Greedy Fiedler Spectral Partitioning for Data-driven Discrete Exterior Calculus

Andy Huang[1], **Nathaniel Trask**[1], **Christopher Brissette** [2], **Xiaozhe Hu** [3]
[1]Sandia National Laboratories, NM
[2] Rensselaer Polytechnic Institute, NY
[3] Tufts University, MA
ahuang@sandia.gov, natrask@sandia.gov, brissc@rpi.edu, xiaozhe.hu@tufts.edu

Abstract

The data-driven discrete exterior calculus (DDEC) structure provides a novel machine learning architecture for discovering structure-preserving models which govern data, allowing for example machine learning of reduced order models for complex continuum scale physical systems. In this work, we present a Greedy Fiedler Spectral (GFS) partitioning method to obtain a chain complex structure to support DDEC models, incorporating synthetic data obtained from high-fidelity solutions to partial differential equations. We provide justification for the effectiveness of the resulting chain complex and demonstrate its DDEC model trained for Darcy flow on a heterogeneous domain.

Data-driven Discrete Exterior Calculus

A discrete exterior calculus structure (Desbrun et al. 2005) provides a notion of differentiation beyond the differentiation of scalar-valued functions. The familiar div, $grad$, and $curl$ operators from vector calculus in $\mathbb{R}^3$ satisfy the identities $div \circ curl = 0$ and $curl \circ grad = 0$, and also appear in the integral Green's, Stokes', and divergence theorems. Although back-propagation provides the differentiation of tensors by regarding the components as scalar functions, preservation of these identities are critical to handling a number of physical systems, particularly electromagnetic phenomena.

In the context of finite element approximation, compatible spatial discretizations construct approximations faithful to these underlying structures, and form the bedrock of discretization of multiphysics problems, particularly for second-order elliptic systems (Arnold et al. 2007; Bochev and Hyman 2006; Arnold, Falk, and Winther 2006). We consider in this work the design of machine learning architectures which preserve a similar structure.

Stokes' theorem plays a central role for expressing conservation laws on a Riemannian manifold: given a differential form ω on a manifold M, we have

$$\int_{\Omega} d\omega = \int_{\partial\Omega} \omega \tag{1}$$

The data-driven exterior calculus (DDEC) (introduced in (Trask, Huang, and Hu 2020)) is a parameterization of the graph exterior calculus which allows the application of popular back propagation techniques from machine learning to learn coordinate-free vector calculus operators from data. The resulting operators are compatible with a discrete version of Stokes' theorem, allowing the learning of physical conservation laws which naturally preserve conservation principles, provide solvability guarantees for second-order elliptic operators, and naturally handle diffusion problems. A key ingredient for this calculus is a chain complex, which informally corresponds to a graph structure generalizing meshes typically used to partition space. We propose a greedy Fiedler spectral (GFS) partitioning scheme which extracts this graph structure from eigenspectrum information of a given data set.

A motivating example: resistor networks

Before introducing DDEC in the abstract, we provide a familiar example to motivate both the discrete exterior calculus and the chain complex structure on which it is defined. The discrete exterior calculus is best exemplified by Kirchhoff's current law and Ohm's law from circuit theory (see (Weyl 1923), (Smale 1972), and (Bamberg and Sternberg 1991)).

As a concrete example, consider an electrical circuit on the graph in Figure 1. Suppose nodal electric potentials are

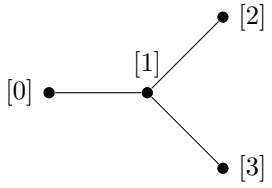

Figure 1: A simple circuit on a graph with 4 nodes.

known: the voltages at nodes [0], [1], [2], and [3] are $10V$, $5V$, $3V$, and $2V$, respectively; we represent this fact by the vector $V = [10, 5, 3, 2]^T$ whose entry i corresponds to the voltage $V([i])$ at node $[i]$. Further, suppose that the oriented current on the edge $[0, 1]$ is $1A$, on edge $[1, 2]$ is $0.7A$, and on edge $[1, 3]$ is $0.3A$; we order the edge labels lexicographically and so represent these currents as $I = [1.0, 0.7, 0.3]$.

