1} = \mathbf{p}_i(t+\delta) \to \mathbf{p}_i(t)$ as $\delta \to 0$. Thus we obtain the continuous dynamical systems of (9) with $\delta \to 0$:

$$\begin{pmatrix} \dot{\mathbf{x}}_1(t) \\ \vdots \\ \dot{\mathbf{x}}_N(t) \end{pmatrix} + \begin{pmatrix} \langle A_1, \mathbf{h}(t)\rangle \\ \vdots \\ \langle A_N, \mathbf{h}(t)\rangle \end{pmatrix} + \begin{pmatrix} \mathbf{p}_1(t) \\ \vdots \\ \mathbf{p}_N(t) \end{pmatrix} = 0, \quad (10)$$

where

$$\mathbf{h}(t) = \sum_{i=1}^{N} A_i \mathbf{x}_i(t) - \mathbf{b}^{\sigma}.$$

Write

$$V(\mathbf{x}_i) = \frac{1}{2}\|A_i\mathbf{x}_i + \sum_{j\neq i} A_j \mathbf{x}_j^k - \mathbf{b}^{\sigma}\|_2^2 + \|\mathbf{x}_i\|_1$$

$$X(t) = \begin{pmatrix} \mathbf{x}_1(t) \\ \vdots \\ \mathbf{x}_N(t) \end{pmatrix}.$$

The system (10) can be written as

$$\nabla V(X(t)) + \dot{X}(t) = 0. \tag{11}$$

Note that the symbol ∇ in (11) represents the subgradient of $V(\cdot)$ (there exist $\mathbf{p}_i \in \partial\|\mathbf{x}_i\|_1$ that satisfy (10)).

Following the work of Franca *et al.* (2018), we formulate the following theorems.

Theorem 1. *Suppose X^* is a local minimizer and an isolated stationary point of function $V(\cdot)$, i.e., there exists $\mathcal{G} \subset \mathbb{R}^n$ such that $X^* \in \mathcal{G}$, $\nabla V(X^*) = 0$ and $\nabla V(X) \neq 0$ for $\forall X \in \mathcal{G}\backslash X^*$, and*

$$V(X) > V(X^*), \quad \forall X \in \mathcal{G}\backslash X^*.$$

Then X^* is an asymptotically stable critical point of the JPGA flow (11).

Theorem 2. *Suppose that $\arg\min V \neq \emptyset$ and $V^* = \min_X V(X)$. Then there is a constant $C > 0$ and $X(t)$ in the JPGA flow (11) with initial condition $X(t_0) = \mathbf{x}^0$ such that*

$$V(X(t)) - V^* \leq \frac{C}{t}. \tag{12}$$

The proof of Theorems 1 and 2 can be found in Appendix.

4. Experimental results

In this section, we test our algorithm on a scattered data fitting simulation, which can be regarded as a special case of (3).

4.1. Scattered data fitting in PSI space. The problem of reconstructing a surface from scattered points is described as follows: Given a set of unorganized points $X = \{x_1, \ldots, x_n\} \subset \Omega \subset \mathbb{R}^d$ and corresponding function values $f|_X = \{f_1, \ldots, f_n\}$, find a function $g \in H$ which fits the data $\{(x_k, f_k)\}|_{k=1}^n$ well. The traditional smooth spline model for solving this problem is

$$\min \sum_{k=1}^{n} (g(x_k) - f_k)^2 + \alpha |g|_{H^m}^2.$$

However, since the function g belongs to a Beppo Levi space, the computation complexity rises with the increasing scale of data sets. In the work of Johnson *et al.* (2009), a space which is generated by a compact support function (PSI space) is proposed to overcome the above disadvantage. Moreover, it makes sparse representation of data possible by connection with wavelets and B-splines.

Let

$$\mathcal{H} = \bigcup_{i=1}^{N} H_i$$

be the approximation space, where $H_i(\subset H_{i+1})$ is a PSI space generated by B-spline functions. Denote by $\{\phi_j^i\}_{j=1}^{n_i}$ the basis functions of H_i, $1 \leq i \leq N$. Then the approximation surface $g = \text{span}\{\phi_j^i\}_{i,j=1}^{N,n_i}$ is given by

$$g = \sum_{i=1}^{N} g_i, \quad g_i \in H_i, \quad g_i = \sum_{j=1}^{m} c_j^i \phi_j^i,$$

where c_j^i are the representation coefficients. Let ϕ be a carefully chosen, compactly supported function, and denote by h the scaling parameter of the PSI space. We follow the definition by Hao *et al.* (2018) to define an index set K as follows:

$$K = \{k \in \mathbb{Z}^2 : \text{supp}\Big(\phi\big(\frac{2^s \cdot}{h} - k\big)\Big) \cap \Omega \neq \emptyset\}.$$

