Sparse approximate solution of fitting surface to scattered points by MLASSO model

Yongxia Hao¹, Chongjun Li²,* & Renhong Wang²

¹Faculty of Science, Jiangsu University, Zhenjiang 212000, China;
²School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China

Received May 16, 2016; accepted April 20, 2017; published online December 11, 2017

Abstract The goal of this paper is to achieve a computational model and corresponding efficient algorithm for obtaining a sparse representation of the fitting surface to the given scattered data. The basic idea of the model is to utilize the principal shift invariant (PSI) space and the $l_1$ norm minimization. In order to obtain different sparsity of the approximation solution, the problem is represented as a multilevel LASSO (MLASSO) model with different regularization parameters. The MLASSO model can be solved efficiently by the alternating direction method of multipliers. Numerical experiments indicate that compared to the AGLASSO model and the basic MBA algorithm, the MLASSO model can provide an acceptable compromise between the minimization of the data mismatch term and the sparsity of the solution. Moreover, the solution by the MLASSO model can reflect the regions of the underlying surface where high gradients occur.

Keywords sparse solution, principle shift invariant space, $l_1$ norm minimization, alternating direction method of multipliers, MLASSO model

MSC(2010) 65D17, 65K99

Citation: Hao Y X, Li C J, Wang R H. Sparse approximate solution of fitting surface to scattered points by MLASSO model. Sci China Math, 2018, 61: 1319–1336, https://doi.org/10.1007/s11425-016-9087-y

1 Introduction

Sparse representation of a function via a linear combination of a small number of functions has recently received a lot of attention in several mathematical fields such as approximation theory (see [14,34,41,42]), compressed sensing, signal and image processing (see [8–11]), etc. The problem can be described as follows. Consider a linearly dependent set of $n$ functions $\{\varphi_i\}_{i=1}^n$ and a function $f$ represented as $f = \sum_{i=1}^n X_i \varphi_i$. Since the set of functions is not linearly independent, this representation is not unique. The problem is then to find the sparsest solution, i.e., the coefficient vector $X = (X_1, X_2, \ldots, X_n)$ has as many zero components as possible (referred to minimizing the $l_0$ norm of the vector $X$). This optimization problem is NP-hard, since the $l_0$ norm is nonconvex and discontinuous. Hence, much attention has been paid to solutions minimizing $\|X\|_1 = \sum_{i=1}^n |X_i|$ instead.

In this paper, we consider the problem of reconstructing a surface from scattered data using a sparse representation. The scattered data fitting problem arises in many applications, such as signal processing,
computer graphics and neural networks (see [33]). In a typical scattered data reconstruction problem, we are given a set of scattered points \( \Xi = \{x_1, x_2, \ldots, x_N\} \subseteq \mathbb{R}^2 \) and associated noisy function values \( \tilde{f} = f \mid_\Xi + n = \{f_1, f_2, \ldots, f_N\} \), where \( n \) is the error vector. Then we seek a function \( g \) which fits the given data \( \{ (x_i, f_i) \}_{i=1}^{N} \) well. There are a lot of existing methods and algorithms in the literature. Various methods can be found in a survey on scattered data interpolation (see [6]). For approximation methods, B-splines have a solid mathematical foundation and have been used in many literatures, such as [35, 51], etc. Wavelet frames have also been used to reconstruct implicit surfaces from unorganized point sets in \( \mathbb{R}^3 \) (see [17]). In order to control the local and global fitting error simultaneously, adaptive methods are presented in [12, 44]. The adaptivity is achieved by a portion of the data with a patch, testing the fit for satisfaction within a given tolerance, and subdividing the patch if the tolerance is not met (see [24]). In addition, several approximation methods employ a multilevel structure to approximate data efficiently. In particular, a multilevel scheme based on B-splines is proposed in [36] to approximate scattered data. These methods run on the approximation space \( S = \bigcup_{j=1}^{J} S_j \), where \( S_j \subseteq S_{j+1} \) are principal shift invariant (PSI) spaces generated by a single compactly supported function \( \varphi \). The multilevel approximation procedure is as follows: for each level \( j \), the point set \( (\Xi, \Delta g_{j-1}) \) is approximated by a function \( g_j \in S_j \) obtained by the least square method, where \( \Delta g_{j-1} = \tilde{f} - (g_1 + \cdots + g_{j-1}) \mid_\Xi \). The procedure is terminated until certain conditions are satisfied. Then the final approximation surface is

\[
g = g_1 + g_2 + \cdots + g_J, \quad g_j \in S_j, \quad j = 1, \ldots, J.
\]

However, the methods above do not produce the sparse representation of the surface.

In this paper, we present an efficient method to obtain a sparse representation of the fitting surface to the given scattered points. We still choose the space \( S \) defined above as the approximation space. But instead of using the multilevel scheme, we put all the basis functions of \( S_j \), \( 1 \leq j \leq J \) together as a frame of \( S \). Denote the basis functions of \( S_j \) as \( \{\varphi_j^i\}_{i=1}^{n_j} \), \( 1 \leq j \leq J \). Then \( S = \text{span}\{\varphi_j^i\}_{i,j=1}^{n_j,J} \). We then try to find the fitting surface \( g \in S \) as

\[
g = \sum_{j=1}^{J} g_j, \quad g_j = \sum_{i=1}^{n_j} X_j^i \varphi_j^i.
\]

Since the functions \( \{\varphi_j^i\}_{i,j=1}^{n_j,J} \) are linearly dependent, the representation of \( g \) as above is not unique and we will seek a relatively sparse one. The choice of the space \( S \) makes a sparse representation of \( g \) exist and the function \( \varphi_j^i \) can be constructed in a multilevel way. We use the similar approach as those used in compressed sensing, i.e., to use the \( l_1 \) norm of the coefficient vector as the regularization term. Thus, the problem can be represented by the following minimization:

\[
\min_{g \in S} \sum_{i=1}^{N} (g(x_i) - f_i)^2 + \sum_{j=1}^{J} \lambda_j \|X_j\|_1,
\]