The discrete exterior calculus boundary operator ∂ is

$$\partial = \begin{bmatrix} -1 & +1 & & \\ & -1 & +1 & \\ & -1 & & +1 \end{bmatrix} \tag{2}$$

and the coboundary operator is defined as $\delta := \partial^T$; explicitly, it maps the edge $[i, j]$ to the linear combination of nodes $+[j] - [i]$ accounting for edge orientation.

In this language of discrete exterior calculus, Kirchhoff's current law states that δI vanishes on interior nodes. Indeed,

$$\delta I = \begin{bmatrix} -1 & & \\ +1 & -1 & -1 \\ & +1 & \\ & & +1 \end{bmatrix} \begin{bmatrix} 1.0 \\ 0.7 \\ 0.3 \end{bmatrix} = \begin{bmatrix} -1.0 \\ 0.0 \\ 0.7 \\ 0.3 \end{bmatrix} \tag{3}$$

and the second entry vanishes since it is the discrete graph divergence of I at node $[1]$, i.e., it is the sum of all edge currents leaving node $[1]$. Ohm's law is realized by the equation

$$\partial V = RI \tag{4}$$

for which R represents the resistances on the edges. R is a matrix with diagonal entries $R_{[0,1]}$, $R_{[1,2]}$, and $R_{[1,3]}$ corresponding to the resistances at the respective edges.

However, Ohm's Law is more accurately expressed as

$$R^{-1}\partial V = I \tag{5}$$

and the operator $\delta^* := R^{-1}\partial$ is a metric-dependent gradient (as opposed to the purely combinatorial ∂). The discrete Stokes' theorem states

$$\int_{[0,1]} \delta^* V = \int_{\partial[0,1]} V \tag{6}$$

That this holds is the expression of Ohm's law. For example, on the edge $[0, 1]$, we see that the left-hand side

$$\begin{aligned} \int_{[0,1]} \delta^* V &= \int_{[0,1]} R^{-1} I \\ &= R_{[0,1]} I([0, 1]) \end{aligned}$$

is equal to the right-hand side

$$\begin{aligned} \int_{\partial_0[0,1]} V &= \int_{[1]-[0]} V \\ &= V([1]) - V([0]) \end{aligned}$$

since $V([1]) - V([0]) = R_{[0,1]} I([0, 1])$.

While simple, this resistor network demonstrates how conservation statements can be encoded onto a graph structure which exposes trainable *metric information* (the resistances) to parameterize diffusion processes. In this example, the circuit network is a 1-dimensional chain complex which supports conservation (encoded in Kirchoff's law) and diffusion (encoded in Ohm's law) expressed using graph divergence and gradient operators. More generally, the DDEC expresses a class of such parameterized vector calculus models, provided a chain complex can be obtained.

The chain complex provided by the GFS partitioning proposed in this paper provides in this example a circuit topology for a given data set.

The data-driven exterior calculus

We gather the requisite definitions from DDEC to define the chain complex and how it's used, referring the interested reader to (Trask, Huang, and Hu 2020) for details. Let $\mathbf{N} = \{n_i\}_{i=1}^{N_N}$ denote a set of nodes. We embed $\mathbf{N}$ in $\mathbb{R}^d$ by associating with each node a unique position $\mathbf{p}_i \subset \mathbf{R}^d$, $i \in 1, \ldots, N_N$. A k-clique is an ordered tuple of k nodes; it is assigned an orientation equal the sign of the permutation which puts these k nodes in increasing order. We associate each k-clique with the $(k-1)$-simplex spanned by its k node positions. A k-chain c_k is a formal $\mathbb{R}$-linear combination of $(k+1)$-cliques, and we denote the space of k-chains by C_k. The *boundary operator* $\partial_k : C_{k+1} \to C_k$ is defined via

$$\partial_k[n_1, ..., n_k] = \sum_i^k (-1)^{i-1}[n_1, ..., \widehat{n_i}, \ldots, n_k], \tag{7}$$

where $\widehat{\cdot}$ denotes an omitted entry. It satisfies the property $\partial_{k-1} \circ \partial_k = 0$. This forms an exact sequence

$$C_0 \xleftarrow[\partial_0]{} C_1 \xleftarrow[\partial_1]{} \ldots \xleftarrow[\partial_{d-1}]{} C_d \tag{8}$$

with the convention that ∂_{-1} maps C_0 to the empty set.

