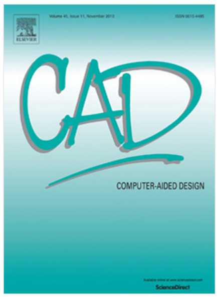
Accepted Manuscript



An Economical Representation of PDE Solution by using Compressive Sensing Approach

Hongmei Kang, Ming-Jun Lai, Xin Li

PII: S0010-4485(19)30195-2
DOI: https://doi.org/10.1016/j.cad.2019.05.021
Reference: JCAD 2707

To appear in: *Computer-Aided Design*

An Economical Representation of PDE Solution by using Compressive Sensing Approach

Hongmei Kang

School of Mathematical Sciences, Soochow University, No.1 Shizi Street, Suzhou 215006, China

Ming-Jun Lai*

Department of Mathematics, The University of Georgia, Athens, GA 30602, United States

Xin Li*

School of Mathematical Sciences, University of Science and Technology of China, No.96 Jinzhai Road, Hefei 230026, China

Abstract

We introduce a redundant basis for numerical solution to the Poisson equation and find a sparse solution to the PDE by using a compressive sensing approach. That is, we refine a partition of the underlying domain of the PDE several times and use the multi-level nested spline subspaces over these refinements to express the solution of the PDE redundantly. We then use a compressive sensing algorithm to find an economical representation of the spline approximation of the PDE solution. The number of nonzero coefficients of an economical representation is less than the number of the standard spline representation over the last refined partition, i.e. finite element solution while we will show that the error of the spline approximation with an economical representation is the same to the standard FEM solution. This approach will be useful, e.g. in the situation when the PDE solver has a much powerful computer than the users of the solution.

Keywords: Isogeometric analysis, Compressive sensing, Sparse solution, PDEs, Economical representation.

1. Introduction

In the standard weak formulation of the Poisson equation, the numerical solution is searched in a finite dimensional Sobolev space by solving the squared system of linear equations. As the exact solution may change rapidly over one subregion and slowly over the other, in order to achieve a higher accuracy, traditionally one has to refine the underlying partition/mesh many times. In this way, the dimension of the solution space increases significantly. Thus, one needs to use a lot of coefficients (more than necessary) to approximate the PDE solution. A straightforward way to correct this problem is to use the adaptive finite element method (AFEM)(cf. [24, 25, 26]). That is, one solves the PDE based on a reasonably refined partition/mesh together with adding locally refined basis functions. Indeed, one compares the right-hand side associated with the numerical solution with the exact right-hand side to induce an posterior error estimate. If the error is not within tolerance, one adds a local refinement in the partition/mesh according to a certain refinement rule and then repeats the computational procedure again.

Isogeometric analysis (IGA for short) was introduced as a new approach for solving PDEs (cf. [28, 29]). The essence of IGA is a collection of methods that uses splines or some of their extensions as approximation spaces which are then used for solving PDEs numerically. There has been a lot of work on developing different kinds of splines used in IGA. Some of them can be found at [29, 31, 32, 33, 34, 35] and the references therein. And most of these splines are locally refinable splines thus they support an adaptive refinement framework when they are used in IGA (cf. [23, 27, 36]). The refinement is performed on elements according to a certain refinement rule based on posterior error estimates. And the posterior error estimates generally come from some existing estimates in FEA. See [24, 25, 26].

It can be seen the above adaptive refinement is a greedy and strategic refinement, thus there are always a lot of redundant elements to be refined. Furthermore, some kinds of splines are defined over meshes with specific structures, so extra elements are refined to satisfy the requirement of mesh structures. For example, the local refinement of T-splines [23] needs extra elements to keep the exact geometry, analysis suitable T-splines [30] and PHT-splines [33] need to satisfy the constraints of analysis-suitable T-meshes and hierarchical meshes respectively. Certainly these specific mesh structures destroy the original uniform structure of partitions. Different problems require different refined meshes. In this sense, the traditional adaptive method has some unsatisfactory side effects. Therefore it is necessary to introduce a new adaptive method for selecting basis functions globally.

In this paper, we propose to use a sparse model to find a solution with an economic number of nonzero coefficients to the PDE with the similar accuracy as the standard weak solution. More precisely, we shall use uniformly refined partitions. The basis functions on different levels are collected together to form a redundant finite dimensional Sobolev space. From this redundant space, we choose the minimal number of basis functions to approximate the solution for the same accuracy as the standard finite element method based on the spline space over the finest partition, i.e. the last level of refinement of the partition of the domain. For example, when using bicubic spline functions over the 6th level of refinement of the unit square Ω to approximate

*Corresponding author

Email addresses: `mjlai@uga.edu` (Ming-Jun Lai*), `lixustc@ustc.edu.cn` (Xin Li*)

the solution of a Poisson equation, if the solution happens to be x^3y^3, one can simply use the bicubic spline functions over the first level of refinement of Ω to represent the solution. The proposed method will find such a simple representation with a much smaller number of nonzero coefficients (the coefficients for the basis functions over the first level of refinement) than the solution from the standard FEM based on the basis functions over the 6th level of refinement of Ω. For another example, if the solution u to a Poisson equation has a constant value over a subdomain with large area inside Ω, then u can have a sparser representation than the standard FEM solution since u can be represented using fewer basis functions over the previous levels of refined partitions than that of the basis functions over the last refined partition. In general, if a solution can be well approximated by using spline functions over the $(n-1)$th refined partition within the tolerance ϵ, our proposed method can find this solution when using all combined spline functions over all the kth refined partitions, $1 \leq k \leq n$. The solution will have a much fewer nonzero coefficients than the weak-form spline solution over the nth refined partition.

The proposed computation can be done by projecting the basis functions in all levels of refined spaces into the last refined space via a Galerkin projection. In this way, we obtain a rectangular stiffness matrix. This stiffness matrix multiplied by an unknown vector equals the projection of the right term in the last refined space. This results in a linear system of rectangular size. Then we find the sparse solution of this rectangular linear system. These concepts will be explained in detail in a later section. We focus on B-splines/tensor product B-splines currently to illustrate the ideas proposed in this paper. Certainly, the ideas can be extended to other kinds of splines for numerical solutions of various partial differential equations.

There are many computational algorithms developed for sparse solutions of long rectangular linear system which is an underdetermined linear system. However, most of these algorithms can only find solutions with a small sparsity, e.g. 30–40% nonzero entries of the solution. We have to experiment many approaches to see which one performs the best. After a rather thorough investigation, certainly not an exhausted search, we found a good approach which is based on a mix of two computational algorithms which can find more 50% nonzero entries of a solution. This new algorithm will be presented in the next section together with some convergence analysis and a summary of sparsity recovery via many well-known computational algorithms will be given to demonstrate that our proposed algorithm works the best. With this tool, we tackle the problem of finding most economic solution to the PDE.