Then the matrix $A = [A_1, \ldots, A_N]$ in (4) is the observation matrix defined by ϕ_j^i,

$$A_i(s,k) = \phi_j\Big(\frac{2^i x_s}{h} - k\Big), \quad k \in K$$

and $\mathbf{b}^{\sigma}$ represents $\{f_i\}$. Then for given data sets $\{x_k, y_k\}$ and the corresponding observation data $\{f(x_k, y_k)\}$, we expect to obtain $g(x, y)$ which fits the unorganized data set best and results in its sparse representation. The approximation function $g(x, y)$ is generated by a tensor product B-spline basis function $\phi_j^i(x, y)(i = 1, \ldots, N; j = 1, \ldots, n_i)$, where N is the number of layers of the multi-level B-spline and the total number of basis function is $n = n_1 + \cdots + n_N$. Thus the undetermined approximation surface is

$$g(x, y) = \sum_{i=1}^{N} \sum_{j=1}^{n_j} c_j^i \phi_j^i(x, y), \tag{13}$$

where $\mathbf{c}^i = (c_1^i, \ldots, c_{n_i}^i)$ is a sparse vector. In this way, the coefficient vector $\mathbf{c} = (\mathbf{c}^1, \ldots, \mathbf{c}^N)^T$ is a piecewise sparse vector. Since the N multi-level B-spline basis functions are linearly dependent, the scattered data fitting problem can be seen as a piecewise sparse approximation problem.

4.2. Numerical comparison. In this part, we utilize the JPGA solving scattered data reconstruction in PSI space for comparison with traditional methods without considering a piecewise structure: PGD (ISTA) (Beck and Teboulle, 2009) and the Bregman iteration (ADMM) for solving the following MLASSO minimization problem (Hao *et al*., 2018):

$$\min_{(\mathbf{x}_1,\ldots,\mathbf{x}_N)} \sum_{i=1}^{N} \lambda_i \|\mathbf{x}_i\|_1 + \frac{1}{2}\|\sum_{i=1}^{N} A_i\mathbf{x}_i - \mathbf{b}^\sigma\|_2^2. \quad (14)$$

We generate noisy data sets $\{((x_k, y_k), f(x_k, y_k)) : k = 1, \ldots, 900\}$ by adding Gaussian noise ($\epsilon_k \sim \mathcal{N}(0, \delta)$):

$$f = f(x_k, y_k) + \epsilon_k, \quad k = 1, \ldots, 900.$$

For the numerical comparison, we employ the 2D tensor product quadratic B-spline as the function ϕ. We test the above three algorithms for $N = 4$ starting from level 1 with 5×5 biquadratic B-spline functions. The lengths of $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ and $\mathbf{x}_4$ are 25, 64, 196 and 625, respectively. We have

$$\begin{cases} f_1(x,y) = 0.75\exp(-(9x-2)^2/4 - (9y-2)^2/4) \\ \qquad + 0.75\exp(-(9x+1)^2/49 \\ \qquad - (9y+1)^2/10) + 0.5\exp(-(9x-7)^2/4 \\ \qquad - (9y-3)^2/4) - 0.2\exp(-(9x-4)^2 \\ \qquad - (9y-7)^2), \\ f_2(x,y) = \dfrac{10(4x-2)}{1+100(4x-2)^2}, \\ f_3(x,y) = \exp(-81(x-0.5)^2 - 81(y-0.5)^2)/4, \\ f_4(x,y) = (x^2-2x)\exp(-x^2-y^2-xy). \end{cases}$$

Measurements. In this part, we use four measurements to compare the results of our tests: (i) fitting error; (ii) normalized RMS (root mean square) error; (iii) CPU running time (sec.); (iv) piecewise sparsity. The fitting error of the given scattered points $\{(x_k, f_k)\}|_{k=1}^n$ is defined as follows:

$$\text{error} = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(g(x_k, y_k) - f(x_k, y_k))^2}.$$

The difference between f and g is measured by the normalized RMS error defined as follows:

$$\text{RMS} = \sqrt{\frac{1}{M_1 N_1}\sum_{i=1}^{M_1}\sum_{j=1}^{N_1}(g(\tilde{x}_i, \tilde{y}_j) - f(\tilde{x}_i, \tilde{y}_j))^2},$$

where

$$\tilde{x}_i = -1 + \frac{2i}{M_1 - 1}, \qquad i = 0, 1, \ldots, M_1 - 1,$$

$$\tilde{y}_j = -1 + \frac{2j}{N_1 - 1}, \qquad i = 0, 1, \ldots, N_1 - 1,$$

Table 1. Multiple regularization parameter setting in ADMM.

Function	Parameter value$((\lambda_1, \lambda_2, \lambda_3, \lambda_4))$
f_1	(0.05,0.01,0.02,0.05)
f_2	(0.03,0.02,0.02,0.04)
f_3	(0.03,0.02,0.02,0.03)
f_4	(0.01,0.01,0.01,0.01)

and $M_1 = N_1 = 50$.

In particular, the piecewise sparsity refers to the piecewise sparse structure of the vector $\mathbf{x}$ we obtained, i.e., $(s_1, \ldots, s_N)$-sparsity. We do not wish a too sparse vector, but prefer a piecewise sparse vector with s_i small but not all zeros.

Remark 1. The regularization parameters λ_i ($i = 1, \ldots, 4$) in Table 1 are selected based on the rule by Hao *et al*. (2018) (Section 2.3).

Figures 2 and 4 show the scattered data points and the corresponding approximation surface of the test function with JPGA, PGD and ADMM on noise levels $\sigma = 0.05$ and $\sigma = 0.1$, respectively. Figures 3 and 5 illustrate the piecewise sparsity of the resulted representation, i.e., the distribution of the support of the B-spline functions with nonzero entries. We use decreasing size rectangles to show the support of the B-splines corresponding to $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ and $\mathbf{x}_4$, respectively. In parallel, Tables 2 and 3 show the piecewise sparsity (the l^0 norm of the resulting $\mathbf{x}_i$), the fitting error, the RMS and the running time for each test.