(1.1)

where \( X_j = (X_j^1, \ldots, X_j^n_j), 1 \leq j \leq J \) are coefficient vectors. The parameters \( \lambda_j > 0, j = 1, \ldots, J \) are called the regularization parameters, which serve as a weight to adjust the balance between the two terms. Large values of \( \lambda_j \) will lead to a sparse function \( g \), at the cost of a potentially large fitting error, while small values of \( \lambda_j \) will lead to a small fitting error, but with a potentially not too sparse fitting function \( g \). In addition, different values of \( \lambda_j \) can lead to different sparsity of \( g \). Let \( f \) denote the column vector \( \{f_i\}_{i=1}^{N} \), then the formulation (1.1) is equivalent to the following minimization:

\[
\min_{X_j, 1 \leq j \leq J} \left\| \sum_{j=1}^{J} A_j X_j - f \right\|_2^2 + \sum_{j=1}^{J} \lambda_j \|X_j\|_1,
\]

(1.2)

where \( A_j \) is the observation matrix defined by

\[
A_j(i,k) = \varphi_k^j(x_i), \quad i = 1, 2, \ldots, N, \quad k = 1, 2, \ldots, n_j, \quad j = 1, 2, \ldots, J.
\]
For the case $\lambda_1 = \cdots = \lambda_J$, the model reduces to the model presented in [31]. Moreover, the $l_1$ related minimization resulting from our proposed model can be efficiently solved using the alternating direction method of multipliers (ADMM) (see [25,54]).

This framework combines the ideas developed in compressed sensing with well-known concepts arising in adaptive and multilevel finite element methods. The solution of the $l_1$ minimization problem and the $l_1$ related minimization resulting from our proposed model can be efficiently solved using the alternating direction method of multipliers (ADMM) (see [25,54]).

This framework combines the ideas developed in compressed sensing with well-known concepts arising in adaptive and multilevel finite element methods. The solution of the $l_1$ minimization problem and the $l_1$ related minimization resulting from our proposed model can be efficiently solved using the alternating direction method of multipliers (ADMM) (see [25,54]).

The rest of the paper is organized as follows. In Section 2, we recall the main ingredients of the PSI space which will be used here. Moreover, we will propose the sparsity based regularization model for scattered data fitting. In Section 3, the ADMM algorithm will be applied to solve the minimization problem resulted from the proposed model. Numerical experiments are also performed to illustrate the algorithm. Section 4 is the conclusion.

2 Sparse solution of PSI approach to scattered data approximation

For a given set of scattered points $\{x_i\}_{i=1}^N \subseteq \Omega \subseteq \mathbb{R}^2$ and the corresponding noisy data $\{f_i\}_{i=1}^N$, our task is to reconstruct a fitting surface with a sparse representation.

2.1 PSI space and $l_1$ regularization

Let $\Omega \subseteq \mathbb{R}^2$ be a bounded domain of interest where all data lie in and let $\varphi$ be a carefully chosen, compactly supported function (e.g., uniform B-spline, box spline, radial basis functions). Denote

$$\Lambda = \{k \in \mathbb{Z}^2 : \text{supp}(\varphi(\cdot/h-k)) \cap \Omega \neq \emptyset\}, \quad S^h(\varphi, \Omega) = \left\{f = \sum_{k \in \Lambda} c_k \varphi(\cdot/h-k) \right\},$$

where $h > 0$ is a scaling parameter that controls the refinement of the space. Denote $S_j = S^{h/2^{j-1}}(\varphi, \Omega)$, $S_1 \subseteq \cdots \subseteq S_J$. Then we look for a function $g \in S_J$ which fits closely the given data. Then $g$ is composed of a sequence of functions as

$$g = g_1 + g_2 + \cdots + g_J,$$

where $g_i \in S_i, i = 1, 2, \ldots, J$.

Here, we choose a proper PSI space generated by B-spline as the approximation space $S$ since it enjoys desirable properties for data fitting. It has a simple structure and provides good approximation to smooth functions, which leads to simple and accurate algorithms. Moreover, it can be associated to a wavelet or frame system and hence one can solve the fitting problem by making use of the advantages that a wavelet (frame) system can offer (see [31]). These advantages include sparse approximation of functions in the wavelet (frame) domain, multilevel structure of basis functions, adaptivity to the data, norm equivalence, etc.

Recall that a function $\varphi$ is said to satisfy the Strang-Fix conditions of order $m$ if

$$\hat{\varphi}(0) \neq 0, \quad D^\alpha \hat{\varphi}(2\pi j) = 0, \quad \forall \ j \in \mathbb{Z}^2 \setminus \{0\}, \quad |\alpha| < m.$$
Let \( g \) vanishing coefficients as possible. Every function of \( g \) where
\[
\text{AGLASSO} \text{ is proposed in [49] as}
\]
\[A\text{GLASSO does not, however, yield sparsity within a group. Moreover, GLASSO suffers from estimation inefficiency and selection inconsistency. To remedy these problems, the adaptive GLASSO method (AGLASSO) is proposed in [49] as}
\]
\[
\text{where } \hat{f} \text{ is the Fourier transform of the function } f \text{ and } |\cdot| = \|\cdot\|_2 \text{ denotes the Euclidean norm.}
\]
\[
\text{if } \varphi \text{ satisfies the Strang-Fix conditions (see [15, 32]), a PSI space provides good approximation to } W_2^m \text{ (see [15, 32]), i.e., } \varphi \text{ satisfies the Strang-Fix conditions of order } m \text{ if and only if for all } f \in W_2^m,
\]
\[
\inf_{s \in S^m(\varphi, \Omega)} \|f - s\|_{L^2(\mathbb{R}^2)} = O(h^m).
\]

Particularly, the B-spline \( B_m \) of order \( m \) satisfies the Strang-Fix conditions of order 2 for all \( m \geq 2 \) (see [17]). For more detailed discussions on PSI space, see [15].