Another advantage of our method over any local refinement T-spline schemes is that the proposed method does not create any T-junction points and has the simplification of evaluation. Indeed, suppose we use refinement level $n = 6$. A sparse solution whose nonzero coefficients will be decoupled into 6 groups to have 6 spline functions over the 6 nested refinements. Thus, we use de Boor's evaluation for 6 spline functions and then add these values together to have the value for the sparse solution. Also, the proposed method is more economic than any triangulation based adaptive finite element method since it produces a set of coefficients as well as a set of particular triangulation (a set of vertices and a list of triangulation) which usually consists of a large data file. One major disadvantage of the proposed method is the computational time, which is much slower than the standard FEM/adaptive FEM when the refined level is large due to the nature of the nonlinear iterative steps. The topic is certainly worthy studying how to improve its computational efficiency. On the other hand, if the person computing the solution to his/her PDE has a much powerful computer than the users of the solution, then this method can be useful. Also, if the solution will be used many times, it is recommended to have a sparse solution form once for all.

The remainder of this section is organized as follows. In section 2, we explain an economical representation of the Poisson Equation based on a sparse model. In section 3, an error estimate of the sparse solution from our proposed method is proved to have the similar error estimate of the classic FEM solution. In section 4, several numerical examples solved by the proposed method are demonstrated. Section 5 concludes the paper with a summary and future work.

2. An Economic Representation of PDE Solution

In this section, we propose a method to find an economic representation of the spline solution to the PDE based on the compressive sensing approach. Mainly, we shall use the greedy and l_1 minimization algorithm to help find an approximation to the PDE solution with fewer nonzero spline coefficients.

2.1. Discretization of PDEs

Consider the Poisson equation:

$$\begin{aligned} -\Delta u &= f, \quad \Omega \subset \mathbb{R}^2 \\ u &= g, \quad \text{on } \partial\Omega \end{aligned} \tag{1}$$

where Ω is a bounded domain with Lipschitz boundary $\partial\Omega$. Let $\mathbf{G}$ be the geometric mapping which maps $[0,1]^2$ to Ω with smooth inverse, that is

$$\mathbf{G} : \xi \in [0,1]^2 \to (x, y) \in \Omega.$$

See Fig. 1 for a reference. To solve the Poisson equation over Ω using the weak formulation, we have

$$a(u, v) := \langle \nabla u, \nabla v \rangle = \langle f, v \rangle, \ \forall v \in H_0^1(\Omega), \tag{2}$$

Let us explain the weak formulation more precisely.

$$\langle f, v \rangle = \int_{[0,1]^2} f(\mathbf{G}(\xi)) v(\mathbf{G}(\xi)) \sqrt{\det(J^\top J)} \mathrm{d}\xi, \tag{3}$$

where $J = \nabla_\xi \mathbf{x}$ with $\mathbf{x} = (x, y)$ and similarly,

$$\langle \nabla u, \nabla v \rangle = \int_{[0,1]\times[0,1]} \nabla_\xi u(\mathbf{G}(\xi)(J^\top J)^{-1} \nabla_\xi v(\mathbf{G}(\xi)) \sqrt{\det(J^\top J)} \mathrm{d}\xi. \tag{4}$$

2.2. The Sparse Model

We use a hierarchy of spline spaces to approximate the solution $u(\mathbf{G}(\xi))$. Let $S_n, n \geq 1$ be a sequence of nested finite dimensional subspaces of $H_0^1(\Omega)$, i.e.

$$S_1 \subset S_2 \subset \cdots \subset S_n.$$

For example, we can choose a nested triangulation $\triangle_n$ of Ω by the standard uniform refinement strategy and let $S_n = S_d^1(\triangle_n)$ be the bivariate spline space of degree d and smoothness 1 over triangulation $\triangle_n$. For a theory of splines, see [14] for more detail. See spline implementations in [1] and [20]. For another example, one can use the nested tensor product B-spline spaces starting

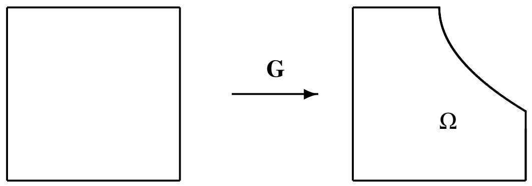


Figure 1: The Geometrical Map **G** from a patch to the physical domain Ω.

with a rectangular parametric domain. This is the approach we adopt in this paper.

Write $S_j = \text{span}\{\phi_{j,1}, \cdots, \phi_{j,N_j}\}$, where N_j is the dimension of S_j and $\phi_{j,1}, \cdots, \phi_{j,N_j}$ are B-spline basis functions spanning the spline space S_j, $j = 1, 2, \cdots, n$. Denote

$$\Phi_j = [a(\phi_{j,1}, \phi_{n,i}), a(\phi_{j,2}, \phi_{n,i}), \cdots, a(\phi_{j,N_j}, \phi_{n,i})]_{i=1,\cdots,N_n}, \quad (5)$$

as the rectangular stiffness matrix of size $N_n \times N_j$ for $j = 1, \cdots, n$, where $a(\phi_{j,1}, \phi_{n,i}) = \langle \nabla\phi_{j,1}, \nabla\phi_{n,i}\rangle$ for all $j = 1, \cdots, n, i = 1, \cdots, N_n$. Let Φ be the basis functions on all levels,

$$\Phi = [\Phi_1, \Phi_2, \cdots, \Phi_n]$$

and $\mathbf{b} = [\langle f, \phi_{n,1}\rangle, \cdots, \langle f, \phi_{n,N_n}\rangle]^\top$. We look for solution $\mathbf{x} \in \mathbb{R}^{N_1+\cdots+N_n}$ such that

$$\min \{ \|\mathbf{x}\|_0, \quad \Phi\mathbf{x} = \mathbf{b} \}, \quad (6)$$

where $\|\mathbf{x}\|_0$ is the number of nonzero entries of $\mathbf{x}$, Φ is of size $N_n \times (N_1 + \cdots + N_n)$ and $\mathbf{b}$ is of size $N_n \times 1$. Let $\mathbf{x_b}$ be the sparse solution of (6) with $\|\mathbf{x_b}\|_0 < N_n$. Write

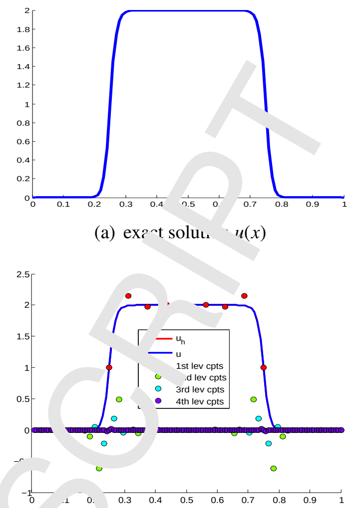
$$\Psi = [\phi_{1,1}, \cdots, \phi_{1,N_1}, \phi_{2,1}, \cdots, \phi_{2,N_2}, \cdots, \phi_{n,1}, \cdots, \phi_{n,N_n}]$$

and let $u^* = \Psi\mathbf{x_b}$. Then $u^* \in S_n$ and satisfies

$$\langle \nabla u^*, \nabla\phi_{n,j}\rangle = \langle f, \phi_{n,j}\rangle, \quad \forall j = 1, \cdots, N_n.$$

By the uniqueness of the weak solution, u^* is the weak solution in S_n for (1). However, the number of nonzero coefficients is the smallest. In this way, we can find the most economical representation of the weak solution in the nested subspace sequence $\{S_1, S_2, \cdots, S_n\}$.