Comparisons in a noisy case: JPGA and PGD. Compared with PGD, JPGA is more likely to perform well in terms of piecewise sparsity and running time. Actually, JPGA is a piecewise version of PGD; the piecewise design makes JPGA run faster and better preserve the piecewise sparse structure than PGD. Since PGD solves (3) with treating $\mathbf{x}$ as a global sparse vector, it does not yield good piecewise sparsity. For example, for function f_2 ($\sigma = 0.1$): $(10, 7, 43, 62)$ in Table 2, PGD do not result in sparse representations at the 3-th and 4-th pieces. In addition, PGD tends to spend more time when noise level grows, while JPGA is not that sensitive to the noise level.

Comparisons in a noisy case: JPGA and ADMM. It is showed that MLASSO solved by ADMM also provide sparser solutions with less error but more time than the traditional LASSO used by Hao *et al*. (2018). The multiple-parameter setting makes ADMM for solving (14) different from JPGA, which does not rely on regularization parameters. Note that one could adjust λ_i constantly until better piecewise sparsity is obtained. The parameter setting process depends greatly on the prior information which may not always be accessible

Table 2. Piecewise sparsity of algorithms for comparison on test functions.

Function	Methods	Piecewise sparsity
$f_1\ (\sigma = 0.05)$	JPGA	(7,11,3,0)
	PGD	(8,16,36,9)
	ADMM	(25,33,81,15)
$f_1\ (\sigma = 0.1)$	JPGA	(6,11,3,0)
	PGD	(7,13,35,44)
	ADMM	(25,33,90,88)
$f_2\ (\sigma = 0.05)$	JPGA	(5,9,20,2)
	PGD	(7,13,27,26)
	ADMM	(25,11,86,72)
$f_2\ (\sigma = 0.1)$	JPGA	(5,7,20,3)
	PGD	(10,7,43,62)
	ADMM	(25,8,94,153)
$f_3\ (\sigma = 0.05)$	JPGA	(0,1,0,0)
	PGD	(3,10,14,3)
	ADMM	(25,9,71,84)
$f_3\ (\sigma = 0.1)$	JPGA	(1,1,0,0)
	PGD	(3,11,40,41)
	ADMM	(25,8,65,195)
$f_4\ (\sigma = 0.05)$	JPGA	(11,11,2,0)
	PGD	(11,18,16,5)
	ADMM	(25,16,46,289)
$f_4\ (\sigma = 0.1)$	JPGA	(10,11,2,0)
	PGD	(12,21,38,45)
	ADMM	(25,12,44,383)

Table 3. Numerical results of three algorithms on test functions.

Function	Methods	Error	RMS	CPU
$f_1\ (\sigma = 0.05)$	JPGA	0.038	0.323	0.868
	PGD	0.049	0.329	1.059
	ADMM	0.045	0.332	10.00
$f_1\ (\sigma = 0.1)$	JPGA	0.083	0.332	0.730
	PGD	0.092	0.339	7.356
	ADMM	0.082	0.343	16.065
$f_2\ (\sigma = 0.05)$	JPGA	0.047	0.154	0.732
	PGD	0.063	0.172	12.291
	ADMM	0.053	0.180	3.101
$f_2\ (\sigma = 0.1)$	JPGA	0.096	0.177	0.991
	PGD	0.098	0.193	14.368
	ADMM	0.083	0.204	4.909
$f_3\ (\sigma = 0.05)$	JPGA	0.040	0.076	0.901
	PGD	0.049	0.079	0.627
	ADMM	0.043	0.082	4.387
$f_3\ (\sigma = 0.1)$	JPGA	0.069	0.116	0.752
	PGD	0.092	0.119	6.961
	ADMM	0.076	0.129	6.910
$f_4\ (\sigma = 0.05)$	JPGA	0.035	0.839	0.717
	PGD	0.049	0.845	3.994
	ADMM	0.048	0.848	1.730
$f_4\ (\sigma = 0.1)$	JPGA	0.065	0.844	0.847
	PGD	0.092	0.851	6.045
	ADMM	0.093	0.857	4.265

in applications. We also observe that the approximation surfaces obtained by ADMM are more fluctuating than others when the noise level increase.

5. Conclusion and discussion

This paper presents an approach to piecewise sparse approximation based on a proximal gradient method. The proposed JPGA can be treated as a differential system for which some traditional tools may be used for convergence rate analysis. Simulations show that JPGA not only runs faster than PGD and ADMM, but preserve the piecewise sparse structure with application to scattered data fitting.

The convex program (3) equipped with JPGA provides a parallel perspective for approximation problems with a piecewise sparse prior. There are some gaps between the performance theory of (3) and sufficient conditions for successful approximation of JPGA. A complete analysis of the theory for both the convex program and the piecewise version of the classical algorithm will be made in our future work.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (grants no. 11901529,11871137 and 61902356), the Open Project Program of the State Key Lab of CAD&CG (grant no. A2115), Zhejiang University, the Fundamental Research Funds for the Central Universities (no. QYWKC2018007), the Science Foundation of Zhejiang Sci-Tech University (grant no. 19062118-Y), and the Fundamental Research Funds of Zhejiang Sci-Tech University (grant no. 2020Q050).