Obviously, the union set of the basis functions of \( S_j \) is not linearly independent. Thus the representation of \( g \) is not unique and we want to determine a relatively sparse one, i.e., a representation with as many vanishing coefficients as possible. Every function \( g_j \) \( \in S_j \) can be written as
\[
g_j = \sum_{k \in I_j} X_j^k \varphi(2^{j-1} x/h - k),
\]
where
\[
I_j = \{k \in \mathbb{Z}^2, \supp(\varphi(2^{j-1} \cdot /h - k)) \cap \Omega \neq \emptyset\}.
\]
Let \( X_j \) and \( f \) denote the column vector \( \{X_j^k\}_{k \in I_j} \) and \( \{f_i\}_{1 \leq i \leq N} \), respectively. Then the problem can be formulated as follows:
\[
x_{j, 1 \leq j \leq J} \min \left\| \sum_{j=1}^J A_j X_j - f \right\|_2^2 + \sum_{j=1}^J \lambda_j \|X_j\|_1,
\]
where \( A_j \) is the observation matrix defined by
\[
A_j(i, k) = \varphi(2^{j-1} x_i/h - k), \quad k \in I_j, \quad i = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, J.
\]

Obviously, (2.1) balances the fitting accuracy and the \( l_1 \) norm. In order to achieve a sparse representation, small coefficients are neglected, i.e., after obtaining the solution \( \{X_j\}_{j=1}^J \) of (2.1), we discard the small elements of \( X_j, 1 \leq j \leq J \). Then the final solution only has large values left which indicate important contributions (fluctuations) of the real surface. Furthermore, comparing with the multilevel approximation approach given in [36], our method has the advantages of simplicity. Another important distinction is that it can be interpretable as a sparse strategy for reconstructing scattered data.

### 2.2 The MLASSO model

Model (2.1) is related to the LASSO model in some extent. Recall that the mathematical model of LASSO is
\[
\min_X \|AX - f\|_2^2 + \mu \|X\|_1.
\]

It was proposed originally in [48], and plays a very influential role in variable selection and dimensionality reduction. The Group LASSO (GLASSO) model proposed in [53] solves the convex optimization problem
\[
x_{j, 1 \leq j \leq J} \min \left\| \sum_{j=1}^J A_j X_j - f \right\|_2^2 + \mu \sum_{j=1}^J \|X_j\|_2^2,
\]
where \( \mu > 0 \) is a regularization parameter. The GLASSO model was proposed to perform variable selection on groups of variables for linear regression models. It has many applications in areas such as computer vision, data mining, etc. Meier et al. [39] extended the GLASSO to logistic regression. The GLASSO does not, however, yield sparsity within a group. Moreover, GLASSO suffers from estimation inefficiency and selection inconsistency. To remedy these problems, the adaptive GLASSO method (AGLASSO) is proposed in [49] as
\[
x_{j, 1 \leq j \leq J} \min \left\| \sum_{j=1}^J A_j X_j - f \right\|_2^2 + \mu J \sum_{j=1}^J \|X_j\|_2^2.
\]
Obviously, (2.1) reduces to the LASSO model when $\mu = \lambda_1 = \cdots = \lambda_J$. Moreover, (2.1) acts like the LASSO at the multilevel. Therefore, we denote (2.1) as the multilevel LASSO model (MLASSO). Compared with the AGLASSO model, the MLASSO model considers an additional penalty on the $l_1$ norm instead of $l_2$ norm of the regression coefficient vector, and it produces a selection of variables with sparsity among different levels. It is known that when $l_2$ norm regularization term is applied to the data set, the resulting surface tends to be smooth without sharp discontinuities but have undesirable oscillations near the discontinuities (see [31]). Recently, several surface reconstruction approaches have been proposed to preserve surface discontinuity by replacing $l_2$ regularization using more sophisticated regularization, e.g., the Huber approximation of $l_2$ norm of function derivatives in [47], the local kernel regularization in [26] and the non-local means regularization in [17]. In our method, we use the $l_1$ regularization instead, in order to obtain the relatively sparse solution. The regularization term $\| \cdot \|_1$ in (2.1) penalizes the roughness of the solution.

### 2.3 The parameter selection

The determination of the proper value of $\{\lambda_j\}$ in the MLASSO model is an important problem and depends on the variance of the noise $n$, the properties of $\{A_j\}$ and $\| \cdot \|_1$. An appropriate choice of the regularization parameters is of vital importance for the quality of the resulting estimate and has been the subject of extensive research (see [3]).

In recent years, there has been a growing interest in sophisticated regularization techniques which use multiple constraints as a mean of improving the quality of the solution (see [3]). Among them, only a few papers discussed the choice of multiple regularization parameters. However, most of them discuss the case that the regularization term is the $l_2$ norm. For example, a multi-parameter generalization of heuristic L-curve has been proposed in [3], a knowledge of noise (covariance) structure is required for a choice of parameter in [1, 2, 13], some reduction to a single parameter choice is suggested in [7]. At the same time, the discrepancy principle, which is widely used and known as the first parameter choice strategy proposed in the regularization theory (see [40]), has been discussed in a multi-parameter context in [38]. Of course, there might be many different methods to choose the regularization parameters satisfying certain principle. In fact, for some regularization models, such as the LASSO model and the standard Tikhonov regularization model, the problem of the choice of the regularization parameters has not been completely solved yet. No choice is available for all the models. Basically, different models need different methods to decide the parameters, and even the same one may have different methods to choose, such as [28, 29, 49, 50, 55].