We shall present one example of 1D Poisson equations to show the above sparse model. The exact solution $u(x) = -tanh(((x - 0.5)^2 - r)/sr) + 1.0$, $x \in [0, 1]$, where $r = 0.0625$, $sr = 0.01$. f is derived from (1). This $u(x)$ has a sharp gradient around $x = 0.2$ and $x = 0.8$, as shown in Fig. 2(a). In order to recover this sharp gradient and have an economical representation, more knots should be located at these two places and less knots are located at the rest domain, when B-splines are applied in solving (1). Fig. 2(b) shows the numerical solution u_h solved with $n = 4$, where the coefficients for each levels are marked by different colors. It can be seen that u_h has much more non-vanishing coefficients around the sharp gradient, while only the coefficients on the first level are non-vanishing on the flat domain. In Table 1, N is equal to $N_1 + \cdots + N_n$, and sparsity here refers to the number and the percentage of nonzero coefficients. For each n, the sparsity of our method is much smaller than N_n and the L_2-norm error solved by our method is the same as that of FEM on each level.



(a) exact solution $u(x)$

(b) numerical solution $u_h(x)$

Figure 2: 1D numerical solution solved by (6) with $n = 4$ and coefficients on different levels.

2.3. Computational Algorithms

Sparse solutions of underdetermined linear system has been actively studied in the last fifteen years. Commonly, the minimization (6) is replaced by

$$\min\{ \|\mathbf{x}\|_1, \quad \Phi\mathbf{x} = \mathbf{b}\}, \quad (7)$$

where $\|\mathbf{x}\|_1$ is the ℓ_1 norm of vector $\mathbf{x} = (x_1, \cdots, x_N)^\top$ with $\|\mathbf{x}\|_1 = \sum_{j=1}^N |x_j|$, and $N = N_1 + \cdots + N_n$. This problem can also be recast as

$$\min\{\|\Phi\mathbf{x} - \mathbf{b}\|^2, \|\mathbf{x}\|_0 \le s\} \quad (8)$$

for a guessed sparsity s. There are many computational algorithms available based on convex minimization and non-convex minimization approaches. We refer to [21], [17], [7], [8], [6], [4], [2], [3], [13], [10], [16], [18], [19], [12], [22], [9], [15], and etc.. Most of them work well when the sparsity of a sparse solution is small. However, the sparsity of a PDE solution may not be very small in general. A straight-forward application of these numerical algorithms does not work well in finding the economical reprsentation of the PDE solution. In particular, when a PDE in the 2D and 3D settings, the solution may not have a small sparsity. Nevertheless, various ideas behind these algorithms provide us hints for finding a good new efficient way. We have experimented many approaches mentioned above and find a good one for economic representation of the PDE solution.

	Our method			FEM	
n	N	sparsity	$\|u - u_h\|_{L^2}$	N_n	$\|u - u_h\|_{L^2}$
2	54	27(50%)	2.9137e-2	35	2.9139e-2
3	121	48(39.7%)	1.7916e-3	67	1.7921e-3
4	252	74(29.4%)	3.4741e-4	131	3.4695e-4
5	511	145(28.4%)	1.5292e-6	259	1.4984e-6
6	1026	345(33.6%)	8.8701e-8	515	8.5733e-8

Table 1: number of non-vanishing coefficients solved with different n of the 1D example.

The MATLAB version of our approach is concluded in **Algorithm 1**. The basic idea is to use the levels of the magnitude of the entries in the sparse solution vector when finding the sparse solution. That is, we first compute the largest entries (top 87%) of the sparse solution vector. Then we use 0.1 to put the columns of the sensing matrix associated with the largest entries in a less important part to have a modified sensing matrix so that we can compute the next batch of the entries of the sparse solution vector. The parameters 0.1 and 0.87 can be adjusted. The values 0.1 and 0.87 were chosen based on a large amount of our experiments.

The main computation of Algorithm 1 is done by **L1min** which is a revised version of the code discussed in [17] and is enclosed in the Appendix. The original **L1min** is used for the L_1 minimization for scattered data interpolation in [17]. Here we rewrote it to find the sparse solution of underdetermined linear system instead. The main ingredient is the interior point method to solve the linear programming problem which is equivalent to finding the solution to L_1 minimization problem.

Algorithm 1 x = lai2012(A,y)

Input: a matrix A of size $m \times n$ ($m < n$), a vector y of size $m \times 1$.
Output: a vector x of size $n \times 1$.

```
NIt=3;
[m,n]=size(A); AW=A; W=ones(n,1); iv=zeros(n,1);
j0=1; x = iv;
for i = 1:NIt
   x = L1min(AW,y,1e-9,iv);
   x = x./W;
   if 1 == i
      [Mx  j0] = max(abs(x));
   end
   Mx = Mx*0.87;
   W = (abs(x) > Mx)/10 + (abs(x)≤ Mx);
   for j = 1:n
      AW(:,j) = A(:,j)/W(j);
   end
end
```

The Algorithm 1 is different from the algorithm described in [13] in the sense that we use L1min instead of the well-known magicL1. The major reason to use L1min is the better performance. Let us illustrate by numerical experiments. Consider a matrix A of size 64×128 with uniform random variables as its entries. Let $\mathbf{x_b}$ be a vector of sparsity s with nonzero entries which are uniform random values. For $y = A\mathbf{x_b}$, we use Algorithm 1 to solve $\mathbf{x}^*$ and measure the maximum norm. For simplicity, we use Gaussian random matrices of 64×128 with sparsity from $1 - 45$. We test Algorithm 1 with L1min replaced by magicL1 from Candés webpage (called **KP** in short, see [13]), Algorithm 1 (called **Lai** in short), iteratively reweighted ℓ_1 minimization (called **CWB** in short, see [?]), the **FISTA** algorithm (cf. [2]), hard thresholding pursuit algorithm (called **HTP** in short, see [10]). In addition, **GAMP** stands for the generalized message passing algorithm (cf. [37]). The method GAMP is very special, only working for Gaussian sensing matrices. We have 500 independent runs of each recovery for sparsity $s = 1, \cdots, 45$. The percentage of recovery (or frequency of successes of recovery) is shown in Fig. 3, where $x-$axis represents the sparsity s and $y-$axis represents frequency of successes of recovery during experiments. Similar performance can be seen for the uniform

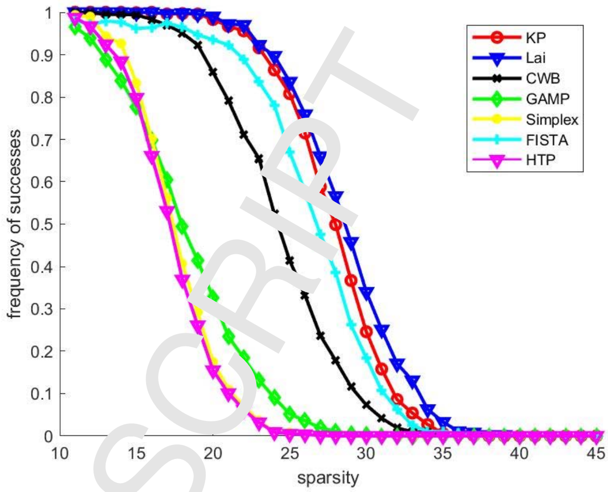


Figure 3: Numerical Results based on Algorithm 1.

random sensing matrices. We omit the graph for convenience. From Fig. 3, our program Algorithm 1 is able to recover sparse solutions with nonzero entries more than 50% of the entire entries with very high frequency.