References

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage thresholding algorithm for linear inverse problems, *SIAM Journal on Imaging Sciences* **2**(1): 183–202, DOI: 10.1137/080716542.

Bingi, K., Ibrahim, R., Karsiti, M.N., Hassam, S.M. and Harindran, V.R. (2019). Frequency response based curve fitting approximation of fractional-order PID controllers, *International Journal of Applied Mathematics and Computer Science* **29**(2): 311–326, DOI: 10.2478/amcs-2019-0023.

Cai, J.F., Osher, S. and Shen, Z. (2009). Convergence of the linearized Bregman iteration for ℓ_1 norm minimization, *Mathematics of Computation* **78**(268): 2127–2136.

Cai, Z., Lan, T. and Zheng, C. (2016). Hierarchical MK splines: Algorithm and applications to data fitting, *IEEE Transactions on Multimedia* **19**(5): 921–934.

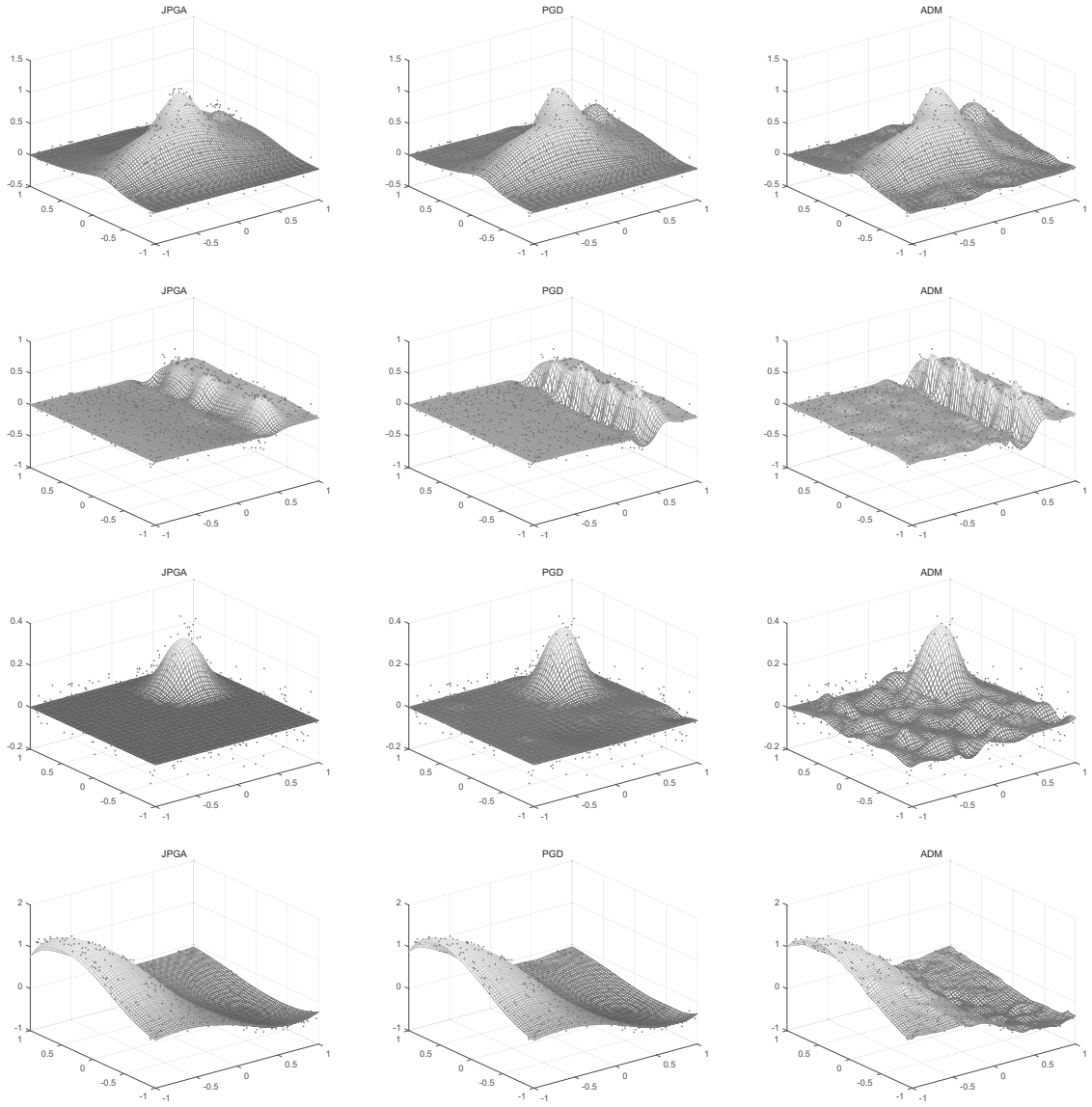


Fig. 2. Approximation surfaces for f_1 (first row), f_2 (second row), f_3 (third row) and f_4 (last row) with $\sigma = 0.05$.

Castaño, D. and Kunoth, A. (2005). Multilevel regularization of wavelet based fitting of scattered data some experiments, *Numerical Algorithms* **39**(1): 81–96.

