

# Mechanical field reconstruction from limited observations

\*Note: Sub-titles are not captured in Xplore and should not be used

Bahador Bahamani  
CEEM Department  
Columbia University  
NY, USA  
bb2969@columbia.edu

**Abstract**—In this project, we aim to reconstruct temperature field (as a case study) from a few noisy measurements which is similar to the real experimental conditions. To this end, we propose a dictionary learning approach that tries to find a sparse combination of dictionary elements. We find such sparse weights using basis pursuit and LASSO formalism and compare these optimization formulations. In this work, the dictionary is constructed with a priori reliable solutions, e.g., from numerical solvers, without any transformation.

**Index Terms**—component, formatting, style, styling, insert

## I. INTRODUCTION

In mechanical engineering, we have inevitable limitations regarding the number and location of sensors for an experimental task. These limitation could be related to expensiveness of sensors or manufacturing restrictions about accessibility of different locations on the surface of samples or inside the samples. On the other hand, we are interested to have as much as possible more information about the entire body of sample to conduct analysis more rigorously. Another concern is that experimental observations come with unavoidable level of noise. Thus, it would be very helpful if we could find a robust approach that can reconstruct a full field from limited and noisy measurements. Due to amazing successes in the field of sparse modeling, we are interested to try some of those ideas in the field of mechanics and specially for heat transfer experiments of heterogeneous materials, in this project. Although we have tried heat transfer problem, the approach could be used for other applications such as stress analysis.

The idea of using sparse modeling to reconstruct a mechanical field is first introduced by [1] for fluid dynamics applications. The main idea of their approach comes from the seminal work of [2] where they introduce a robust approach for image recognition using sparse representation and dictionary learning. In another line of research, [3] introduces a neural network based method for field reconstruction from limited measurements, again for fluid dynamics applications. Our method used in this manuscript is similar to [1], [2] but for a different domain application.

In this project, we assume that governing equations of the heat transfer experiment are completely known to use and the actual experimental data should satisfy the conservation of energy if the experiment is conducted perfectly fine without noise. Therefore, we hypothesize that if sufficiently many admissible solutions (e.g., from a numerical simulator) are collected with the same boundary condition as the actual experiment, then the experimental observation can be represented as a linear combination of those admissible solutions.

## II. TECHNICAL APPROACH

We follow the same line of idea presented in [1], [2]. For a state vector  $y \in \mathbb{R}^n$  which represents a mechanical field, for example temperature, over a set of grid points, we are interested to find and its approximation called  $y^h \in \mathbb{R}^n$  by knowing a subset of its components  $y_{\text{meas}} = My$  where  $y_{\text{meas}} \in \mathbb{R}^m$  and  $m \ll n$ . We assume that  $y$  admits a linear combination of library elements  $\{\psi_j\} \in \mathbb{R}^n$  where  $j = 1, 2, \dots, r$ , hence we have:

$$y^h \approx \Psi c = \begin{pmatrix} | & | & \cdots & | \\ \psi_1 & \psi_2 & \cdots & \psi_r \\ | & | & \cdots & | \end{pmatrix} c, \quad (1)$$

where  $c \in \mathbb{R}^r$  is the coefficient vector that indicates the contribution of each mode presented in the library  $\Psi \in \mathbb{R}^{n \times r}$ . Within this set-up, the reconstruction problem is equivalent to finding an appropriate coefficient vector that can satisfy the following condition:

$$y_{\text{meas}}^h \approx M \Psi \hat{c}, \quad (2)$$

where  $\hat{c}$  is an unknown coefficient vector that must be found such that the right hand side in the above equation be as much as close to the left hand side according to an appropriate metric (e.g., L2-norm). As mentioned, in real application the number of measurements are much less than the total degrees of freedom in the underlying mechanical system, and hence the above equation is underdetermined ( $m < r$ ). It is well-accepted that most of the physical problems admit a form of sparse representation (at least from empirical perspective) in the ambient space or an appropriate transformed space such

as Fourier or wavelet space. Therefore, it is natural to assume the coefficient vector has sparsity structure to regularize the above mentioned problem.

To promote sparsity in the coefficient vector, we can reformulate the problem with different optimization formulations. Here, we consider two approaches: 1-basis pursuit regression and 2-LASSO regression. In the context of basis pursuit, we look for the sparsest coefficient  $\hat{c}$  in the L1-norm sense such that:

$$\hat{c} = \underset{c}{\operatorname{argmin}} \|c\|_1 \text{ s.t. } y_{\text{meas}} = M\Psi\hat{c}, \quad (3)$$

when there is no noise in the data. For noisy measurements we have:

$$\hat{c} = \underset{c}{\operatorname{argmin}} \|c\|_1 \text{ s.t. } \|y_{\text{meas}} - M\Psi\hat{c}\|_2 < \epsilon, \quad (4)$$

where  $\epsilon$  is an hyperparametr that controls the level of deviation of the reconstructed measurements from noisy measurements. In the LASSO regression we solve the following optimization problem for regardless having noisy data or not:

$$\hat{c} = \underset{c}{\operatorname{argmin}} \|y_{\text{meas}} - M\Psi\hat{c}\|_2 + \lambda\|c\|_1, \quad (5)$$

where  $\lambda$  is a penalty parameter that controls the level of sparsity in the coefficient vector. In this project, we formulate these convex optimization problem using `cvxpy` package [4].