We now give a convergence analysis of Algorithm 1. It is easy to see that the algorithms above are equivalent to solving

$$\mathbf{x}^{(k)} := \arg\ \min\{(W^{(k-1)})^\top |\mathbf{x}|, \quad A\mathbf{x} = y\}, \tag{9}$$

where $|\mathbf{x}| = (|x_1|, |x_2|, \cdots, |x_n|)^\top$ denotes the absolute value of $\mathbf{x} = (x_1, x_2, \cdots, x_n)^\top$. Note that $W^{(k-1)}$ divides the indices of $\mathbf{x}$ into two groups: one is the less important portion of the indices collected in $J_{M_{k-1}}$ which is scaled by 0.1 and the other is more important portion of the indices denoted by $I_{M_{k-1}}$. Here M_{k-1} is equal to the variable Mx at step k iteration in Algorithm 1. Heuristically, in each step the larger components of the iterative solution $\mathbf{x}^{(k)}$ are found and moved in the less important group while the smaller components of $\mathbf{x}^{(k)}$ are needed to compute more accurately and hence are moved to the important group.

To study the convergence of the iterative solutions $\mathbf{x}^{(k)}$, we first show that $\|\mathbf{x}^{(k)}\|_1, k \ge 1$ are bounded. To this end, we define three functions:

$$L_M(\mathbf{x}) = \sum_{i=1}^{n} g_M(x_i) + 0.1 f_M(x_i), \tag{10}$$

where $g_M(x) = \min\{|x|, M\}$ and $f_M(x) = \max\{|x|, M\}$ for any $x \in (-\infty, \infty)$. Note that for each $x \in (0, \infty)$,

$$L_M(x) = g_M(x) + 0.1 f_M(x)$$

is concave. It can be seen as in Fig. 4.

It is easy to see that $L_M(x) \le L_N(x)$ if $M \le N$. A crucial observation is the subgradients of L_M, g_M and f_M are connected in the following way:

$$\partial L_M(x) = \partial g_M(x) + 0.1 \partial f_M(x) = I_M(x) + 0.1 J_M(x) \tag{11}$$

for each $x \in (-\infty, \infty)$. Also, $L_M(\mathbf{x}) = (\partial L_M(\mathbf{x}))^\top |\mathbf{x}|$. The steps inside lai2012.m are

$$\mathbf{x}^{(k)} := \min_{\mathbf{x}}\{\partial L_{M_{k-1}}(\mathbf{x}^{k-1})^\top |\mathbf{x}|, \quad A\mathbf{x} = y\}. \tag{12}$$

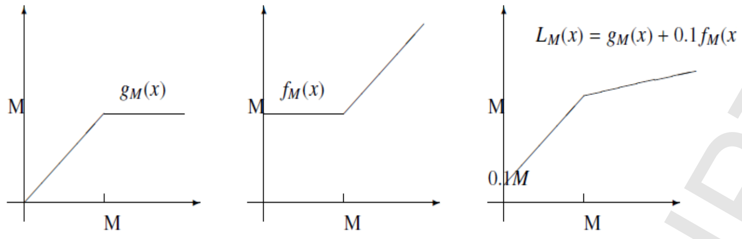


Figure 4: Functions g_M, f_M and L_M

where $|\mathbf{x}| = (|x_1|, |x_2|, \cdots, |x_n|)^\top$ for any $\mathbf{x} = (x_1, x_2, \cdots, x_n)^\top$.

We now claim that

$$L_{M_k}(\mathbf{x}^{(k+1)}) \le L_{M_{k-1}}(\mathbf{x}^{(k)}) \tag{13}$$

for all $k \ge 1$. Indeed, due to the concavity of L_M and (12), we have

$$\begin{aligned} L_{M_k}(\mathbf{x}^{(k+1)}) &\le L_{M_k}(\mathbf{x}^{(k)}) + \partial L_{M_k}(\mathbf{x}^{(k)})^\top(|\mathbf{x}^{(k+1)}| - |\mathbf{x}^{(k)}|) \\ &= L_{M_k}(\mathbf{x}^{(k)}) + \min_{\mathbf{x}} \partial L_{M_k}(\mathbf{x}^{(k)})^\top(|\mathbf{x}| - |\mathbf{x}^{(k)}|) \\ &\le L_{M_k}(\mathbf{x}^{(k)}) \le L_{M_{k-1}}(\mathbf{x}^{(k)}) \end{aligned}$$

since $M_k \le M_{k-1}$. It therefore follows

Lemma 1 *Suppose that $\|\mathbf{x}^{(2)}\|_1$ is bounded. Then there exists a convergent subsequence from $\mathbf{x}^{(k)}, k \ge 1$ and a limit $\mathbf{x}^*$ such that $\mathbf{x}^{(k_j)} \to \mathbf{x}^*$ as $j \to \infty$.*

Proof. It has $g_{M_{k-1}}(\mathbf{x}^{(k)}) + f_{M_{k-1}}(\mathbf{x}^{(k)}) = \|\mathbf{x}^{(k)}\|_1 + M_{k-1} \ge \|\mathbf{x}^{(k)}\|_1$. By using (13), we have

$$\begin{aligned} 0.1\|\mathbf{x}^{(k)}\|_1 &\le 0.1(g_{M_{k-1}}(\mathbf{x}^{(k)}) + f_{M_{k-1}}(\mathbf{x}^{(k)})) \le L_{M_{k-1}}(\mathbf{x}^{(k)}) \\ &\le \cdots \le L_{M_1}(\mathbf{x}^{(2)}) \le \|\mathbf{x}^{(2)}\|_1 \end{aligned}$$

for each $k \ge 1$. It follows that $\mathbf{x}^{(k)}, k \ge 1$ are bounded and hence, there exists a convergent subsequence from $\mathbf{x}^{(k)}, k \ge 1$ and a limit $\mathbf{x}^*$ such that $\mathbf{x}^{(k_j)} \to \mathbf{x}^*$ for $j \to \infty$. □

Lemma 2 *Let $\widehat{\mathbf{x}}$ be the sparsest vector which satisfies $A\mathbf{x} = y$. Then the limit $\mathbf{x}^*$ of any subsequence of $\mathbf{x}^{(k)}$ satisfies*

$$\|\mathbf{x}^*\|_1 \le \|\widehat{\mathbf{x}}\|_1. \tag{14}$$

Furthermore, if $\mathbf{x}^*$ and $\mathbf{y}^*$ be two limits of the subsequences of $\mathbf{x}^{(k)}$, $\|\mathbf{x}^*\|_1 = \|\mathbf{y}^*\|_1$.