Deng, J., Chen, F., Li, X., Hu, C., Yang, Z. and Feng, Y. (2008). Polynomial splines over hierarchical T-meshes, *Graphical Models* **70**(4): 76–86.

Franca, G., Robinson, D. and Vidal, R. (2018). ADMM and accelerated ADMM as continuous dynamical systems, *International Conference on Machine Learning, PMLR 2018, Stockholm, Sweden*, pp. 1559–1567.

Giannelli, C., Jüttler, B. and Speleers, H. (2012). THB-splines: The truncated basis for hierarchical splines, *Computer Aided Geometric Design* **29**(7): 485–498.

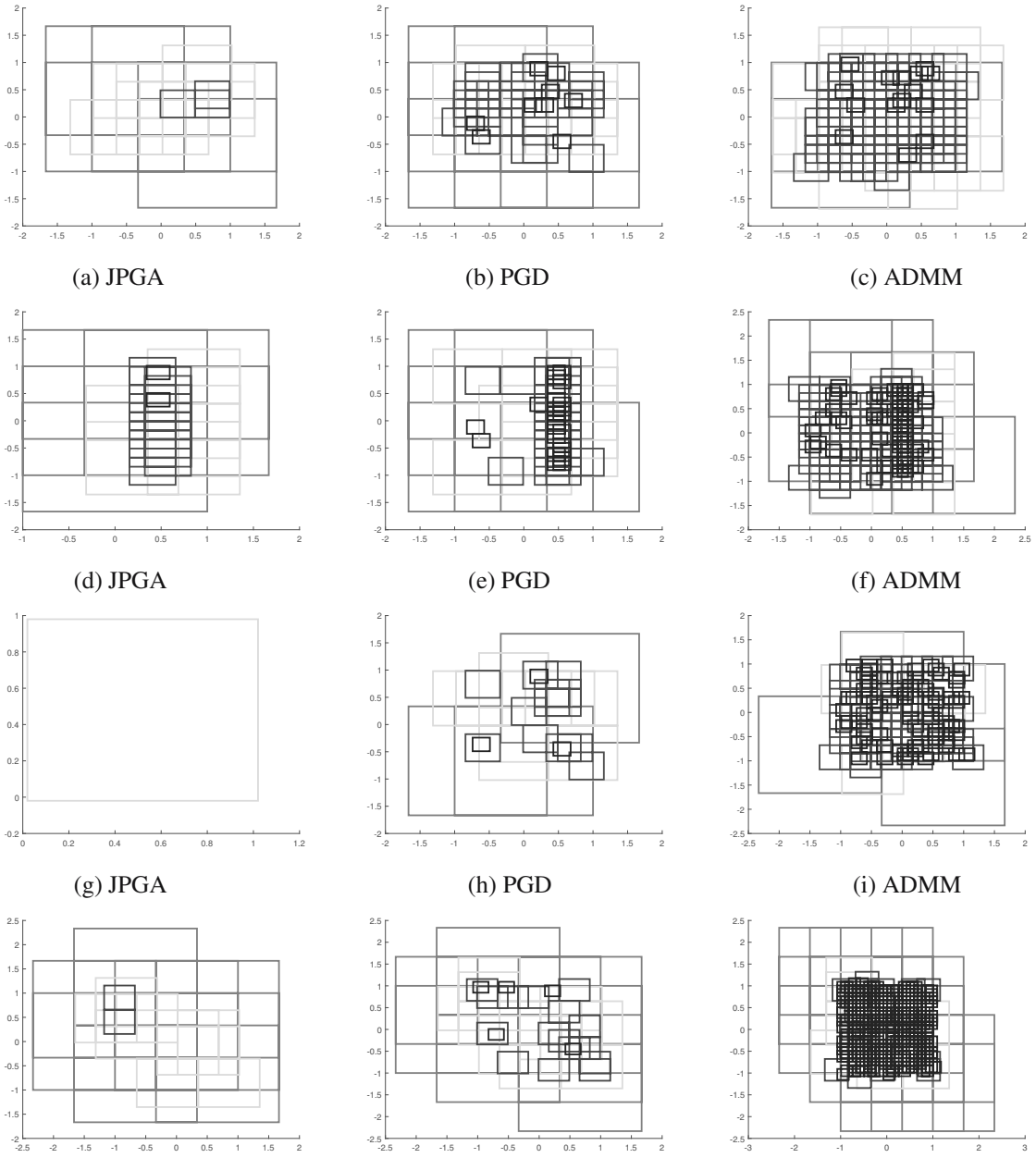
Hao, Y., Li, C. and Wang, R. (2018). Sparse approximate solution of fitting surface to scattered points by MLASSO model, *Science China Mathematics* **61**(7): 1319–1336, DOI: 10.1007/s11425-016-9087-y.

Hirsch, M.W., Smale, S. and Devaney, R.L. (2004). *Differential Equations, Dynamical Systems, and an Introduction to Chaos*, Academic Press, Amsterdam.

Johnson, M.J., Shen, Z. and Xu, Y. (2009). Scattered data reconstruction by regularization in B-spline and associated wavelet spaces, *Journal of Approximation Theory* **159**(2): 197–223, DOI: 10.1016/j.jat.2009.02.005.

Kraft, D. (1997). Adaptive and linearly independent multilevel B-splines, *in* Le Méhauté *et al.* (Eds), *Surface Fitting and Multiresolution Methods*, Vanderbilt University Press, Nashville, pp. 209–218.