For the MLASSO model, instead of discussing similar methods as listed above, we propose two simple criteria for the choice of the parameters $\{\lambda_j\}_{j=1}^J$ according to the sparsity and the support of the basis functions of the different levels. On one hand, since the length of $X_j$ is less than that of $X_{j+1}$, so if one wants to obtain sparser solution, the parameters of the last several levels should be larger. In particular, choosing the same value for all the parameters, i.e., $\lambda_1 = \cdots = \lambda_J$, obtains the global sparsity in the whole space $S$. On the other hand, the support of the basis functions of the $j$-th level is larger than that of the $(j+1)$-th level, so if one wants smaller support, the parameters of the first several levels should be larger. Hence, the more appropriate choice of the parameters is to make the parameters of the first and last several levels greater than the middle several levels. We can only give such a qualitative guideline, since quantitative guidance for regularization parameters is rarely available. In this way, the solution has smaller support and sparser representation with good approximation accuracy. Numerical results also confirm this selection method as shown in Tables 1–8. Moreover, in contrast with the methods listed above, this choice is easier and cheaper in the sense of computational cost.

Based on the above, the proposed method offers some interesting advantages.

1. Related to the LASSO model, the MLASSO model has a strong statistical background.
2. Compared with the LASSO model, the parameters in the MLASSO model can be chosen differently, thus one can attain high flexibility for the approximation accuracy and the sparsity of the solution.
3. Compared with the AGLASSO model, the MLASSO model can provide the sparsity of the solution.
by choosing different parameters and the distribution of the solution can reflect the large fluctuations of the underlying surface.

(4) Utilizing the ADMM algorithm, the MLASSO model can be efficiently solved.

3 Numerical algorithm

3.1 Algorithm

In this section, we use the ADMM algorithm to solve the minimization model (2.1) for experimental evaluation. It turns out that ADMM is equivalent to or closely related to many famous algorithms, such as the Douglas-Rachford splitting method in PDE literature (see [19, 20, 37]), the Bregman iterative algorithms for $l_1$ problems in signal processing (see [27]) and many others. In particular, we refer to [22, 45, 46, 52] for the relationship between ADMM and the split Bregman iteration scheme which is very influential in the area of image processing. Convergence analysis of the ADMM was given in [5, 21, 30].

In order to solve the MLASSO model (2.1) and guarantee the convergence, we denote $A = (A_1, A_2, \ldots, A_J), \quad X = (X_1^T, X_2^T, \ldots, X_J^T)^T$, 

$\lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_J)$ is a diagonal matrix with $\lambda_j, j = 1, \ldots, J$ as the main diagonal. Then we can rewrite (2.1) as

$$\min_X \|AX - f\|_2^2 + \|\lambda X\|_1.$$  

By introducing an auxiliary variable $d = \lambda X$, we convert the unconstrained minimization problem (3.1) into a constrained one

$$\min_{X, d = \lambda X} \|AX - f\|_2^2 + \|d\|_1.$$  

In this way, the MLASSO model (2.1) is turned into a classical $l_1$ minimization problem. The augmented Lagrangian function of (3.2) is

$$L(X, d, b) = \|AX - f\|_2^2 + \|d\|_1 + \frac{\beta}{2} \|\lambda X - d\|_2^2 + \langle b, \lambda X - d \rangle,$$

where $\beta > 0$ is a parameter of the algorithm. Then by applying the ADMM method, given the initialization $\{X^0, d^0, b^0\}$ and the parameters $\{\lambda_j\}_{j=1}^J$, it results in the following optimization algorithm:

$$\begin{align*}
X^{k+1} &= \arg \min_X \|AX - f\|_2^2 + \frac{\beta}{2} \|\lambda X - d^k + \frac{b^k}{\beta}\|_2^2, \\
d^{k+1} &= \arg \min_d \|d\|_1 + \frac{\beta}{2} \|\lambda X^{k+1} - d + \frac{b^k}{\beta}\|_2^2, \\
b^{k+1} &= b^k + (\lambda X^{k+1} - d^{k+1}).
\end{align*}$$  

First of all, the above algorithm is convergent, since it is just the classical ADMM for two blocks of variables. Secondly, the system can be simply solved. The first step of each iteration in (3.3) is

$$(2A^T A + \beta \lambda^T \lambda) X = 2A^T f + \beta \lambda^T \left( d^k - \frac{b^k}{\beta} \right).$$

This linear system is positive definite and therefore it can be solved by the conjugate gradient method (CG). When the matrix is ill-posed, i.e., its condition number is huge, the convergence rate of the CG will be very slow. Under this case, the preconditioned CG (see [4, 23, 43]) can be used instead to reduce the condition number of the coefficient matrix and improve the convergence speed. The second subproblem has a simple analytical solution based on soft-thresholding operator (see [18]), i.e.,

$$d^{k+1} = T_{\frac{1}{\beta}} \left( \lambda X^{k+1} + \frac{b^k}{\beta} \right),$$
where $T_\theta$ is the soft-thresholding operator defined by

$$T_\theta : x = [x_1, x_2, \ldots, x_M] \rightarrow T_\theta(x) = [t_\theta(x_1), t_\theta(x_2), \ldots, t_\theta(x_M)],$$

where $t_\theta(\xi) = \text{sgn}(\xi)\max\{0, |\xi| - \theta\}$.

The complete description of the algorithm for solving (2.1) is provided as Algorithm 1.

**Algorithm 1** Adapted ADMM for solving the MLASSO model (2.1)

**Require:** $J$, $\{X^0, \hat{a}^0, b^0\}$, $\{\lambda_j\}_{j=1}^4$, $\beta$ and two thresholds $\sigma$ and $\epsilon$.

1. For $k = 0, 1, \ldots$, perform the iteration (3.3) until convergence.
2. Assume $X$ is the solution obtained from Step 1, if $|\hat{X}(i)| < \sigma$, i.e., the absolute value of the $i$-th element of $\hat{X}$ is less than $\sigma$, set $\hat{X}(i) = 0$.
3. The final solution $X$ are $\hat{X}$ after the treatment of Step 2.