Proof. Let $\alpha = \min_{\widehat{\mathbf{x}}_i \neq 0} |\widehat{\mathbf{x}}_i| > 0$. For k large enough, we have $M_k < \alpha$ and hence, $L_{M_k}(\mathbf{x}^{(k+1)}) \le L_{M_k}(\widehat{\mathbf{x}}) = 0.1\|\widehat{\mathbf{x}}\|_1$. It follows that $0.1\|\mathbf{x}^*\|_1 \le 0.1\|\widehat{\mathbf{x}}\|_1$ since $L_{M_{k_j}}(\mathbf{x}^*) \to 0.1\|\mathbf{x}^*\|_1$. Thus, we have (14).

Similarly, we have $0.1\|\mathbf{x}^*\|_1 \le L_{M_{k_j}}(\mathbf{y}^*)$ for $j \to \infty$. That is, $0.1\|\mathbf{x}^*\|_1 \le 0.1\|\mathbf{y}^*\|_1$. This statement can be reversed. These complete the proof. □

Therefore, we have obtained the following

Theorem 1 *Suppose that the sparse solution $\widehat{\mathbf{x}}$ is solved by the standard ℓ_1 minimization:*

$$\widehat{\mathbf{x}} := \min_{\mathbf{x}\in\mathbb{R}^n}\{\|\mathbf{x}\|_1 : \quad A\mathbf{x} = y\}. \tag{15}$$

For example, the RIC δ_{2s} of Φ satisfies $\delta_{2s} < 1$ or δ_s of Φ satisfies $\delta_s < 1/3$ (see [5]). Then lai2012.m converges and the limit $\mathbf{x}^*$ is equal to $\widehat{\mathbf{x}}$.

Proof. By Lemma 2 above, the limit $\mathbf{x}^*$ of any subsequence from $\mathbf{x}^{(k)}$ obtained inside lai2012.m satisfies $\|\mathbf{x}^*\|_1 \le \|\widehat{\mathbf{x}}\|_1$ and $\Phi\mathbf{x}^* = \mathbf{b}$. It follows that $\mathbf{x}^* = \widehat{\mathbf{x}}$. Thus, lai2012.m converges. □

3. Approximation of Our Economical Solution of PDE

Let $\mathbf{x}^*$ be a sparse solution satisfying $\Phi\mathbf{x}^* = \mathbf{b}$. Let s^* be the spline function with coefficient vector $\mathbf{x}^*$. The $\Phi\mathbf{x}^* = \mathbf{b}$ implies that

$$\langle \nabla s^*, \nabla s \rangle = \langle f, s \rangle, \forall s \in S_n. \tag{16}$$

It is known that $\langle \nabla u, \nabla s \rangle = \langle f, s \rangle$ and hence, we have

$$\langle \nabla(u - s^*), \nabla s \rangle = 0, \forall s \in S_n. \tag{17}$$

Using the coercivity, we have

$$\|\nabla(u - s^*)\|^2 = \langle \nabla(u - s^*), \nabla(u - s) \rangle \le \|\nabla(u - s^*)\| \cdot \|\nabla(u - s)\|$$

for any $s \in S_n$. In particular, if we choose a quasi-interpolatory spline $s(u)$ of u, we should have

$$\|\nabla(u - s^*)\| \le \min_{s\in S_n} \|\nabla(u - s)\| \le \|\nabla(u - s(u))\| \le Ch^m$$

for a positive constant C independent of h when $u \in H^{m+1}(\Omega)$ and h is the size of the partition corresponding to the space S_n. Therefore, we have established the following

Theorem 2 *Suppose that the solution u is in Sobolev space $H^{m+1}(\Omega)$ for a real number $m \ge 1$. Let s^* be the spline solution with sparse coefficient vector $\mathbf{x}^*$. Then*

$$\|\nabla(u - s^*)\| \le Ch^m$$

for a positive constant C independent of h.

4. Numerical Simulation Results

In this section, we shall give several examples to demonstrate the efficiency of the proposed method. Denote by **DOF** $= \sum_{i=1}^n N_i$ the sum of degree of freedom of S_i, $i = 1, 2, \cdots, n$. The **sparsity** here refers to the number as well as the percentage of nonzero coefficients of the numerical solution. The convergence rate CR with respect to the norm $\|\cdot\|$ at the refinement level l is roughly defined as

$$CR = \frac{2\log(\|e_{h,l}\|/\|e_{h,l-1}\|)}{log(n_{l-1}/n_l)},$$

where n_l denotes the number of the degree of freedom and $e_{h,l}$ denotes the error $u - u_h$ at refinement level l.

First of all, we have tested the correctness of our program by finding the sparse spline approximation of a Poisson equation whose solution is a polynomial like $u = x(1-x)y(1-y)$. Our sparse solution only needs a very few coefficients (about 9) while the FEM solution requires more than 1000 nonzero coefficients when the refinement level is 5. In the same fashion, if a solution can be approximated very well by using spline functions over the $(n-1)$th refined partition and using the spline functions over the nth refined partition can not improve the accuracy any more, our proposed method will find the solution over the $(n-1)$th refined partition instead and hence, have an economic representation of the PDE solution.

Next we present a table to show the comparison of the sparsity of the coefficient vectors of the standard FEM and our sparse solution.

Example 1 *Let $u = arctan((8x-4)^2 - (8y-4)^2)$ be the solution of the Poisson equation (1) with the right-hand side f which is derived from the exact solution u. We solve it by using the standard FEM and our sparse solution method (SSM). In Table 2, we show the accuracies in L_2 norm and H^1 semi-norm at different levels of refinement of the two methods. In addition, we show the number of columns of the stiffness matrix (DOF) for the standard FEM as well as the number of columns of the rectangular stiffness matrix for our sparse solution method for various levels of refinement. The sparsity is calculated based on the absolute value of a coefficient is larger or equal to $1e-6$. Finally, we present the computational times for standard FEM and sparse solution method.*

We have also repeated the above computation for $u(x,y) = tanh(40y-80x)^2) - tanh(40x-80y)^2$, $(x,y) \in [0,1]\times[0,1]$. The numerical results are similar as shown in Table 3.

From Tables 2 and 3, we can see that our solution presentations have a smaller number of nonzero coefficients than the standard FEM solution. The higher level of refinement, the fewer nonzero coefficients. This is because of the solutions are constants over several places. The place where the solution has a constant needs a fewer nonzero coefficients than the FEM solution. In general, the place where the solution can be well approximated by the spline functions over the first few levels of refinement will have fewer nonzero coefficients than the FEM solution over the last level of the refinement.

One difficulty is that it takes much more time to find the sparse solution than the FEM solution. This is still a research problem how to speed up the computation of sparse solution.

Next we compare our method with IGA based on hierarchical B-splines (HB-IGA for short) [31]. Hierarchical B-splines, composed of B-splines with different resolution, is a nature way of refining tensor product splines adaptively. In IGA, the numerical solution is represented by hierarchical B-splines and a posterior error estimator is constructed to induce the refinement. This method is integrated into the software GeoPDEs [11]. We use this software to obtain the solution solved by HB-IGA. We are going to use three different functions to compare and will make some conclusive remarks after the following three examples.