The cochains C^k consist of linear functionals acting on C_k. Given $\phi \in C^k$, we denote the value associated with the k-chain $t_{i_1 i_2 \ldots i_k}$ via the shorthand $\phi_{i_1 i_2 \ldots i_k} := \phi(t_{i_1 i_2 \ldots i_k})$. Note that the cochains inherit the orientation of the underlying chains, e.g. $\phi_{ij} = -\phi_{ji}$ via the definition of π. Introducing the *coboundary operator* $\delta_k : C^k \to C^{k+1}$, we next arrive at the following cochain complex

$$C^0 \xrightarrow{\delta_0} C^1 \xrightarrow{\delta_1} \ldots \xrightarrow{\delta_{d-1}} C^d \tag{9}$$

the resulting coboundary operators result in the traditional graph div/grad/curl from the combinatorial Hodge theory [REF]. Duality between the combinatorial boundary and coboundary operators is expressed through the inner product

$$\langle \phi, \partial_k t \rangle = \langle \delta_k \phi, t \rangle. \tag{10}$$

The DDEC operators on the chain complex are parameterized metric perturbations of these, defined as

$$\mathsf{d}_k := \mathbf{B}_{k+1} \delta_k \mathbf{B}_k^{-1}, \quad \text{and} \quad \mathsf{d}_k^* = \mathbf{D}_k^{-1} \delta_k^* \mathbf{D}_{k+1} \tag{11}$$

where $\mathbf{B}_k$ and $\mathbf{D}_k$ are diagonal matrices for each k.

In this paper, we specialize to chain complexes whose largest cells are 2-dimensional, where we introduce notation alluding to the vector calculus operators:

$$CURL_h = \mathbf{B}_1 \delta_0 \mathbf{B}_0^{-1}, \qquad DIV_h = \mathbf{B}_2 \delta_1 \mathbf{B}_1^{-1}, \tag{12}$$

$$CURL_h^* = \mathbf{D}_0^{-1} \delta_0^* \mathbf{D}_1, \quad GRAD_h^* = \mathbf{D}_1^{-1} \delta_1^* \mathbf{D}_2, \tag{13}$$

DDEC chain complex through mesh partitioning

Given a domain mesh M consisting of n-dimensional cells used in a finite element discretization of a physics PDE problem, it may be tempting to use M itself as a graph structure. However, working with M directly is typically prohibitively expensive since it is subject to the curse of dimensionality. In addition, with a solution to the PDE on M in hand, it is

suitable to incorporate that physics information in obtaining a coarse representation of M for use with a DDEC.

We now describe partitioning of the cells of M as a means to obtain a chain complex which comes equipped with a DDEC structure. We consider the topological dual graph T to M. Its construction is as follows: for every n-cell of M, associate to it a node of T; for every pair of n-cells of M sharing an $(n-1)$-dimensional face, create an edge between their corresponding nodes of T; by repeating this process, i.e., creating a k-cell for T if $k+1$ corresponding n-cells of M share an $(n-k)$-dimensional face, then T becomes the topological dual to the mesh M. This is a chain complex with combinatorial δ_k and δ_k^* operators as described in the preceding section.

On the one hand, this discrete exterior calculus structure on T itself can be perturbed into a DDEC by introducing the B_k and D_k metrization as in (11). However, T is typically prohibitively large and subject to the curse of dimensionality. Instead, by partitioning the nodes of T, a coarsened chain complex can be obtained which naturally admits a DDEC structure. As noticed in (Trask, Huang, and Hu 2020), nodal functions on the fine-scale chain complex are projected onto the coarsened chain complex through averaging over each partition. Since DDEC training involves minimizing the error through approximation by these coarse functions, we are motivated to obtain a coarse chain complex which has good function approximation quality.

We illustrate the induced chain complex coarsening by a partitioning of the nodes of T in the case it is a 2-dimensional chain complex. In Figure 2, F_0, F_1, and F_2 denote the 0-, 1-, and 2-chains on T, i.e., the nodes, edges, and cells; F_0, F^1, and F^2 denote the 0-, 1-, and 2-chains on T, i.e., the linear functionals on the nodes, edges, and cells.