![Figure 1](image-url)

**Figure 1** The scattered data points and the corresponding approximation surfaces: (a)–(d) for $f_1$, (e)–(h) for $f_2$, (i)–(l) for $f_3$, (m)–(p) for $f_4$
Figure 2 The distribution of the support for $f_1$ with $J = 3$

Table 1 The $l_0$ norm and the approximation errors for different parameters shown in Figure 2 for $f_1$ with $J = 3$

| (lo(X1), lo(X2), lo(X3)) | Error     | RMS       | Iterations | Time (sec) |
|--------------------------|-----------|-----------|------------|------------|
| [36]                     |           |           |            |            |
| Figure 2(a)              | (24, 64, 193) | 2.1241E−3 | 2.3512E−3 | 3          | 0.0241     |
| Figure 2(b)              | (18, 53, 143) | 2.538E−3  | 4.346E−3  | 17         | 0.0431     |
| Figure 2(c)              | (13, 28, 84) | 3.8215E−3 | 5.6138E−3 | 1,322      | 2.0032     |
| Figure 2(d)              | (4, 41, 88) | 5.143E−3  | 7.3768E−3 | 1,841      | 2.4456     |
| Figure 2(e)              | (15, 22, 49) | 5.3102E−3 | 8.7471E−3 | 2,436      | 3.2345     |
| Figure 2(f)              | (20, 71, 116) | 6.9973E−1 | 6.2737E−1 | 1          | 0.3917     |
| Figure 2(g)              | (24, 61, 181) | 5.0691E−1 | 5.4231E−1 | 1          | 0.2961     |
| Figure 2(h)              | (24, 63, 190) | 4.2005E−1 | 4.6263E−1 | 1          | 0.3524     |

In our numerical experiments, the initializations are

$$X^0 = d^0 = b^0 = 0, \quad \beta = 1$$

and the stopping criteria is

$$\|d^k - \lambda X^k\|_2 \leq \epsilon.$$
Given a test function $f(x, y)$ with $\Omega = [-1, 1] \times [-1, 1]$, we first sample data points with certain noises from it, i.e.,

$$\{(x_i, y_i, f(x_i, y_i) + \varepsilon_i)\}.$$
The error vector \( \mathbf{n} = (\varepsilon_i)_i \), whose entries consist of the pseudorandom values drawn from the standard uniform distribution on the open interval

\[
\left( -\max_i \frac{|f(x_i, y_i)|}{10}, \max_i \frac{|f(x_i, y_i)|}{10} \right).
\]

Then apply different methods to obtain the approximation function \( g \). The difference between \( f \) and \( g \)
Figure 5 The distribution of the support for \( f_2 \) with \( J = 4 \)

Table 4 The \( l_0 \) norm and the approximation errors for different parameters shown in Figure 5 for \( f_2 \) with \( J = 4 \)

|         | \((l_0(\mathbf{X}_1), l_0(\mathbf{X}_2), l_0(\mathbf{X}_3), l_0(\mathbf{X}_4))\) | Error          | RMS            | Iterations | Time (sec) |
|---------|-----------------------------------------------------------------------------|----------------|----------------|-------------|------------|
| [36]    | (22, 63, 195, 673)                                                          | 2.1342E–3      | 3.2496E–3      | 4           | 0.7326     |
| Figure 5(a) | (16, 28, 45, 73)                                                              | 3.0054E–3      | 3.4124E–3      | 378         | 5.0312     |
| Figure 5(b) | (17, 31, 77, 54)                                                              | 2.3321E–3      | 3.0061E–3      | 1,214       | 9.7821     |
| Figure 5(c) | (15, 25, 56, 24)                                                              | 4.1072E–3      | 5.1132E–3      | 2,601       | 13.8871    |
| Figure 5(d) | (4, 61, 18, 0)                                                                | 5.7461E–3      | 6.3242E–3      | 3,261       | 15.3487    |
| Figure 5(e) | (16, 52, 91, 40)                                                              | 1.5002E–1      | 2.1392E–1      | 1           | 5.1902     |
| Figure 5(f) | (20, 64, 185, 471)                                                            | 1.2301E–1      | 2.0342E–1      | 1           | 4.8106     |
| Figure 5(g) | (21, 64, 180, 589)                                                            | 2.0039E–1      | 2.1759E–1      | 1           | 4.7951     |
| Figure 5(h) | (20, 64, 185, 596)                                                            | 3.1492E–1      | 3.1132E–1      | 1           | 4.6283     |

is measured by computing the normalized RMS (root mean square) error (see [36]), i.e.,

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

\[
x_i = -1 + \frac{2i}{M_1 - 1}, \quad i = 0, 1, \ldots, M_1 - 1,
\]

\[
y_j = -1 + \frac{2j}{N_1 - 1}, \quad j = 0, 1, \ldots, N_1 - 1, \quad M_1 = N_1 = 50.
\]
Figure 7 The distribution of the support for $f_3$ with $J = 4$

Table 6 The $l_0$ norm and the approximation errors for different parameters shown in Figure 7 for $f_3$ with $J = 4$

| (l₀(X₁), l₀(X₂), l₀(X₃), l₀(X₄)) | Error | RMS | Iterations | Time (sec) |
|---------------------------------|-------|-----|------------|------------|
| [36] (25, 63, 192, 674)         | 1.6373E−3 | 3.8077E−3 | 4          | 0.7348     |
| Figure 7(a) (16, 47, 105, 38)  | 1.7665E−3 | 5.6282E−3 | 573        | 8.3451     |
| Figure 7(b) (13, 31, 58, 10)   | 2.3442E−3 | 5.6422E−3 | 978        | 13.6259    |
| Figure 7(c) (7, 25, 96, 0)     | 3.6609E−3 | 6.4672E−3 | 854        | 11.7498    |
| Figure 7(d) (0, 51, 36, 0)     | 4.3713E−3 | 8.7693E−3 | 1,588      | 19.8770    |
| Figure 7(e) (14, 18, 19, 0)    | 4.6868E−2 | 5.3092E−2 | 1          | 5.2829     |
| Figure 7(f) (20, 38, 73, 119)  | 5.1103E−2 | 5.5647E−2 | 1          | 5.1456     |
| Figure 7(g) (22, 46, 68, 279)  | 6.9144E−2 | 7.0939E−2 | 1          | 4.3378     |
| Figure 7(h) (23, 34, 101, 328) | 1.0138E−1 | 1.1552E−1 | 1          | 3.6595     |