Lee, S., Wolberg, G. and Shin, S. (1997). Scattered data interpolation with multilevel B-splines, *IEEE Transactions*



(a) JPGA (b) PGD (c) ADMM

(d) JPGA (e) PGD (f) ADMM

(g) JPGA (h) PGD (i) ADMM

(j) JPGA (k) PGD (l) ADMM

Fig. 3. Distribution of the support for f_1 (first line), f_2 (second line), f_3 (third line) and f_4 (last line) with $\sigma = 0.05$.

on Visualization and Computer Graphics **3**(3): 228–244, DOI: 10.1109/2945.620490.

Li, C. and Zhong, Y.J. (2019). Piecewise sparse recovery in union of bases, *arXiv* 1903.01208.

McCoy, M.B. and Tropp, J.A (2014). Sharp recovery bounds for convex demixing, with applications, *Foundations of Computational Mathematics* **14**(3): 503–567.

Moon, S. and Ko, K. (2018). A point projection approach for improving the accuracy of the multilevel b-spline approximation, *Journal of Computational Design and Engineering* **5**(2): 173–179.

Ni, Q., Wang, X. and Deng, J. (2019). Modified PHT-splines, *Computer Aided Geometric Design* **73**(1): 37–53.

Parikh, N. and Boyd, S. (2014). Proximal algorithms, *Foundations and Trends in Optimization* **1**(3): 127–239, DOI: 10.1.1.398.7055.

Pięta, P. and Szmuc, T. (2021). Applications of rough sets in big data analysis: An overview, *International Journal of Applied Mathematics and Computer Science* **31**(4): 659–683, DOI: 10.34768/amcs-2021-0046.

Rockafellar, R.T. (1970). *Convex Analysis*, Princeton University Press, Princeton.

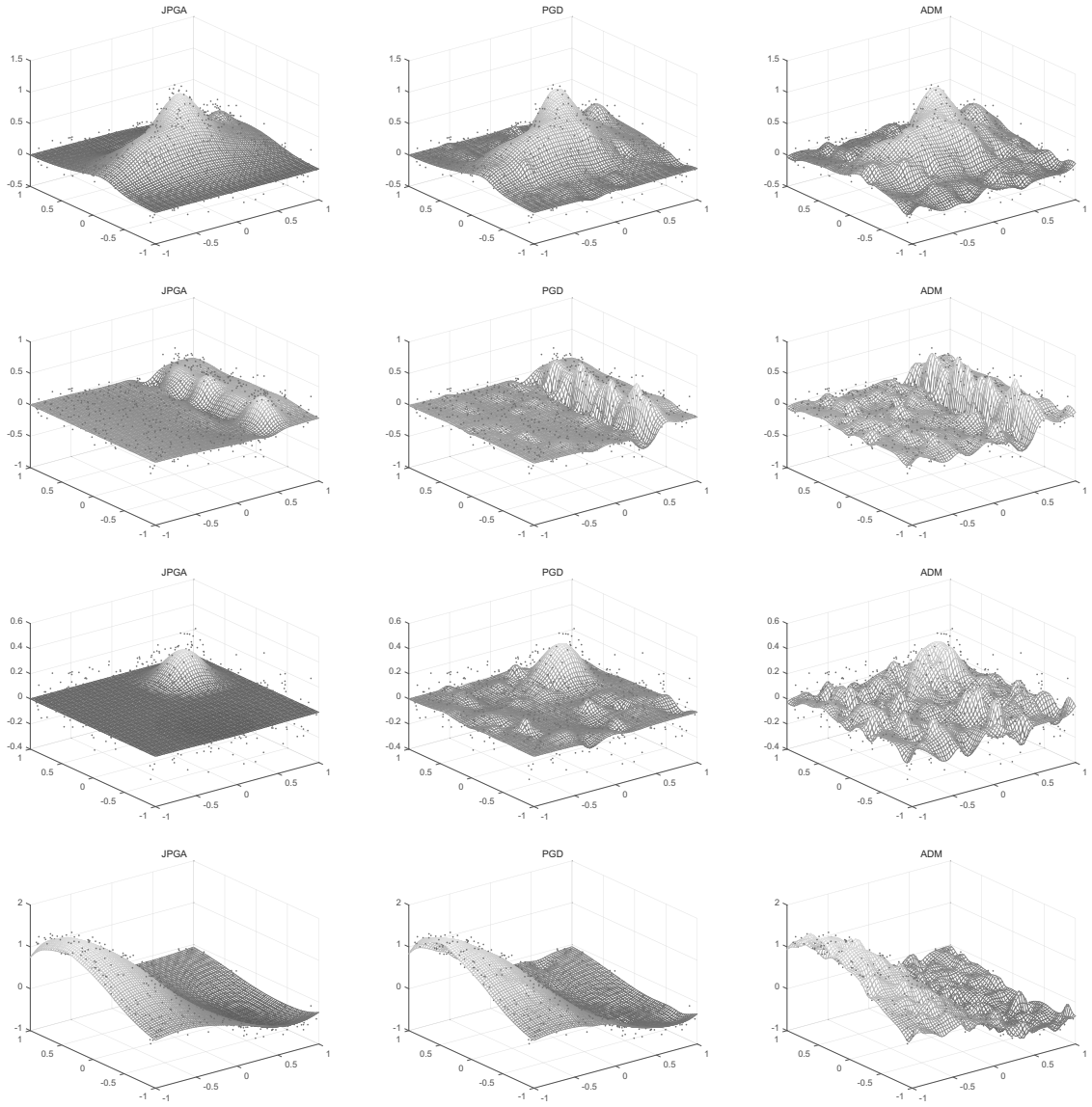


Fig. 4. Approximation surfaces for f_1 (first row), f_2 (second row), f_3 (third row) and f_4 (last row) with $\sigma = 0.1$.