Example 2 *The exact solution*

$$u(x,y) = -tanh(\frac{\sqrt{(x-0.5)^2+(y-0.5)^2}-r}{sr}) + 1.0,$$

with $(x,y) \in [0,1]\times[0,1]$, $r = 0.25$ and $sr = 0.03$. f is derived from (1).

The exact solution u has a sharp gradient around the circle $(x-0.5)^2 + (y-0.5)^2 = 0.25^2$, referring to Fig. 5. Thus more degree of freedom is needed to capture this feature. Fig. 6 shows the non-vanishing coefficients solved by our sparse method when $n = 5$. It can be seen our sparse method can adaptively select the basis functions to get an economical representation. Table 4 shows the result obtained by our sparse method, including the degree of freedom, sparsity, L_2-norm error and H_1-norm error.

From Table 4, these two types of solution methods (SSM and HB-IGA) are really hard to compare with. For any fixed level of refinement, HB-IGA does not produce the most accurate solution while the SSM finds a near best approximation. On the other hand, for the similar accuracy, the sparsity of SSM is not as good as the HB-IGA. However, HB-IGA needs elements from additional levels of refinement. Thus we compare the convergence rate of our method with HB-IGA and TB-IGA (IGA based tensor product B-splines) in Fig. 7. It can be seen that the convergence rate of our method is similar to that of HB-IGA, but faster than TB-IGA under uniform refinement. Also, the partition associated with HB-IGA solution is complicated as it depends on a posterior error estimate. The representation of the solution in HB-IGA format will require, not only coefficients, but also the structure of the resulting mesh with many T-joints. Evaluation can be more complicated than the SSM which simply use the de Boor evaluation algorithm. For the SSM, the coefficients with additional index component of the level of refinement are needed as the nested partitions are standard. In these senses, the SSM gives a more economic representation than that of HB-IGA.

5. Conclusion and Discuss

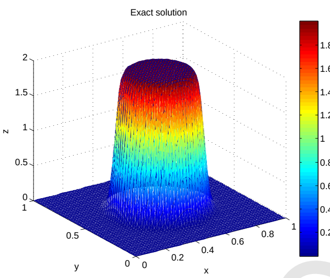
We have developed a computational algorithm to find a sparse solution to Poisson equations based on B-splines or tensor product of B-splines over uniform refinements of the underlying domain. Our the sparse solution has fewer nonzero coefficients than the standard FEM solution. We have shown that the sparse solution has the same approximation power as the standard FEM solution. As we use multi-level refined partitions which have structured basis functions and hence, the evaluation based on de Boor's algorithm will be much easier than the hierarchical T-spline basis functions. In addition, we introduce an effective sparse solution solver based on a greedy ℓ_1 strategy invented in [13] and an interior point method for the ℓ_1 minimization as used in [17]. Numerical experimental results show that this approach works well. This approach can certainly be extended to any elliptic partial differential equations by using any other spline spaces. We leave it to the interested reader to explore. On the other hand, we are not sure that the number of nonzero coefficients is the smallest possible. This is not easy to figure out as it depends on the behavior of the PDE solution and the performance of the sparse solution algorithm. Although we have demonstrate the performance of our sparse solution solver under the setting of Gaussian random matrices and uniform random matrices, the performance of the solver to the rectangular systems from a PDE is not known. Numerical results from Tables 2 and 3 show that less than 75% and 65% coefficients for the two PDE solutions,

methods	level	L_2 error	H^1 error	DOF	sparsity	time
FEM	4	0.0469378	4.75221	361	361	0.54s
SSM	4	0.0469378	4.75221	556	316 (56.8%)	0.7[illegible]s
FEM	5	0.0115948	1.94281	1225	1225	1[illegible]
SSM	5	0.0115948	1.94281	1781	1140 (64.0%)	8.80s
FEM	6	0.00137769	0.422249	4489	4489	3.[illegible]s
SSM	6	0.00137769	0.422249	6270	4136 (66.0%)	103.[illegible]1s
FEM	7	5.71136e-05	0.0351749	17161	17161	1[illegible]73s
SSM	7	5.71142e-05	0.0351749	23431	12556 (53.6%)	2[illegible]69.82s

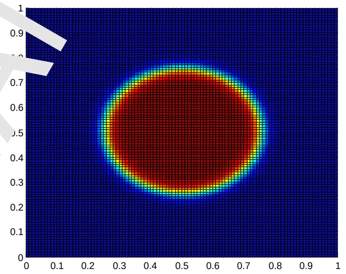
Table 2: Detailed Comparison between Standard FEM (FEM) and Sparse Solution Method (SSM)

methods	level	L_2 error	H^1 error	DOF	sparsity	time
FEM	4	0.125646	10.9247	361	361	0.43s
SSM	4	0.125646	10.9247	556	342 (6[illegible].5%)	0.71s
FEM	5	0.0471463	6.68833	1225	1225	1.16s
SSM	5	0.0471463	6.68833	1781	1[illegible]74 (65.9%)	8.88s
FEM	6	0.00993537	2.42931	4489	4[illegible]89	3.68s
SSM	6	0.00993537	2.42931	6270	3754 (59.[illegible]%)	105.05s
FEM	7	0.000674981	0.326701	17161	17[illegible]61	12.41s
SSM	7	0.000674981	0.326701	234[illegible]1	1061[illegible] (45.3%)	2179.78s

Table 3: Detailed Comparison between Standard FEM and Sparse Solution Method



(a) exact solution $u(x, y)$



(b) contour of exact solution $u(x, y)$

Figure 5: The exact solution of Example 2.

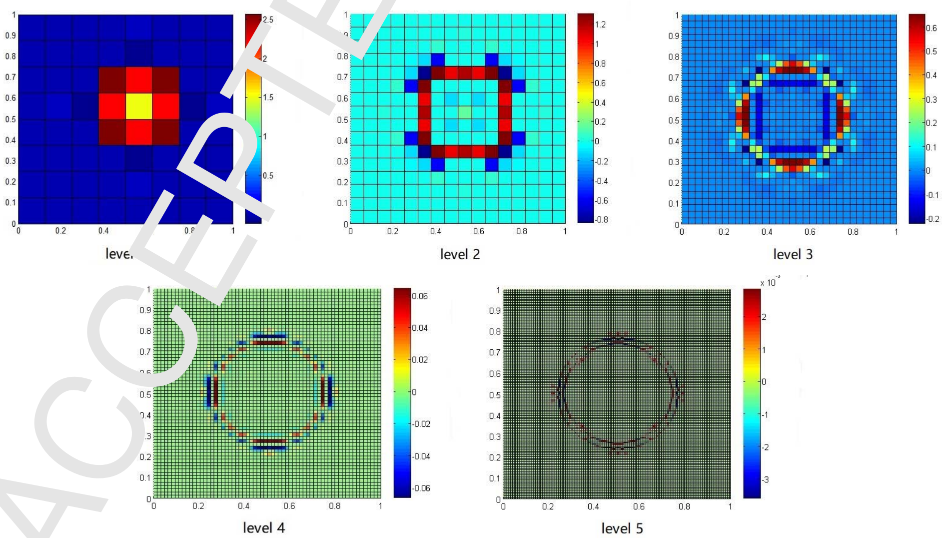


Figure 6: The coefficients solved by our method for Example 2.