Moreover, denote

\[
\text{Error} = \sqrt{\frac{\sum_{i=1}^{N} (g(x_i, y_i) - f(x_i, y_i))^2}{N}}
\]

as the fitting error of the given scattered points

\[
\{(x_i, y_i)\}_{i=1}^{N} \subseteq \Omega.
\]
To demonstrate the accuracy of the proposed algorithm, we perform experiments with four functions: a discontinuous function $f_1$, a non-smooth function $f_2$, a smooth function $f_3$ and the Franke test function $f_4$ as follows:

Table 7 The $l_0$ norm and the approximation errors for different parameters shown in Figure 8 for $f_4$ with $J = 3$

| Parameter | $l_0(X_1), l_0(X_2), l_0(X_3)$ | Error | RMS | Iterations | Time (sec) |
|-----------|-------------------------------|-------|-----|------------|------------|
| [36]      | (25, 62, 193)                 | 3.2254E−3 | 5.0711E−3 | 3 | 0.0834 |
| Figure 8(a) | (17, 41, 110) | 3.7421E−3 | 6.3902E−3 | 19 | 1.1642 |
| Figure 8(b) | (15, 34, 57) | 4.0324E−3 | 6.2097E−3 | 938 | 1.7439 |
| Figure 8(c) | (9, 38, 46) | 4.5021E−3 | 8.1093E−3 | 1,463 | 2.4762 |
| Figure 8(d) | (15, 29, 33) | 6.2043E−3 | 8.0072E−3 | 2,230 | 2.7803 |
| Figure 8(e) | (21, 45, 126) | 2.0414E−1 | 6.0056E−1 | 1 | 0.7439 |
| Figure 8(f) | (24, 60, 161) | 2.1369E−1 | 3.1057E−1 | 1 | 0.5648 |
| Figure 8(g) | (23, 57, 166) | 3.2451E−1 | 2.2731E−1 | 1 | 0.4846 |
| Figure 8(h) | (24, 58, 179) | 5.3021E−1 | 2.1237E−1 | 1 | 0.4218 |
Table 8 The $l_0$ norm and the approximation errors for different parameters shown in Figure 9 for $f_4$ with $J = 4$

|       | $(l_0(X_1), l_0(X_2), l_0(X_3), l_0(X_4))$ | Error     | RMS        | Iterations | Time (sec) |
|-------|--------------------------------------------|-----------|------------|------------|------------|
| [36]  | (25, 63, 194, 675)                         | 4.1124E−4 | 3.2017E−3 | 4          | 2.1406     |
| Figure 9(a) | (14, 34, 84, 108)                     | 5.0121E−4 | 5.0512E−3 | 687        | 12.1132    |
| Figure 9(b) | (15, 33, 48, 31)                      | 1.1897E−3 | 4.6645E−3 | 1.152      | 18.4120    |
| Figure 9(c) | (13, 27, 52, 8)                        | 4.1321E−3 | 7.1452E−3 | 1.333      | 18.2461    |
| Figure 9(d) | (12, 22, 70, 6)                        | 4.2134E−3 | 7.6731E−3 | 2.231      | 28.1121    |
| Figure 9(e) | (22, 47, 126, 145)                     | 1.5782E−1 | 1.7287E−1 | 1          | 5.1631     |
| Figure 9(f) | (24, 58, 163, 491)                      | 2.2056E−1 | 2.2142E−1 | 1          | 5.4315     |
| Figure 9(g) | (24, 58, 164, 557)                     | 3.0349E−1 | 3.4397E−1 | 1          | 4.3341     |
| Figure 9(h) | (25, 60, 160, 524)                     | 5.2570E−1 | 5.3329E−1 | 1          | 3.9764     |

$$f_1(x, y) = \begin{cases} 
\frac{x^2 y}{x^2 + y^2}, & x^2 + y^2 \leq 1, \\
\frac{x+y}{\sqrt{x^2 + y^2}}, & x^2 + y^2 > 1, 
\end{cases} \quad f_2(x, y) = \begin{cases} 
\frac{xy}{\sqrt{x^2 + y^2}}, & x^2 + y^2 \leq 1, \\
\frac{xy}{x^2 + y^2}, & x^2 + y^2 > 1, 
\end{cases}$$
\[
f_3(x, y) = \frac{1.25 + \cos(5.4y)}{6 + 6(3x - 1)^2},
\]
\[
f_4(x, y) = 0.75 \exp \left[ -\frac{(9x - 2)^2 + (9y - 2)^2}{4} \right] + 0.75 \exp \left[ -\frac{(9x + 1)^2}{49} - \frac{(9y + 1)^2}{10} \right] + 0.5 \exp \left[ -\frac{(9x - 7)^2 + (9y - 3)^2}{4} \right] - 0.2 \exp [- (9x - 4)^2 - (9y - 7)^2].
\]

For the numerical implementation in this paper, we employ the 2D tensor product quadratic B-spline as the function \(\varphi\). We show that such a simple system with Algorithm 1 can be used to effectively reconstruct surface of sparse representation from a scattered data set. The choices of the threshold \(\sigma\) and \(\epsilon\) can be chosen according to the accuracy and sparsity. We consider the four functions with \(\sigma = 10^{-3}\), \(\epsilon = 10^{-4}\), \(N = 900\) and the 900 scattered points are chosen randomly. Moreover, we compare Algorithm 1 with the basic MBA algorithm presented in [36] under the termination condition \(|\Delta \theta_j| \leq O(10^{-3})\) and the AGLASSO model, both applying the same thresholding \(\sigma = 10^{-3}\) as Algorithm 1 to their results.