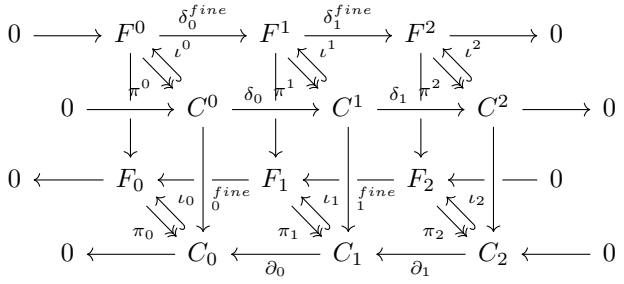

Figure 2: A diagram of domain partitioning for a 2-dimensional chain complex. This summarizes the relationship between the nodes, edges, and faces of a T and the nodes, edges, and faces of a coarsened chain complex resulting from partitioning T.

With a partition π_2 of the nodes of T, we agglomerate the fine-scale chains of F_2 into coarse chains C_2. By applying the boundary operator to these chains, we establish the coarse 1-chains C_1 from F_1, etc. For more details of this partitioning process, we defer to (Trask, Huang, and Hu 2020). For the purposes of this paper, it suffices to produce a partitioning of the highest dimensional cells because it induces a partitioning of the entire chain complex. We will see in Figure 8 an example induced chain complex partitioning where the coarse cells correspond to the color-coded regions.

Greedy Fiedler Spectral Partitioning

We aim to apply a spectral partitioning scheme on the nodes of T. Spectral methods have been used for compression of information and partitioning in many domains. Examples include JPEG image compression, manifold eigenfunction parameterization (Jones, Maggioni, and Schul 2008), segmentation of 3D models (Sharma et al. 2009), and partitioning of finite element meshes for parallel processing (Kaveh and Davaran 1999).

Graph Laplacian and Fiedler partitioning

To define spectral partitioning of T, we first define some some mathematical objects used for that purpose. Consider the graph obtained from the nodes and edges of T. The Laplacian on T is defined as

$$L = D - A,$$

where D is the degree of the nodes and A is the (weighted) adjacency matrix. For non-negative weights, it was first noticed in (Fiedler 1975) that the eigenvector (called the Fiedler eigenvector) of L corresponding to the second smallest eigenvalue ψ can be used to partition T into two connected subgraphs, T_+ and T_-, for which ψ is positive on the nodes of T_+ and negative on the nodes of T_-, and whose edges of T not belonging to T_+ or T_- are exactly those on which ψ takes a positive value on one terminal node and a negative value on the other (ignoring the special case when ψ may take a value of 0 on some nodes). In this way, the value 0 can be treated as a cut for this Fiedler eigenvector of the graph Laplacian and used for graph partitioning.

Typically, a recursive hierarchical bi-partitioning is used to further partition a graph beyond the first cut. Each of the two partitioned subgraphs can be cut using their Fiedler eigenvector, and so on. However, for the purposes of good approximation, since the cochains take the average value of the field on each partition, any partitioning of the mesh near small field variations is unnecessary. So, for a given nodal field V on T, we consider a different iterative approach: as in adaptive quadrature, at each iteration, we choose to cut the partition with the worst approximation by a piecewise constant function (which is bounded a priori by the L^2 norm of V on each partition). The GFS partitioning scheme is detailed in Algorithm 1. We use $|\nabla V|$ for the node weights of T, and each partition weight is the sum of all nodal weights within the partition.

Implementation

We implemented the GFS partitioning in PYTHON using the packages NUMPY, SCIPY, and NETWORKX for graph manipulation. The pseudocode in Algorithm 1 can be implemented using NETWORKX in a straightforward manner - calculuation of edge weights by partition, deletion of edges, and obtaining the positive and negative connected components after a Fiedler eigenvector nodal cut are possible using NETWORKX. The Fiedler eigenvectors were calculated using

