

Spectral clustering for compressing physical simulations

Christopher Brissette^{*,†}

Andy Huang^{*}, Nathaniel Trask^{*}, George M. Slota^{*,*}

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. This research was funded through the Laboratory Directed Research and Development (LDRD) program at Sandia.

^{*}Rensselaer Polytechnic Institute

^{*} Sandia National Labs

[†]*brissc@rpi.edu*

February 25, 2022

A Few Observations:

- large physics meshes are used to simulate semiconductors
- simulating finite elements or finite differences is very costly
- real world circuits have many semiconductors in them

⇒ **large arrays of semiconductors are incredibly expensive to simulate**

Question: **Can we automate the generation of reduced order models for circuit simulation?**

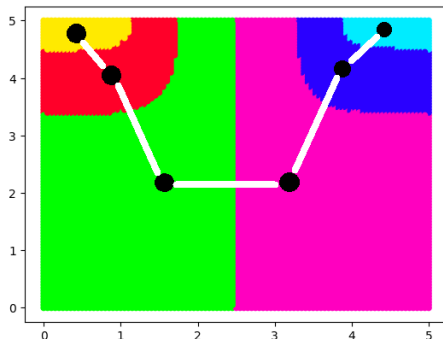
- This is a big question, with a lot of parts (machine learning / data analysis / numerical PDE).
- The particular question we will work towards today is only a single step in that direction, **mesh coarsening**.

Topic Description

The general idea:

- 1 Take a simulation that has already been run for a semiconductor device.
- 2 Partition similarly behaved vertices into super-nodes.
- 3 Approximate the true mesh behavior with this smaller graph.

The last step here is where ML comes in since these edges are likely to have non-linear weights. We focus today on how to approach step 2.



We can reframe this in terms of **functional approximation**.

- We have a discrete function $u(\vec{x})$ in \mathbb{R}^n on M points.
- Given $k < M$ points we want to approximate this function in the best possible way.

Main Question:

Given a function $u(\vec{x}) \in \mathbb{R}^n$ how do we best approximate it with k piece-wise constant partitions?

We consider an underlying mesh defined by a graph structure $\mathbf{G} = (V, E)$.

- Each node $v_i \in V$ is associated with a unique position $p_i \in \mathbb{R}^d$.
- Cliques then define simplices (think of them as ordered tuples).
- These simplices come with an orientation as well (required to define a differential form).

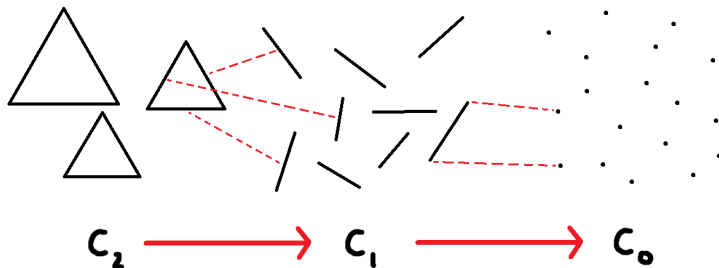
Discrete Exterior Calculus

A **k-chain** c_k is defined as a linear combination of $(k+1)$ -cliques. Denote the set of k -chains by C_k .

We introduce the **boundary operator**, $\partial_k : C_{k+1} \rightarrow C_k$

$$\partial_k[v_1, \dots, v_k] = \sum_{i=1}^k (-1)^{i-1} [v_1, \dots, n_{i-1}, \hat{n}_i, n_{i+1}, \dots, n_k]$$

Here $\hat{}$ denotes an omitted entry.



We additionally consider the **dual cochains**, C^k : the linear functionals acting on C_k (think row vectors versus column vectors).

We also introduce the **coboundary operator** $\delta_k : C^k \rightarrow C^{k+1}$.

$$C_0 \xleftarrow{\partial_0} C_1 \xleftarrow{\partial_1} \cdots \xleftarrow{\partial_{d-1}} C_d$$

$$C^0 \xrightarrow{\delta_0} C^1 \xrightarrow{\delta_1} \cdots \xrightarrow{\delta_{d-1}} C^d$$

Using these cochain complexes one can derive a general version of the graph Laplacian that works for any DEC structure (The Hodge Laplacian).

$$L_k = \delta_{k-1}\delta_{k-1}^* + \delta_k^*\delta_k \quad (1)$$

In fact the standard graph Laplacian can simply be written as follows.

$$L_0 = \delta_0^*\delta_0 \quad (2)$$

For more about this we direct everyone to [1, 2]

Eigenvectors of the Laplacian

Luckily, this ends up being the same as the familiar combinatorial Laplacian.

$$L_0 = \mathcal{L} = D - A \quad (3)$$

This is what we will be using for the rest of this talk. But! Remember that we have a general version of this for discrete geometry on k-forms!