We experiment the above three methods for \(J = 3\) and \(J = 4\) starting from level 1 with 5\(\times\)5 biquadratic B-spline functions, respectively. Figure 1 shows the scattered data points and the corresponding approximations of the four functions with Algorithm 1, the method in [36] and the AGLASSO model, respectively. Figures 2–9 illustrate the distribution of the support of the B-spline functions with nonzero coefficients for \(J = 3\) and \(J = 4\), respectively. In addition, the rectangles from big to small denote the supports of the B-splines corresponding to \(X_1, X_2, X_3\) and \(X_4\), respectively. Moreover, Tables 1–8 give the approximation accuracy, the iterations, the running time and the \(l_0\) norm of the solution with different parameters for \(f_1 - f_4\), where the length of \(X_1, X_2, X_3\) and \(X_4\) is 25, 64, 196 and 625, respectively. In addition, all our calculations are done in Matlab on a laptop with Inter Core i7 (2.90GHZ) CPU and 8.0G RAM.

**Remark 3.1.** The numerical results demonstrate that Algorithm 1 and the method in [36] have almost the same approximation errors, while Algorithm 1 obtains the sparse solution. Compared with the AGLASSO model, Algorithm 1 provides the sparser solutions with less error, though more iterations and more time.

Through the first step of Algorithm 1, we can obtain an approximation solution of the MLASSO model with no sparsity and the solution has some big components and some small ones which reflect different importance and contribution. Then by the second step, we throw out small components of the computed solution which means we keep only the important ones with great contribution to the solution. Therefore, by choosing appropriate regularization parameters, the final solution can indicate the important parts we are interested in and identify the important features within the selected levels simultaneously of the exact surface since they are all determined by the big components. Experiment results verify this conclusion.

Taken together, Algorithm 1 can reconstruct the test functions reasonably with a sparse representation within a few levels by choosing some appropriate regularization parameters.

### 4 Conclusion

This paper presents an approach for scattered data fitting using the PSI space and the \(l_1\) regularization. It is concluded into the MLASSO model which allows us to balance the accuracy and the sparsity of the fitting surface. The model can be solved using Algorithm 1 with the ADMM algorithm. Numerical examples demonstrate that compared to the basic MBA algorithm in [36] and the AGLASSO model, the MLASSO model provides an efficient, sparse, flexible and reasonable solution. Moreover, the distribution of the basis functions of the sparse solution can identify the regions of the underlying surface where large fluctuations occur.

**Acknowledgements** This work was supported by National Natural Science Foundation of China (Grant Nos. 11526098, 11001037, 11290143 and 11471066), the Research Foundation for Advanced Talents of Jiangsu University (Grant No. 14JJDG034), the Natural Science Foundation of Jiangsu Province (Grant No. BK20160487) and the Fundamental Research Funds for the Central Universities (Grant No. DUT15LK44).
References