### III. DATA GENERATION AND PROCESS

As mentioned earlier, our approach relies on a dictionary consist of sufficiently many admissible solutions that satisfy the underlying governing equations of heat transfer with boundary and initial conditions similar to the actual. Therefore, we need to collect such solutions as elements of our library. We assume heat conduction is the only heat transfer mechanism in the actual experiment, hence the conservation of energy leads to the following parabolic partial differential equation (PDE):

$$\rho c_T \frac{\partial T}{\partial t} - \nabla \cdot (-\kappa(x, t) \nabla T(x, t)) = 0, \quad (6)$$

where  $\rho$ ,  $c_T$ ,  $T(x, t)$ ,  $\kappa(x, t)$ ,  $x$ , and  $t$  are density, heat capacity, temperature field, heat conductivity, spatial coordinate and time, respectively. In this project, we use euler method to discretize the above PDE in time and utilize Galerkin approximation to discretize in space within the context of finite element method (FEM). We have implemented our simulator in an open-source FEM library called `fenics` [5].

Typically our mesh for use in FEM has a an irregular topology (see Fig. 1) and does not look like an square shaped grid which is common for sensor placement in an experimental test or images in computer vision studies. To be as much as close to the experiential conditions, we perform some basic image processing operations including (1) binarizing temperature field as a single color in gray scale, (2) cropping image to exclude the boundary effects, and (3) down sampling the image to reduce the computational cost for later optimization tasks. These basic image processing operations are performed with an open-source packaeg called `scikit-image` [6]. Figure 2 shows two snapshots before and after this process.

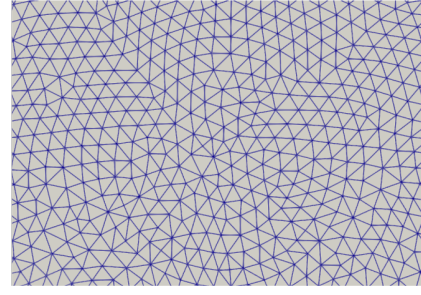


Fig. 1. a zoomed window of the used FEM mesh.

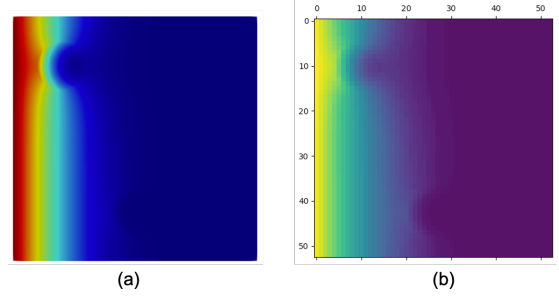


Fig. 2. (a) high resolution image from nodal solution of FEM shown in `paraview` software [7] and (b) binarized an down sampled image ( $55 \times 55$  pixels) for use in the optimization task. Notice that these images are not corresponding to each other; we just show two cases as illustration.

### IV. EXPERIMENTS

In this section, we first introduce a toy example which doesn't require any numerical simulator to construct the library. This allows interested reader tries the main idea with a simpler problem at hand, and also it is a good practice for verification exercise. In the second example, we solve our target heat transfer problem.

#### A. toy problem: artificial data with known analytical expression

We generate a 1000 snapshots of a target field at different times  $t \in [0, 0.76]$  with the following expression:

$$y = c_1 \exp\left(-\frac{(x - x_1)^2}{2\sigma_1^2}\right) \sin(2\pi f_1 t) + c_2 \exp\left(-\frac{(x - x_2)^2}{2\sigma_2^2}\right) \sin(2\pi f_2 t). \quad (7)$$

The parameters we used for data generation are  $c_1 = 1$ ,  $x_1 = 0.5$ ,  $\sigma_1 = 0.6$ ,  $f_1 = 1.3$ ,  $c_2 = 1.2$ ,  $x_2 = -0.5$ ,  $\sigma_2 = 0.3$ , and  $f_2 = 4.1$ . We regularly sample spatial coordinate  $x \in [-2, 2]$  with 400 points. 500 snapshots are randomly selected to built our dictionary and, measurements are gathered from 8 random points among the 400 possible places on the  $x$  axis. Therefore, in this problem we have  $n = 400$ ,  $r = 300$ , and  $m = 8$ . Note that we normalize our data along the temporal axis to have zero mean and unit standard deviation. Figure 3 shows the result of reconstructed field and coefficient vector for the basis pursuit problem when there is no noise in the measurements. Figure 4 plot the same quantities but for the LASSO problem. We observe that both formulation can accurately recover the

full field from just 8 random measurements, however basis pursuit approach performs three orders of magnitude better than LASSO approach in terms of normalized error defined below:

$$\text{err}(y_{\text{true}}, y_{\text{recov}}) = \frac{\|y_{\text{true}} - y_{\text{recov}}\|_2}{\|y_{\text{true}}\|_2}. \quad (8)$$