Eigenvectors of the Laplacian

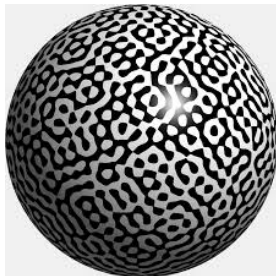
Remember our goal: **Given a function $u(\vec{x}) \in \mathbb{R}^n$ approximate it with k piece-wise constant partitions.**

Idea: Use the nodal sets of the Laplacian eigenvectors to define regions for approximation.

Eigenvectors of the Laplacian

We want to leverage some intuition provided to us from the continuous case of the Laplacian on Riemannian manifolds. Mainly that the eigenfunctions on the n -sphere S^n tend towards having evenly spaced nodal sets as we increase our eigenfunctions [4]. Some of this behavior is proven to also exist in the discrete case [3].

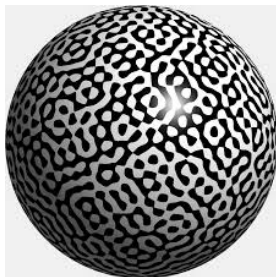
$$N_\lambda = \{x \in M \mid \phi_\lambda(x) = 0\} \quad (4)$$



Eigenvectors of the Laplacian

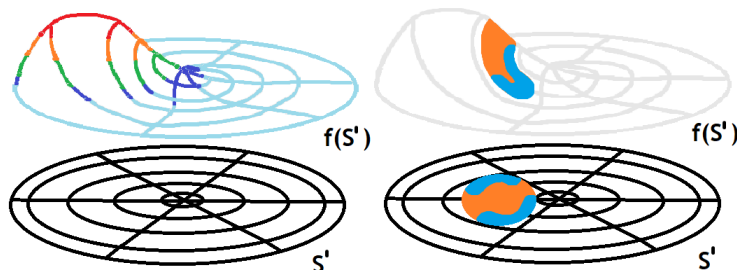
The **nodal domains** are then the connected components of the complement of N_λ . Use the nodal domains for our piece-wise approximation.

$$S \cap \bar{N}_\lambda$$



Eigenvectors of the Laplacian

- Deform S^n to $f(S^n)$. These nodal sets should then deform and provide more fidelity in regions with a higher gradient!
- Perhaps the nodal sets of the k^{th} eigenvector will yield the approximation properties we want!



Eigenvectors of the Laplacian

- The graph Laplacian \mathcal{L} does not care about values of f directly.
- Instead we need to reflect the behavior in f that we want to see through edge weights.
- Intuitively higher weights correspond to closer nodes.

Eigenvectors of the Laplacian

One natural choice is to take the weights to be the inverse difference of nodes.

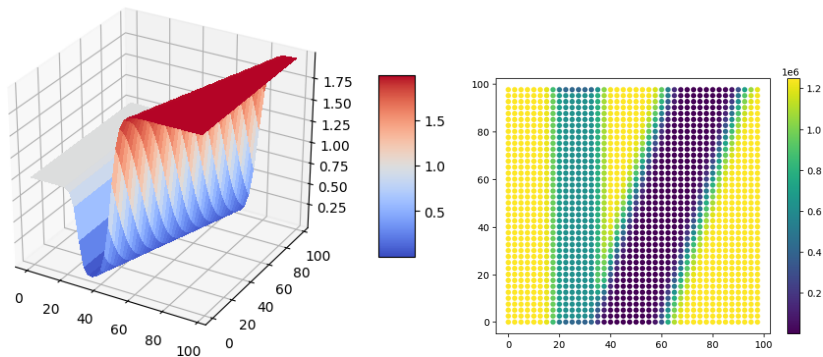
$$w_{ij} = \frac{1}{|f(v_i) - f(v_j)| + \epsilon} \quad (5)$$

We instead take weights to be the inverse of the squared directional derivative.