Data: points $X = \{x_n\}$, field values $V_n = V(x_n)$
Result: partitioning $\Pi = (P_1, \ldots, P_{Nparts})$ such that $\bigsqcup_{i=1}^{Nparts} P_i = X$
Construct graph $T = (X, X_{ij} = (x_i, x_j))$
Assign edge weights $|V_j - V_i|$ on X_{ij}
Initialize: $\Pi = [G]$, $W_{parts} = [W_G]$
for $j \in \{1, ..., \texttt{Nparts}\}$ **do**
  P $\leftarrow$ G$_{parts}[-1]$
  Compute Fiedler eigenvector v_P of P
  for $e_{ij} \in E_P$ **do**
    if $v_P(i) \cdot v_P(j) < 0$ **then**
      Remove edge e_{ij} from P
    end
  end
  Identify partition: $P = P_+ \sqcup P_-$
  Update: $G_{parts}[-1] = P_-$
  Append: $G_{parts} += [P_+]$
  for $G \in G_{parts}$ **do**
    Calculate and update partition weights
  end
  Sort G_{parts} with respect to partition weights
end

Algorithm 1: Greedy Fiedler Spectral Partitioning algorithm for partitioning the domain of a field.

`networkx.fiedler_vector`. We note that a more efficient implementation can be made possible by evaluating subgraph Laplacians efficiently; since $L = D - A$, and A restricts to subgraphs simply by referincing the appropriate submatrix, only the degree matrix D needs to be updated properly between iterative cuts.

One-dimensional function approximation example

To better understand the GFS algorithm, we apply it to a simple one-dimensional function $f(x) = tanh(x) + 2$ on $x \in [-5, 5]$. This function plotted in Figure 3 along with approximate functions given by uniform partitions and GFS partitions for comparison (the average value of f over each partition is used for the approximate functions). The fine-scale graph is given by the nodes $\{\frac{k}{1000} | k = 0, 1, \ldots, 1000\}$ and the edges are $\{[\frac{k}{1000}, \frac{k+1}{1000}] | k = 0, 1, \ldots, 999\}$. Uniform partitioning here refers to cutting at the equally spaced points $\frac{k}{n}$ for $k = 1, 2, \ldots, n-1$ for an n-partition scheme.

Intuitively, to minimize the error of a coarse cochain approximation - which is piecewise constant on each partition - it is prudent for a partitioning algorithm to apply most refinement between $[-2, 2]$ where the derivative is great. Given the weights $|V_j - V_i|$ in Algorithm 1 we can see that partitions within the interval $[-2, 2]$ will carry the most partition weights, and thus the greedy iteration to partition by the heaviest weight will result in most approximation fidelity in this region. Indeed, we see this refinement of partitioning over $[-2, 2]$ in Figure 3. Furthermore, Figure 4 compares the GFS algorithm to a uniform partitioning scheme, and demonstrates GFS yielding a lower L^2-error, as expected. As shown in Figure 4, while the rate of convergence is the same,

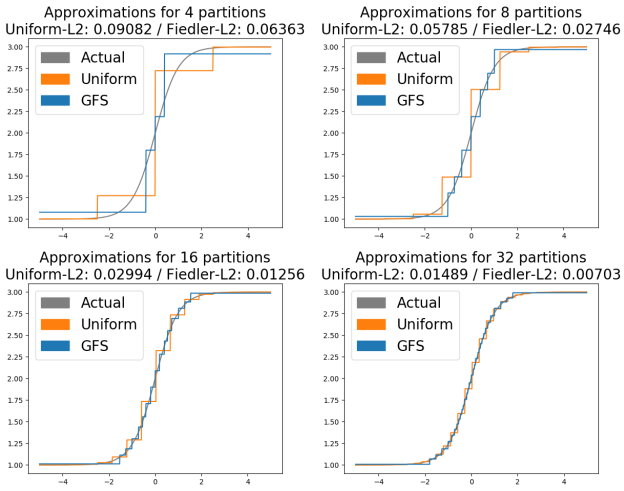

Figure 3: GFS approximations of $f(x) = tanh(x) + 2$ versus a uniform approximation with the same number of partitions. The four plots display the cochain approximations of f defined by the GFS approximation and the uniform approximation with the same number of partitions. The L^2 errors associated with each method are also shown.