methods	level	L_2 error	H^1 error	DOF	sparsity	time
SSM	4	4.0473e-2	2.5000	556	266 (47.8%)	3.21s
SSM	5	4.3717e-3	5.0432e-1	1781	924 (51.9%)	16.19s
SSM	6	1.1217e-4	3.1969e-2	6270	2592 (41.3%)	65.11s
SSM	7	3.9247e-6	2.6146e-3	23431	6860 (29.3%)	1577.71s
HB-IGA	4	4.0473e-2	2.5000		214	[illegible].48s
HB-IGA	5	4.3728e-3	5.0432e-1		506	5.20s
HB-IGA	6	4.5169e-4	4.8061e-2		1122	15.[illegible]s
HB-IGA	7	9.3030e-5	8.4712e-3		2777	[illegible]0.41s
HB-IGA	8	8.7084e-6	2.1392e-3		5653	85.6[illegible]s
HB-IGA	9	1.1308e-6	5.9735e-4		1080[illegible]	[illegible]89s

Table 4: Detailed Comparison between HB-IGA and Sparse Solution Method for Example 2.

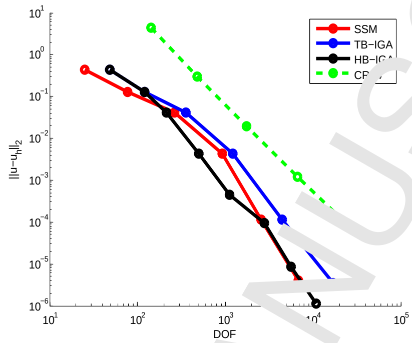
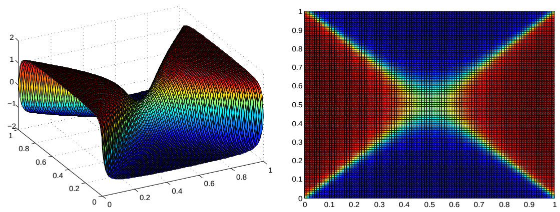


Figure 7: Convergence rate of our method SSM (red), HB-IGA (black) and TB-IGA (blue) under uniform refinement for Example 2.

respectively at refinement level 7 are needed. These indeed are big save. The major difficulty is the computational time for finding a sparse solution when the size of linear system is large. See [9] for one attempt. We leave the study how to speed it up to a future research problem.



(a) Exact Solution in 3D (b) Contour View

Figure 8: Exact solution(3D and Contour View)

We have also compared with the well-known hierarchical tensor-product B-spline functions for numerical solution of PDE. For any fixed level of refinement, our method can produce a more accurate solution than that of the HB-IGA method. However, for any fixed accuracy, the solution of HB-IGA is less sparse than our SSM. Especially when the solution does not have a lot of zeros, this phenomenon is more remarkable. For example the exact solution is chosen as $u(x, y) = arctan((8x-4)^2-(8y-4)^2)$, $(x, y) \in [0, 1] \times [0, 1]$, which is almost constants in several areas. The graph of this solution is shown in Fig. 8, where the sharp gradient locates at two diagonal line segments of the square. Table 5 shows the result obtained by our sparse method, including the degree of freedom, L_2 norm error and H_1 semi-norm. We compare the convergence rate of our method with HB-IGA and TB-IGA in Fig. 9. It can be seen that the convergence rate of our method is faster than IGA under uniform refinement, but is not so good with that of HB-IGA because the solution does not have a lot of zeros. This leads to a grand challenge problem: for a fixed accuracy, what is the most economic way to solve a PDE? HB-IGA combing with some techniques of reducing the scale of the sparse problem ([illegible]) (for example the low rank method proposed in [38]) can be a candidate to the challenge problem. We leave the problem to the interested reader.

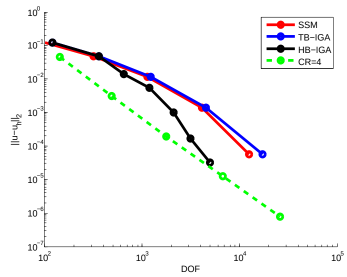


Figure 9: Convergence rate of our method (red), HB-IGA (black) and TB-IGA (blue) under uniform refinement

methods	level	L_2 error	H^1 error	DOF	sparsity	time
SSM	4	0.04693	4.7522	556	316 (56.8%)	1.79s
SSM	5	1.1594e-2	1.9428	1781	1140 (64.0%)	9.11s
SSM	6	1.3776e-3	0.42224	6270	4130 (65.9%)	105.3[illegible]s
SSM	7	5.7114e-5	3.5175e-2	23431	12556 (53.6%)	2205.[illegible]5s
HBIGA	5	0.01395	2.03324		645	[illegible].64[illegible]
HBIGA	6	5.3576e-3	0.61194		1197	[illegible]7.10[illegible]
HBIGA	7	1.6565e-4	5.4275e-2		3125	42.[illegible]s
HBIGA	8	3.2329e-5	1.4126e-2		4973	[illegible]9.98s

Table 5: Detailed Comparison between HB-IGA and Sparse Solution Method

Acknowledgement

The authors thank the reviewers for providing useful comments and suggestion. Hongmei Kang is supported by the National Natural Science Foundation of China (No. 11801393) and the Natural Science Foundation of Jiangsu Province (No. BK20180831). Ming-Jun Lai is supported by the National Science Foundation under the grant #DMS 1521537. Xin Li is supported by the NSF of China (No.61872328), NKBRPC (2011CB302400), SRF for ROCS SE, and the Youth Innovation Promotion Association CAS.

Appendix

The following matlab program was used in our computation. We make it available so that more people can use it.

```
%function x=L1min(C,b,tol,x0)
%% This function is to min $\|x\|_1$, subject to
%% $Cx=b$. $tol = 0.0001$, $x0$ is an initial
%% guess satisfying $Cx=b$.
%% It is written based on a paper "L^1 Spline
%% Methods for Scattered Data Interpolation
%% and Approximation", M. J. Lai and P. Wenston
%% Advances in Computational Mathematics,
%% vol. 21 (2004) pp. 293--315.
%[k,m]=size(C); x=x0;
%alpha=norm(b,inf);
%w=(2/(3*alpha))*x;
%it_count=0; max_it=25;
%Z=zeros(m,1);
%cvg=0;
%while ~cvg & it_count <= max_it
%D=spdiags(1-abs(w),0,m,m);
%xnew=[[(D)'*(D),C'];[C,sparse(k,k)];...
% sparse(1,m+k)]\[Z;b;0];
%xnew=xnew(1:m);
%x=xnew;
%p=D^2*x;
%alpha=max(max(p'./(1-w'),-p'./(1+w')));
%w=w+(2/(3*alpha))*p;
%err=norm(x,1)-w'*x;
%cvg=err<tol;
%it_count=it_count+1;
%end;
```

References

[1] G. Awanou, M. J. Lai, P. Wenston, The multivariate spline method for numerical solution of partial differential equations, in *Wavelets and Splines*, Nashboro Press, Brentwood, 24–74, 2006.