$$w_{ij} = \frac{\|[x_j, y_j] - [x_i, y_i]\|^2}{\|f(v_i) - f(v_j)\|^2 + \epsilon} \quad (6)$$

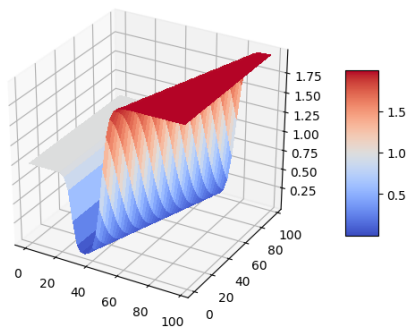
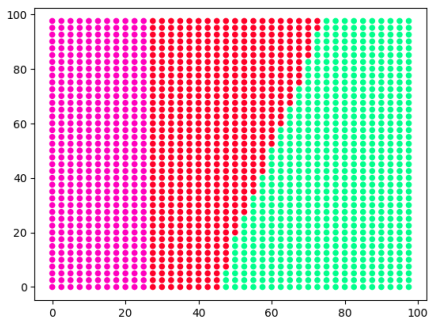
Eigenvectors of the Laplacian

Let's examine a test function made out of a sum of two logistic functions!



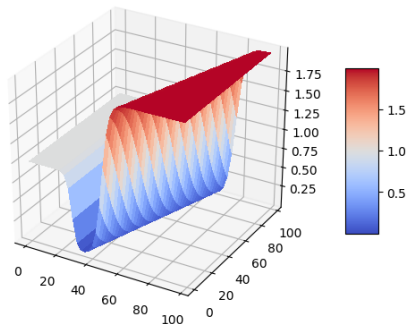
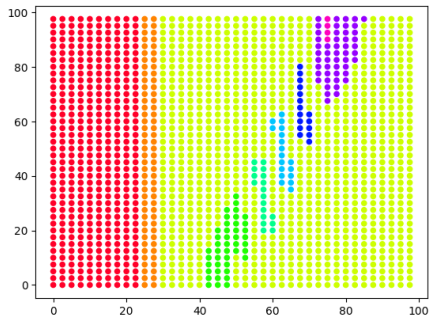
Eigenvectors of the Laplacian

The nodal sets of the third eigenvector look pretty good! They are exactly where we would want them!



Eigenvectors of the Laplacian

The 16th eigenvector is not as nice. We have higher granularity in some regions and it is well placed. But the approximation in the surrounding yellow region has been ruined!



Eigenvectors of the Laplacian

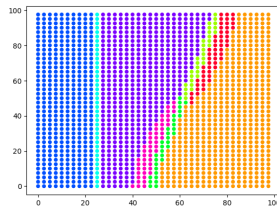
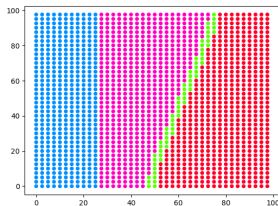
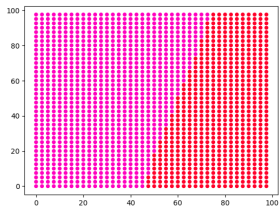
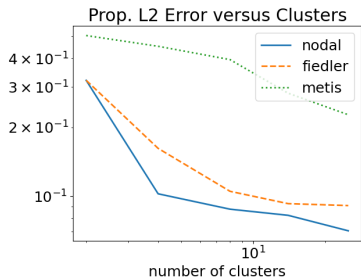
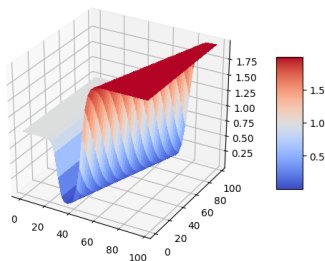
Two big problems with simply using the nodal sets of the k^{th} eigenvector:

- ① We do not have a clear lower bound on the number of nodal sets.
- ② More nodal sets doesn't always mean better piece-wise accuracy.