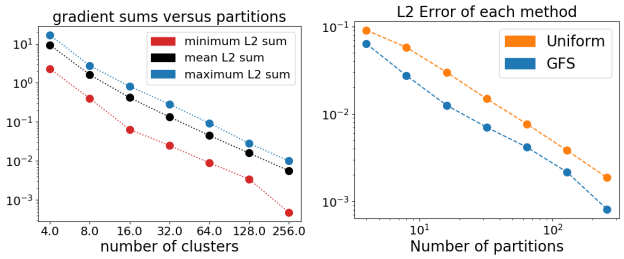

Figure 4: Error and partition weight versus number of partitions, again comparing partitioning by GFS and a uniform partitioning scheme on $f(x) = tanh(x) + 2$. The left plot shows the maximum, mean, and minimum over all partition weights. The right plot displays the resulting L^2 error of GFS approximation versus uniform approximation, as a function of the number of partitions.

the GFS method performs better and has less L^2 error than that of uniform partitioning.

Two-dimensional electrostatics/Darcy flow example

Next, we consider an application of the GFS partitioning to data obtained from a textbook electrostatics problem. Consider an infinite dialectric cylinder in $\mathbb{R}^3$ co-axial with the z-axis, of radius b and relative capacivity K. Due to the invariance in the z-direction, the problem is analyzed on the $x - y$ plane. In polar coordinates, the background electric field is given by $V_{back} = E_{inf} r \cos(\theta)$. The induced electric potential due to the charge on the cylinder which matches the boundary conditions imposed by V_{back} on the cylinder

surface interface is described analytically by

$$V(r,\theta) = \begin{cases} E_{inf}\left(-\frac{K-1}{K+1}\frac{b^2}{r}\right)\cos(\theta) & \text{for } r > b \\ \left(\frac{E_{inf}}{K+1} - E_{inf}\right) r\cos(\theta) & \text{for } r \leq b \end{cases} \tag{14}$$

Notably, a sharp interface is experienced by the gradient of the field at $r = b$ as depicted n Figure 5.

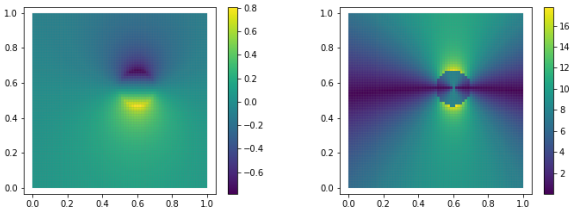

Figure 5: Electric potential V (left) and norm of the gradient of the electric potential $||\nabla V||$ (right) over a Cartesian mesh of $[0,1] \times [0,1]$.

We obtained point data by sampling the analytic expression for the electric potential due to the dialectric cylinder. For the purpose of this example, we chose a simple Cartesian mesh of $[0,1] \times [0,1]$ with 70 cells in each dimension. Figure 6 displays the partitioning of the 70×70 cells, comparing the GFS partition to a uniform partitioning obtained by METIS. We see a notable clustering of partitions near $r = b$. Furthermore, in Figure 7, we display the better approximation error of GFS compared to METIS.

Using GFS partitioning for DDEC: Example Electrostatics/Darcy flow problem

Finally, we display the utility of a GFS partitioning to obtain a chain complex supporting a DDEC. The DDEC will be trained to solve a system of PDEs which includes a conservation law, resulting in a machine learned surrogate model.

First, we apply GFS partitioning to obtain a coarse chain complex. In Figure 8, we display the GFS partitioning of the 70×70 unit square fine-scale cells into 16 coarse cells, and the resulting chain complex. Increased resolution is seen near the $r = b$ interface. Nodes, edges, and faces shaded by different colors to distinguish the different components.

Next, on that GFS partitioning-obtained chain complex, we pose the electrostatics problem

$$\vec{D} = -\epsilon\nabla\phi \tag{15}$$

$$\nabla \cdot \vec{D} = 0 \tag{16}$$

where $\vec{D}$, ϕ, ϵ are the electric displacement field, the electric potential, and the inhomogeneous capacivity. Mathematically equivalently, this system describes the Darcy flow of a fluid, where these variables correspond to the velocity vector field, the pressure, and the permeability, respectively, and where the symbols $\vec{u}$, p, and κ more commonly employed. In the DDEC formulation, we have the system of equations

$$\begin{aligned} \mathbf{F} + \kappa\nabla\phi &= 0 \\ \nabla \cdot \mathbf{F} &= f \end{aligned} \quad \longrightarrow \quad \begin{aligned} \mathbf{w}_1 - GRAD_h^* \mathbf{u}_0 &= 0 \\ DIV_h \mathbf{w}_1 &= \mathbf{f}_0 \end{aligned} \tag{17}$$