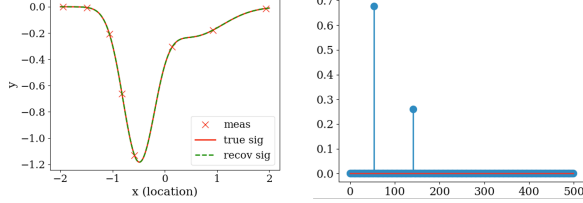


Fig. 3. Results of basis pursuit problem. (left) recovered full field solution shown by dashed green line from a few measurements shown by cross red markers. Red curve is the ground truth, and normalized l2 error between recovered and exact solutions is  $1.06e-6$ . (right) founded coefficient vector.

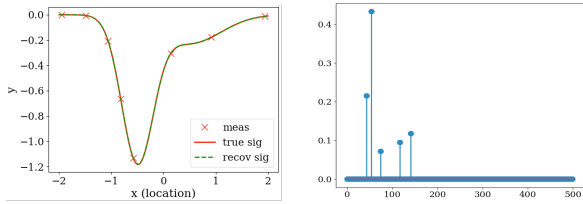


Fig. 4. Results of LASSO problem when  $\lambda = 0.01$ . (left) recovered full field solution shown by dashed green line from a few measurements shown by cross red markers. Red curve is the ground truth, and normalized l2 error between recovered and exact solutions is 0.002. (right) founded coefficient vector. This suggested  $\lambda$  is found by some trial-errors to best fit the true available solution.

To testify the robustness of our proposed approaches, we add artificial noise to the measurements. As shown in Figs. 5 and 6 both approaches can recover a smooth field close to the ground truth from such a noisy sparse measurements. Although the accuracy is significantly reduced compared to noiseless recovery, still the recovery performance is acceptable. Note that in the noisy data condition, the accuracy of LASSO solution is slightly higher than basis pursuit solution. As expected, we observe the coefficient vector is sparser for basis pursuit solutions regardless of the amount of presented noise.

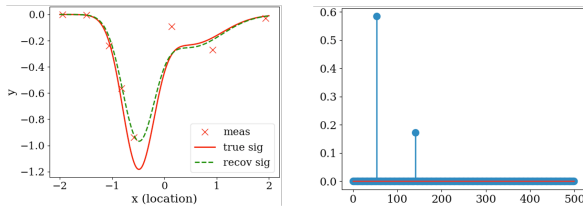


Fig. 5. Results of basis pursuit problem when random Gaussian noise (zero mean and standard deviation 0.3) is added to the measurements. (left) recovered full field solution shown by dashed green line from a few measurements shown by cross red markers. Red curve is the ground truth, and normalized l2 error between recovered and exact solutions is 0.17. (right) founded coefficient vector.

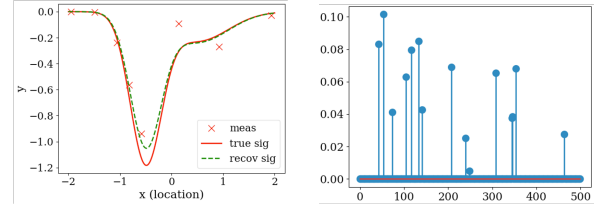


Fig. 6. Results of LASSO problem when  $\lambda = 0.01$ . (left) recovered full field solution shown by dashed green line from a few measurements shown by cross red markers. Red curve is the ground truth, and normalized l2 error between recovered and exact solutions is 0.1. (right) founded coefficient vector. This suggested  $\lambda$  is found by some trial-errors to best fit the true available solution.

## B. Two dimensional heat transfer in a heterogeneous domain

In this case, we generate admissible solutions by solving the heat transfer problem for the heterogeneous material shown in 7 using FEM. We solve the problem, for 300 time steps until the solution converges to the steady state. The temperature field in the domain is temporally normalized to have a zero mean and unit standard deviation at each pixel among all time steps. Then we choose 240 snapshots of the transient solution as elements of our dictionary; these elements are temperature solution at the ambient space and without any transformation such as Fourier transform.