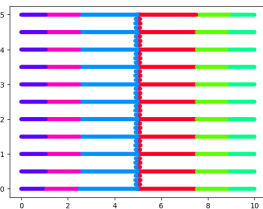
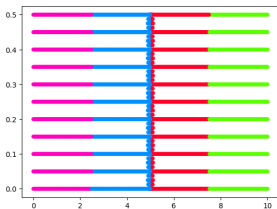
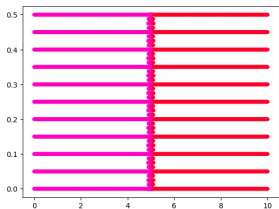
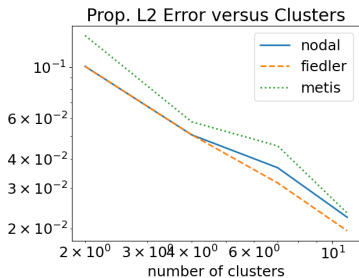
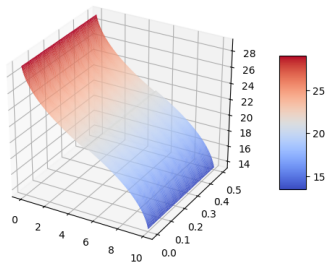
Idea: Instead, we consider the intersection of nodal sets!

- ① Allows us to leverage the spacing of nodal sets
- ② Guarantees monotonic decrease in error

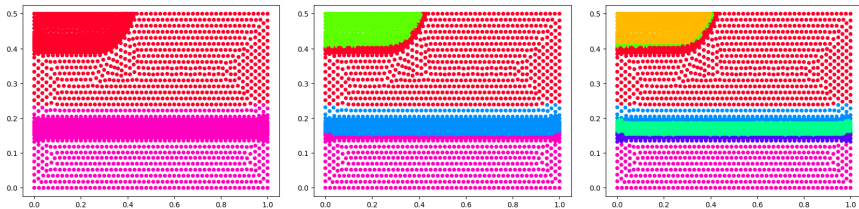
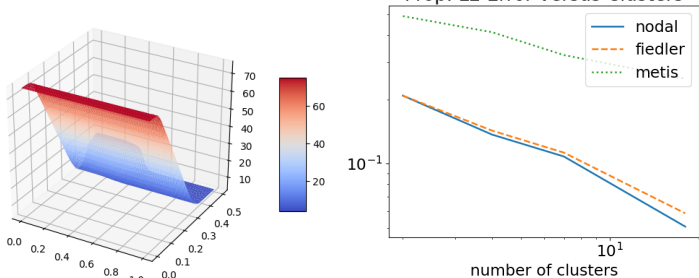
Logistic Test-Function



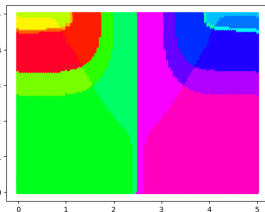
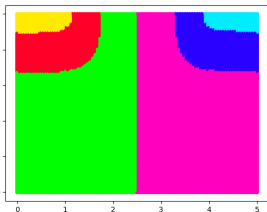
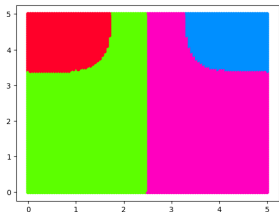
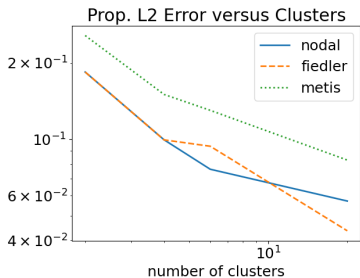
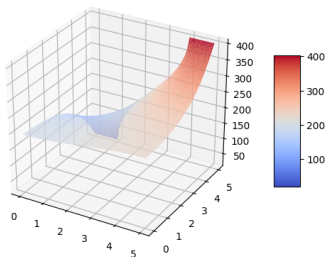
Diode



Prop. L2 Error versus Clusters



Resistor



Bibliography



L. J. GRADY AND J. R. POLIMENI, *Discrete calculus: Applied analysis on graphs for computational science*, Springer Science & Business Media, 2010.



N. TRASK, A. HUANG, AND X. HU, *Enforcing exact physics in scientific machine learning: a data-driven exterior calculus on graphs*, arXiv preprint arXiv:2012.11799, (2020).



J. C. URSCHEL, *Nodal decompositions of graphs*, Linear Algebra and its Applications, 539 (2018), pp. 60–71.



S. ZELDITCH, *Eigenfunctions and nodal sets*, arXiv preprint arXiv:1205.2812, (2012).