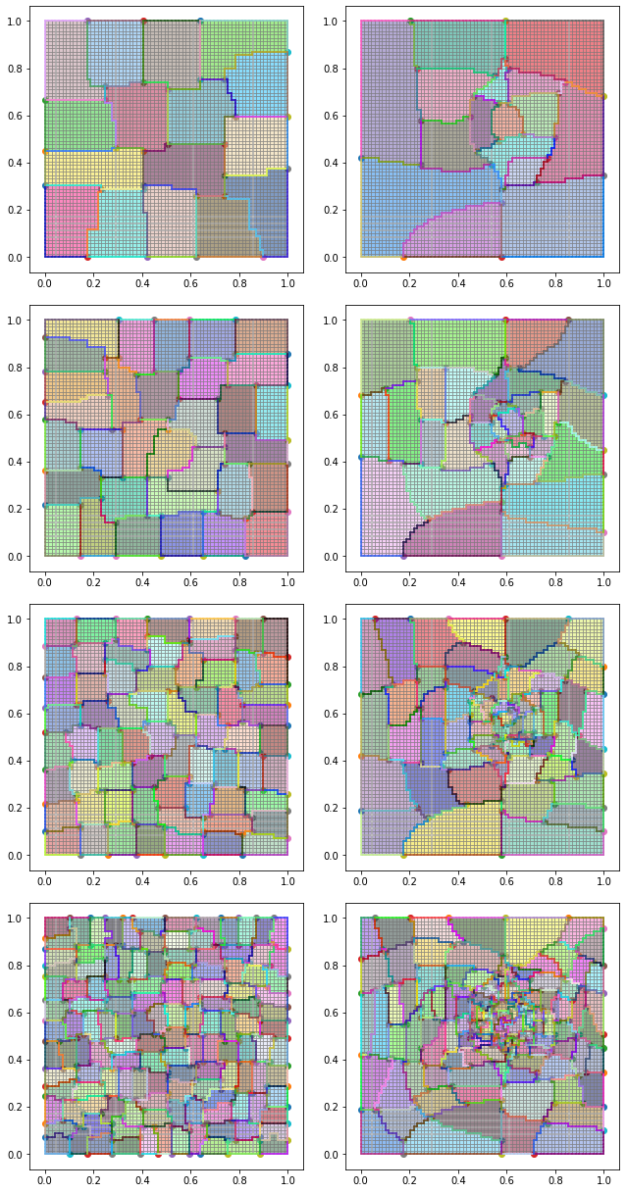

Figure 6: Juxtaposition of domain partitioning performed by METIS (*left column*) versus GFS (*right column*), using electric field magnitude through a dialectric cylinder as node weights. The rows correspond to 16, 32, 64, and 128 resulting partitions, in increasing order from top to bottom.

where we have identified $\mathbf{F}$ as the electric displacement field and the ϕ as the electric potential. The field κ was taken to be 10 inside the circle of radius 0.1 centered at $(0.6, 0.57)$; the asymmetry of this configuration was chosen to break the symmetry of domain bisection.

Finally, we train this DDEC model on the boundary value data of the analytic solution (following the procedure in (Trask, Huang, and Hu 2020) to learn $\mathbf{F}$ and ϕ). Figure 9 displays good agreement between the DDEC ϕ with the average of the analytic solution over each partition.

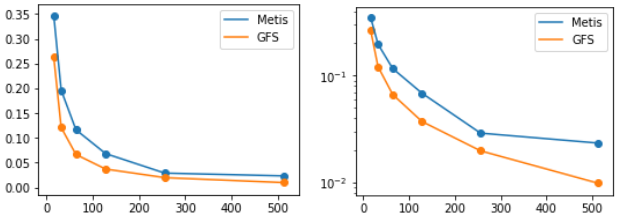

Figure 7: Approximation error comparison between piecewise constant averaging from METIS and GFS.