1. Bauer F, Ivanyszyn O. Optimal regularization with two interdependent regularization parameters. Inverse Problems, 2007, 23: 331–342
2. Bauer F, Pereverzev S V. An utilization of a rough approximation of a noise covariance within the framework of multi-parameter regularization. Int J Tomogr Stat, 2006, 4: 1–12
3. Belge M, Kilmer M E, Miller E L. Efficient determination of multiple regularization parameters in a generalized L-curve framework. Inverse Problems, 2002, 18: 1161–1183
4. Benzi M. Preconditioning techniques for large linear systems: A survey. J Comput Phys, 2002, 182: 418–477
5. Boyd S, Parikh N, Chu E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn, 2010, 3: 1–122
6. Bozzi M, Rossini M. Testing methods for 3D scattered data interpolation. Monogr Real Acad Ci Exact Fís-Quím Nat Zaragoza, 2002, 20: 111–135
7. Brezinski C, Redivo-Zaglia M, Rodriguez G, et al. Multi-parameter regularization techniques for ill-conditioned linear systems. Numer Math, 2003, 94: 203–228
8. Candès E J, Romberg J, Tao T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Trans Inform Theory, 2006, 52: 489–509
9. Candès E J, Romberg J, Tao T. Stable signal recovery from incomplete and inaccurate measurements. Comm Pure Appl Math, 2006, 59: 1207–1223
10. Candès E J, Tao T. Decoding by linear programming. IEEE Trans Inform Theory, 2005, 51: 4203–4215
11. Candès E J, Tao T. Near-optimal signal recovery from random projections: Universal encoding strategies. IEEE Trans Inform Theory, 2006, 52: 5406–5425
12. Castaño D, Kunoth A. Adaptive Fitting of Scattered Data by Spline-Wavelets. Vanderbilt: Vanderbilt University Press, 2003
13. Chen Z, Lu Y, Xu Y, et al. Multi-parameter Tikhonov regularization for linear ill-posed operator equations. J Comput Math, 2008, 26: 37–55
14. Cohen A, Dahmen W, DeVore R. Compressed sensing and best k-term approximation. J Amer Math Soc, 2009, 22: 211–231
15. De Boor C, Ron A. Fourier analysis of the approximation power of principal shift-invariant spaces. Constr Approx, 1992, 8: 427–462
16. DeVore R A, Jawerth B, Lucier B J. Surface compression. Comput Aided Geom Design, 1992, 9: 219–239
17. Dong B, Shen Z. Wavelet frame based surface reconstruction from unorganized points. J Comput Phys, 2011, 230: 8247–8255
18. Donoho D. De-noising by soft-thresholding. IEEE Trans Inform Theory, 1995, 41: 613–627
19. Douglas J, Rachford H H. On the numerical solution of heat conduction problems in two and three space variables. Trans Amer Math Soc, 1956, 82: 421–439
20. Eckstein J. Splitting methods for monotone operators with applications to parallel optimization. PhD Thesis. Cambridge: Massachusetts Institute of Technology, 1989
21. Eckstein J, Bertsekas D. On the Douglas-Rachford Splitting Method and the Proximal Point Algorithm for Maximal Monotone Operators. Mathematical Programming, vol. 55. Amsterdam: North Holland, 1992
22. Esser E. Applications of Lagrangian-based alternating direction methods and connections to split Bregman. CAM Report, 09-31. Https://doi.org/10.1007/s00453-014-9909-1, 2009
23. Fasi M, Langou J, Robert Y, et al. A backward/forward recovery approach for the preconditioned conjugate gradient method. J Comput Sci, 2016, 17: 522–534
24. Forsey D R, Bartels R H. Surface fitting with hierarchical splines. ACM Trans Graph, 1995, 14: 134–161
25. Gabay D, Mercier B. A dual algorithm for the solution of nonlinear variational problems via finite element approximations. Comput Math Appl, 1976, 2: 17–40
26. Gijbels I, Lambert A, Qiu P. Edge-preserving image denoising and estimation of discontinuous surfaces. IEEE Trans Pattern Anal Mach Intell, 2006, 28: 1075–1087
27. Goldstein T, Osher S. The split Bregman method for $l_1$-regularized problems. SIAM J Imaging Sci, 2009, 2: 323–343
28. Hanke M, Scherer O. Inverse problems light: Numerical differentiation. Amer Math Monthly, 2001, 108: 512–521
29. Hao D N, Chuong L H, Lesnic D. Heuristic regularization methods for numerical differentiation. Comput Math Appl, 2012, 63: 816–826
30. He B, Yuan X. On the $O(\frac{1}{k^\gamma})$ convergence rate of Douglas-Rachford alternating direction method. SIAM J Numer Anal, 2012, 50: 700–709
31. Ji H, Shen Z, Xu Y H. Wavelet frame based scene reconstruction from range data. J Comput Phys, 2010, 229: 2093–2108
32. Jia R Q. The Toeplitz theorem and its applications to approximation theory and linear PDE’s. Trans Amer Math Soc, 1995, 347: 2585–2594
33 Johnson M J, Shen Z, Xu Y H. Scattered data reconstruction by regularization in B-spline and associated wavelet spaces. J Approx Theory, 2009, 159: 197–223
34 Kunis S, Rauhut H. Random sampling of sparse trigonometric polynomials II-orthogonal matching pursuit versus basis pursuit. Found Comput Math, 2008, 8: 737–763
35 Lee B G, Lee J J, Kwon K R. Quasi-interpolants based multilevel B-spline surface reconstruction from scattered data. In: Computational Science and Its Applications. Lecture Notes in Computer Science, vol. 3482. Berlin-Heidelberg: Springer, 2005, 1209–1218
36 Lee S Y, Wolberg G, Shin S Y. Scattered data interpolation with multilevel B-splines. IEEE Trans Vis Comput Graph, 1997, 3: 229–244
37 Lions P L, Mercier B. Splitting algorithms for the sum of two nonlinear operators. SIAM J Numer Anal, 1979, 16: 964–979
38 Lu S, Pereverzer S V. Multi-parameter regularization and its numerical realization. Numer Math, 2011, 118: 1–31
39 Meier L, Van de Geer S, Bühlmann P. The group lasso for logistic regression. J R Stat Soc Ser B Stat Methodol, 2008, 70: 53–71
40 Phillips D. A technique for the numerical solution of certain integral equation of the first kind. J Assoc Comput Mach, 1962, 9: 84–97
41 Rauhut H. Random sampling of sparse trigonometric polynomials. Appl Comput Harmon Anal, 2007, 22: 16–42
42 Rauhut H. Stability results for random sampling of sparse trigonometric polynomials. IEEE Trans Inform Theory, 2008, 54: 5661–5670
43 Sajo-Castelli A M, Fortes M A, Raydana M. Preconditioned conjugate gradient method for finding minimal energy surfaces on Powell-Sabin triangulations. J Comput Appl Math, 2014, 268: 34–55
44 Schmitt F J M, Barsky B A, Du W H. An adaptive subdivision method for surface fitting from sampled data. Comput Graph, 1986, 20: 179–188
45 Setzer S. Split bregman algorithm, douglas-rachford splitting and frame shrinkage. In: Scale Space and Variational Methods in Computer Vision. Lecture Notes in Computer Science, vol. 5567. Berlin-Heidelberg: Springer, 2009, 464–476
46 Steidl G, Teuber T. Removing multiplicative noise by Douglas-Rachford splitting methods. J Math Imaging Vision, 2010, 36: 168–184
47 Stevenson R, Schmitz B, Delp E. Discontinuity preserving regularization of inverse visual problems. IEEE Trans Syst Man Cybernet, 1994, 3: 455–469
48 Tibshirani R. Regression shrinkage and selection via the Lasso. J R Stat Soc Ser B Stat Methodol, 1996, 58: 267–288
49 Wang H, Leng C. A note on adaptive group lasso. Comput Statist Data Anal, 2008, 52: 5277–5286
50 Wang Y B, Hon Y C, Cheng J. Reconstruction of high order derivatives from input data. J Inverse Ill-Posed Probl, 2009, 14: 205–218
51 Wang Z, Yu J, Xie X. An improved algorithm for surface fitting based on B-spline function. In: 2011 Fourth International Conference on Information and Computing. Washington: IEEE Computer Society, 2011, 80–82
52 Wu C, Tai X-C. Augmented lagrangian method, dual methods and split-bregman iterations for ROF, vectorial TV and higher order models. SIAM J Imaging Sci, 2010, 3: 300–339
53 Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. J R Stat Soc Ser B Stat Methodol, 2006, 68: 49–76
54 Yuan X M. Alternating direction method for covariance selection models. J Sci Comput, 2012, 51: 261–273
55 Zou H. The adaptive lasso and its oracle properties. J Amer Statist Assoc, 2006, 101: 1418–1429