Sun, D., Toh, K.C. and Yang, L. (2015). A convergent 3-block semiproximal alternating direction method of multipliers for conic programming with 4-type constraints, *SIAM Journal on Optimization* **25**(2): 882–915.

Starck, J.L., Elad, M. and Donoho, D.L. (2005). Image decomposition via the combination of sparse representations and a variational approach, *IEEE Transactions on Image Processing* **14**(10): 1570–1582, DOI: 10.1109/TIP.2005.852206.

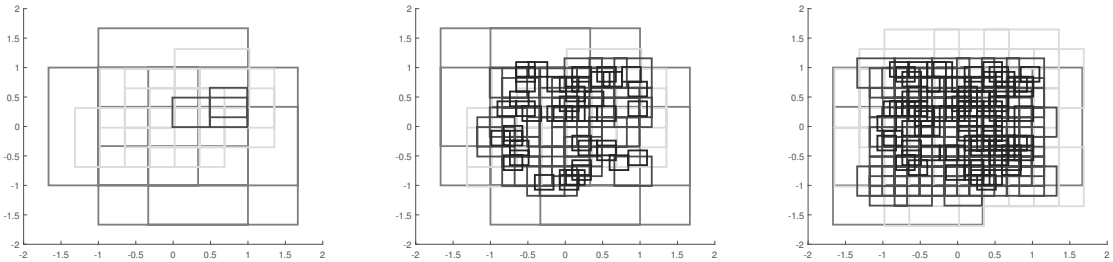
Wang, F., Cao, W. and Xu, Z. (2018). Convergence of multi-block Bregman ADMM for nonconvex composite problems, *Science China Information Sciences* **61**(12): 1–12.

Wei, D., Lai, M.J., Reng, Z. and Yin, W. (2017). Parallel multi-block ADMM with $o(1/k)$ convergence, *Journal of Scientific Computing* **71**(2): 712–736.

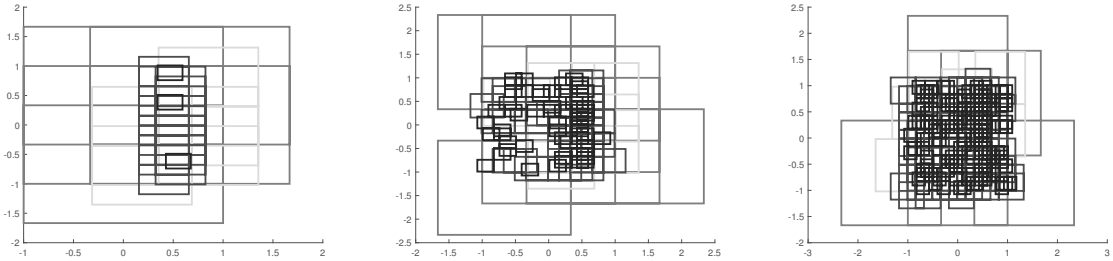
Zhang, J. and Luo, Z.Q. (2020). A proximal alternating direction method of multiplier for linearly constrained nonconvex minimization, *SIAM Journal on Optimization* **30**(3): 2272–2302.

Zhang, X., Burger, M. and Osher, S. (2011). A unified primal-dual algorithm framework based on Bregman iteration, *Journal of Scientific Computing* **46**(1): 20–46.

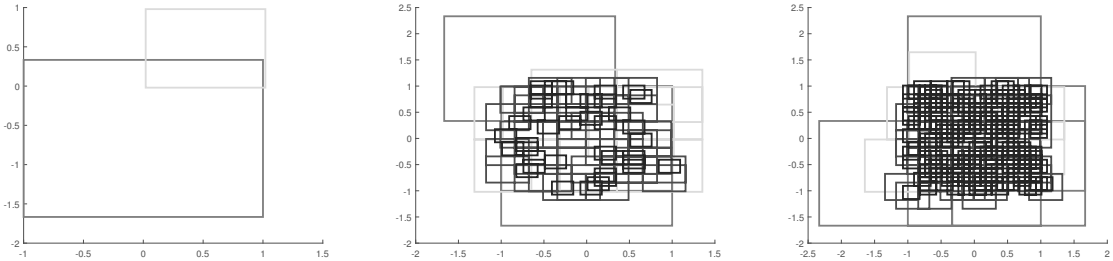
Zhong, Y. and Li, C. (2020). Piecewise sparse recovery via piecewise inverse scale space algorithm with deletion rule, *Journal of Computational Mathematics* **38**(2): 375–394, DOI: 10.4208/jcm.1810-m2017-0233.