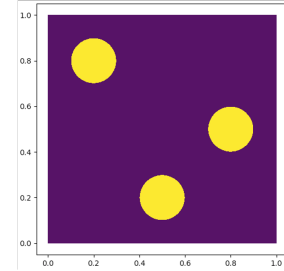


Fig. 7. experimental sample consist of circular fibers (shown by yellow color) and surrounding matrix (shown by purple domain). In the experiment, temperature of the left boundary is set to one and right boundary has zero temperature.

To see the performance of the proposed method, we select a solution field at a random time step (see Fig. 8) which is not appeared in the dictionary. We randomly choose 2 percent of the pixels as sensors locations to observe the temperature values. These sensor locations are shown by white cross markers in Fig 8. For sparse recovery problem we only use the temperature values at these points and we do not have any access to other locations.

## V. CONCLUSION

### REFERENCES

- [1] Callahan, Jared L., Kazuki Maeda, and Steven L. Brunton. "Robust flow reconstruction from limited measurements via sparse representation." *Physical Review Fluids* 4.10 (2019): 103907.
- [2] Wright, John, et al. "Robust face recognition via sparse representation." *IEEE transactions on pattern analysis and machine intelligence* 31.2 (2008): 210-227.
- [3] Erichson, N. Benjamin, et al. "Shallow neural networks for fluid flow reconstruction with limited sensors." *Proceedings of the Royal Society A* 476.2238 (2020): 20200097.

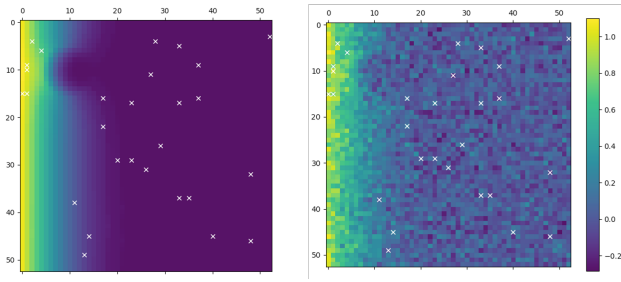


Fig. 8. White cross markers are measurement locations which are randomly cover only 2 percents of pixels. In the sparse recovery task we only have access to the temperature of these locations, i.e., sensors. In (left) observations do not have any noise, but in (right) gaussian noise with zero mean an 0.1 standard deviation.

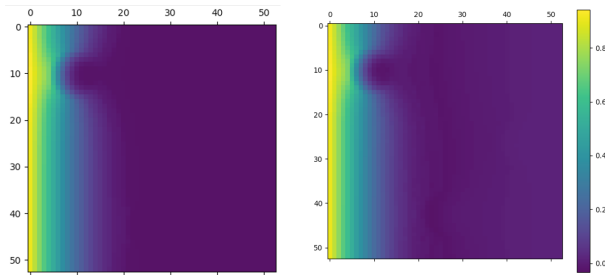


Fig. 9. (recovered solutions from noiseless (left image) and noisy (right image) measurements with LASSO algorithm. We set  $\lambda$  equal to 0.01.

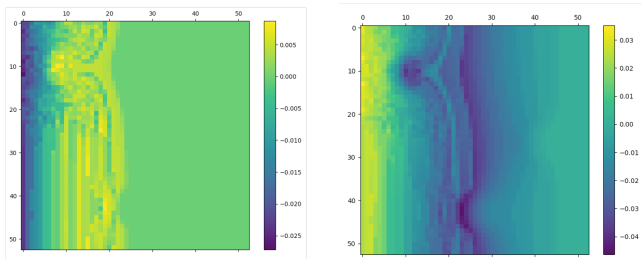


Fig. 10. error between recovered and ground truth fields when measurements are noiseless (left image) and noisy (right image) with LASSO algorithm.

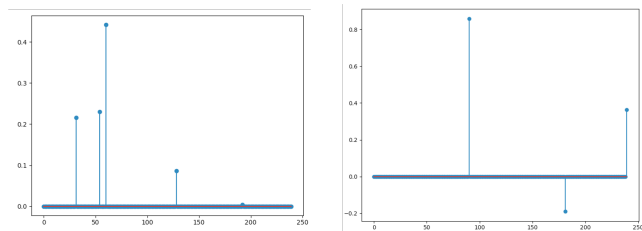


Fig. 11. founded coefficient vector by the LASSO algorithm when measurements are noiseless (left image) and noisy (right image).

- [4] Diamond, Steven, and Stephen Boyd. "CVXPY: A Python-embedded modeling language for convex optimization." The Journal of Machine Learning Research 17.1 (2016): 2909-2913.
- [5] Alnaes, Martin, et al. "The FEniCS project version 1.5." Archive of Numerical Software 3.100 (2015).
- [6] Van der Walt, Stefan, et al. "scikit-image: image processing in Python." PeerJ 2 (2014): e453.
- [7] Ahrens, James, Geveci, Berk, Law, Charles, ParaView: An End-User