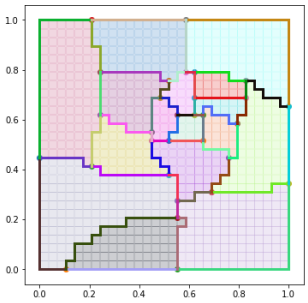

Figure 8: The DDEC complex obtained by GFS partitioning of the unit square mesh using the electric potential gradient. Partitioning results in coarse cells, edges, and nodes.

Conclusion and Further Work

In this article, we addressed the question of how to obtain a chain complex for use in the data-driven discrete exterior calculus. We introduced a Greedy Fiedler Spectral partitioning scheme of a finite-element mesh to obtain a coarse chain complex with favorable approximation of a fine-scale scalar field on the mesh. The question of how to obtain a suitable graph to support data-driven modeling is requisite for successful application of the DDEC. We demonstrate the utility of this calculus in a number of upcoming works.

Acknowledgments. 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-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. SAND Number: SAND2020-XXXX.

The work of N. Trask is supported by the U.S. Department of Energy, Office of Advanced Scientific Computing Research under the Collaboratory on Mathematics and Physics-Informed Learning Machines for Multiscale and Multiphysics Problems (PhILMs) project. N. Trask is supported by the Department of Energy early career program. The work of A. Huang, N. Trask, and C. Brissette is supported by the Sandia National Laboratories Laboratory Directed Research and Development (LDRD) program.

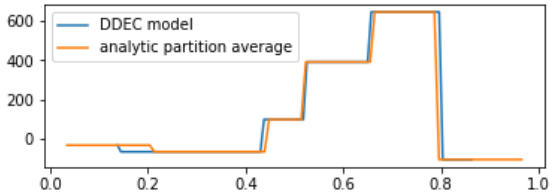

Figure 9: Trained DDEC model compared to analytic solution averaged over each partition (sliced through $y = 0.53$).

References

Arnold, D. N.; Bochev, P. B.; Lehoucq, R. B.; Nicolaides, R. A.; and Shashkov, M. 2007. *Compatible spatial discretizations*, volume 142. Springer Science & Business Media.

Arnold, D. N.; Falk, R. S.; and Winther, R. 2006. Finite element exterior calculus, homological techniques, and applications .

Bamberg, P.; and Sternberg, S. 1991. *A Course in Mathematics for Students of Physics*, volume 2.

Bochev, P. B.; and Hyman, J. M. 2006. Principles of mimetic discretizations of differential operators. In *Compatible spatial discretizations*, 89–119. Springer.

Desbrun, M.; Hirani, A. N.; Leok, M.; and Marsden, J. E. 2005. Discrete Exterior Calculus .

Fiedler, M. 1975. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal* 25: 619–633.

Jones, P. W.; Maggioni, M.; and Schul, R. 2008. Manifold parametrizations by eigenfunctions of the Laplacian and heat kernels. *Proceedings of the National Academy of Sciences* 105(6): 1803–1808. ISSN 0027-8424. doi:10.1073/pnas.0710175104. URL https://www.pnas.org/content/105/6/1803.

Kaveh, A.; and Davaran, A. 1999. Spectral bisection of adaptive finite element meshes for parallel processing. *Computers Structures* 70(3): 315 – 323. ISSN 0045-7949. doi:https://doi.org/10.1016/S0045-7949(98)00170-9. URL http://www.sciencedirect.com/science/article/pii/S0045794998001709.

Sharma, A.; Horaud, R.; Knossow, D.; and Von Lavante, E. 2009. Mesh Segmentation Using Laplacian Eigenvectors and Gaussian Mixtures. In Souvenir, R., ed., *AAAI Fall Symposium on Manifold Learning and its Applications*, 50–56. AAAI, Arlington, VA, United States: AAAI Press. URL https://hal.inria.fr/inria-00446990.

Smale, S. 1972. On the mathematical foundations of electrical circuit theory. *J. Differential Geom.* 7(1-2): 193–210. doi:10.4310/jdg/1214430827. URL https://doi.org/10.4310/jdg/1214430827.

Trask, N.; Huang, A.; and Hu, X. 2020. Enforcing exact physics in scientific machine learning: a data-driven discrete exterior calculus on graphs. *preprint* arXiv:2012.11799.

Weyl, H. 1923. Repartición de corriente en una red conductora. *Revista Matemática Hispano-Americana* 5: 153–164.