[2] A. Beck and M. Teboulle, A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems, SIAM J. Imaging Sciences, 2:183–202, 2009.

[3] T. Blumensath and M. E. Davies, Iterative hard thresholding for compressed sensing, Appl. Comput. Harmon. Anal., 27:265–274, 2009.

[4] J.-F. Cai, S. Osher and Z. Shen, Convergence of the Linearized Bregman Iteration for ℓ_1-Norm Minimization, Math. Comp., 78(268):2127–2136, 2009.

[5] T. T. Cai and Zhang, A., Sharp RIP bound for sparse signal and low-rank matrix recovery, Applied And Computational Harmonic Analysis, 35:74–93, 2013.

[6] E. J. Candès, M. B. Wakin, and S. Boyd, Enhancing sparsity by reweighted ℓ_1 minimization, Journal of Fourier Analysis and Applications, 14:877–905, 2008.

[7] E. J. Candès and T. Tao, Decoding by linear programming, IEEE Trans. Inform. Theory, 51(12):4203–4215, 2005.

[8] Candes, E. J., J. K. Romberg, and T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Comm. Pure Appl. Math., 59:1207–1223, 2006.

[9] Deng, W., Lai, M. J., Peng, Z. M. and Yin, W. T., Parallel Multi-Block ADMM with $o(1/k)$ Convergence, Journal of Scientific Computing, 71:712–736, 2017.

[10] S. Foucart, Hard thresholding pursuit: an algorithm for compressive sensing, SIAM Journal on Numerical Analysis, 49(6):2543–2563, 2011.

[11] C. de Falco, A. Reali, R. Vázquez, GeoPDEs: A research tool for Isogeometric Analysis of PDEs, Advances in Engineering Software, 42(12), 2011.

[12] P. Gong, C. Zhang, Z. Lu, J. Huang, J. Ye, A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems, The 30th International Conference on Machine Learning (ICML), 37–45, 2013.

[13] K. Kozlov and A. Petukhov, Sparse solution of underdetermined linear systems, in Handbook of Geomathematics, W. Freeden, M.Z. Nashed, T. Sonar (Eds.), Springer, 1243–1259, 2010.

[14] M. -J. Lai and L. L. Schumaker, Spline Functions over Triangulations, Cambridge University Press, 2007.

[15] M. -J. Lai and A. Varghese, On Convergence of the Alternating Projection Method for Matrix Completion and Sparse Recovery Problems , submitted , 2018.

[16] M. -J. Lai and J. Wang, An unconstrained ℓ_q minimization for sparse solution of under determined linear systems, SIAM J. Optimization, 21:82–101, 2011.

[17] M. -J. Lai, and Wenston, P., L^1 Spline Methods for Scattered Data Interpolation and Approximation, Advances in Computational Mathematics, 21:293–315, 2004.

[18] M. -J. Lai, Xu, Y. Y. and Yin, W. T., Improved Iteratively Reweighted Least Squares for Unconstrained Smoothed ℓ_p Minimization, SIAM Journal on Numerical Analysis, 51:927–957, 2013.

[19] M. -J. Lai and Yin, W. T., Augmented ℓ_1 and Nuclear-Norm Models with a Globally Linearly Convergent Algorithm, SIAM Journal Imaging Sciences, 6:1059–1091, 2013.

[20] L. L. Schumaker, Spline Functions: Computational Methods, SIAM Publication, 2015.

[21] J. A. Tropp, Greed is good: algorithmic results for sparse approximation, IEEE Trans. Inform. Theory, 50(10):2231–2242, 2004.

[22] J. Wang, P. Wonka, and J. Ye, Lasso screening rules via dual polytope projection, J. Machine Learning Research, 16(1):1063–1101, 2015.

[23] Michael R. Döfel, Bert Jüttler, Bernd Simeon, Adaptive isogeometric analysis by local h-refinement with T-splines, Comput. Methods Appl. Mech.

Eng., 199(5–8):264–275, 2010.
[24] I. Babuska, W.C. Rheinboldt, Error estimates for adaptive finite element computations, SIAM J. Numer. Anal., 15:736–754, 1987.
[25] M. Ainswort, J. Oden, A Posteriori Error Estimation in Finite Element Analysis, Wiley-Interscience, New York, 2002.
[26] R. Verfürth, A Review of a Posteriori Error Estimation and Adaptive Mesh-refinement Techniques, Wiley-Teubner, Chichester, UK, 1996.
[27] Y. Bazilevs, L. Beirao da Veiga, J.A. Cottrell, T.J.R. Hughes, G. Sangalli, Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes, Math. Models Methods Appl. Sci., 16(7):1031–1090, 2006.
[28] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Computer methods in applied mechanics and engineering, 194(39-41): 4135–4195, 2005.
[29] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, Isogeometric analysis: toward integration of CAD and FEA, John Wiley & Sons, 2009.
[30] Xin Li, M. A. Scott, Analysis-suitable T-splines: Characterization, Refinablility and Approximation, Mathematical Models and Methods in Applied Sciences, 24, 2014.
[31] A.-V. Vuong, C. Giannelli, B. Jüttler and B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis, Comput. Methods Appl. Mech. Engrg., 200:3554–3567, 2001.
[32] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Huges, S. Lipton, M.A. Scott, T.W. Sederberg, Isogeometric analysis using T-splines, Comput. Methods Appl. Mech. Engrg., 199: 229–263, 2010.
[33] Wang P., Xu J., Deng J., Chen F., Adaptive isogeometrican alysis using rational PHT-splines, Computer-Aided Design, 43:1438–1448, 2011.
[34] Johannessen K A, Kvamsdal T, Dokken T. Isogeometric analysis using L-R B-splines, Computer Methods in Applied Mechanics and Engineering, 269:471–514, 2014.
[35] Evans E.J., Scott M.A., Li X., Hierarchical analysis-suitable T-splines: Formulation, Bézier extraction, and application as an adaptive basis for isogeometric analysis, Comput. Methods Appl. Mech. Engrg., 284:1–20, 2015.
[36] Cottrell J.A., Hughes T.J.R., Reali A., Studies of refinement and continuity in isogeometric analysis, Comput. Methods Appl. Mech. Engrg., 196(41-44):4160–4183, 2007.
[37] David Donoho and Andrea Montanari, High dimensional robust M-estimation: asymptotic variance via approximate message passing. Probab. Theory Related Fields, 166(3-4):935–969, 2016.
[38] Angelos Mantzaflaris, Bert Jüttler, Boris N.Khoromskij, Ulrich Lnger, Low rank tensor methods in Galerkin-based isogeoemtric analysis, Computer Methods in Applied Mechnics and Engineering, 316:1062–1085, 2017.

Highlights

- We find an economical representation of the spline approximation of the PDE by using a compressive sensing approach.
- The number of nonzero coefficients of the proposed economical representation is less than the number of the standard spline representation over the last refined partition, while the error of the spline approximation with an economical representation is the same to the standard FEM solution.
- The sparsity of a PDE solution may not be very small in general. We present a new way to solve a sparse solution of an underdetermined system in order to adapt to computing an economic representation of PDE solution.

Conflict of interest

The authors declared that they have no conflicts of interest to this work.