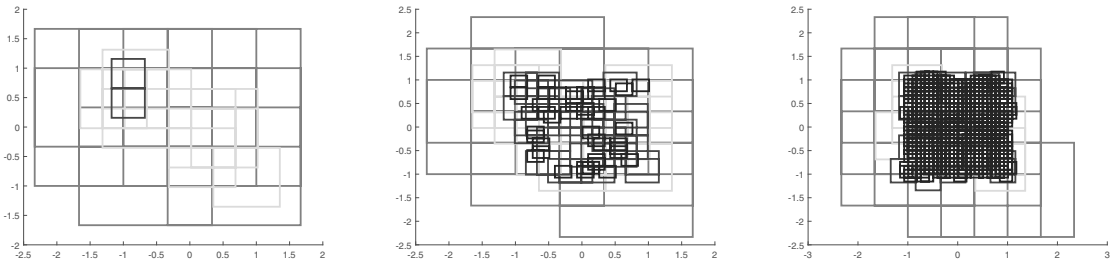
(a) JPGA (b) PGD (c) ADMM



(d) JPGA (e) PGD (f) ADMM



(g) JPGA (h) PGD (i) ADMM



(j) JPGA (k) PGD (l) ADMM

Fig. 5. Distribution of the support for f_1 (first row), f_2 (second row), f_3 (third row) and f_4 (last row) with $\sigma = 0.1$.

Yijun Zhong holds a BS degree in applied mathematics (2010) from Jiangxi Normal University, Nanchang, China, as well as an MS degree in operations and control theory (2013) and a PhD degree in computational mathematics (2018) from the Dalian University of Technology, China. She is currently a lecturer in the Department of Mathematical Sciences at Zhejiang Sci-Tech University, China. Her research interests include sparse optimization, sparse recovery, computer graphics and computer aided geometric design.

Chongjun Li holds a BS (1999) degree in mathematics from Yunnan University, Kunming, China, as well as MS (2001) and PhD (2004) degrees in computational mathematics from the Dalian University of Technology, China. He joined the Dalian University of Technology in 2004, where he is currently a professor. He specializes in multivariate splines and their applications. His present research interests include computational geometry, multivariate splines and spline finite element methods.

Zhong Li received his PhD degree in mathematics from Zhejiang University, Hangzhou, China, in 2003. He joined Zhejiang Sci-Tech University in 2007, and then Huzhou University in 2021, where he is currently a professor. His present research interests include computational geometry, deep learning and data analysis, computer graphics and computer aided geometric design.

Xiaojuan Duan holds a BS degree in information and computing science (2004) from the Hefei University of Technology, China, as well as an MS degree in mathematics (2009) and a PhD degree in applied mathematics (2015) from Zhejiang University, Hangzhou, China. She is currently a lecturer in the School of Science at Zhejiang Sci-Tech University, China. Her research interests include computer graphics and computer aided geometric design.

Appendix

A1. Proof of Theorem 1

First, we discuss some basic concepts of stability and Lyapunov function referring to Hirsch *et al*. (2004).

In order to prove that X^* is asymptotically stable, we define

$$\mathcal{E}(X) \equiv V(X) - V(X^*). \tag{A1}$$

Using $\nabla V(X^*) = 0$ and (11), we have

$$\dot{\mathcal{E}}(X) = \langle \nabla V(X), \dot{X} \rangle = -\|\dot{X}\|_2^2, \tag{A2}$$

which shows that $X \in \mathcal{G} \backslash X^*$, $\dot{\mathcal{E}}(X) < 0$. Thus, by using Theorem 5 by Franca *et al*. (2018), X^* is an asymptotically stable critical point of (11).

A2. Proof of Theorem 2

Assume that $X^* \in \arg\min V(X)$. Then $V(X^*) = V^*$. Define

$$\mathcal{E}(X,t) = t[V(X) - V(X^*)] + \frac{1}{2}\|X - X^*\|^2.$$

The derivative of $\mathcal{E}$ at t is

$$\begin{aligned}\dot{\mathcal{E}} = {} & V(X) - V(X^*) + t\langle \nabla V(X), \dot{X}(t) \rangle \\ & + \langle X - X^*, \dot{X}(t) \rangle. \end{aligned} \tag{A3}$$

From (11) we get $\nabla V(X) = -\dot{X}$. Observe that the right-hand side of (A3) can be rewritten as

$$-t\|\dot{X}\|^2 + V(X) - V(X^*) + \langle X^* - X, \nabla V(X) \rangle. \tag{A4}$$

$V(\cdot)$ is convex and

$$V(X) - V(X^*) + \langle X^* - X, \nabla V(X) \rangle \leq 0.$$

Accordingly, (A4) is nonpositive. Thus we have $\dot{\mathcal{E}} \leq 0$, which means $\mathcal{E}(X,t)$ is decreasing in t, i.e.,

$$\mathcal{E}(X,t) \leq \mathcal{E}(X_0, t_0)$$

for all $t \geq t_0$. Thus

$$\begin{aligned} V(X) - V(X^*) &= \frac{1}{t}\mathcal{E}(X,t) - \frac{1}{2t}\|X - X^*\|^2 \\ &\leq \frac{1}{t}\mathcal{E}(X_0, t_0). \end{aligned}$$

Then we get

$$V(X(t)) - V(X^*) \leq \frac{C}{t}.$$

Received: 16 December 2021
Revised: 27 April 2022
Accepted: 13 June 2022