DATA 1: Cell Complex

Moo K. Chung

University of Wisconsin-Madison, USA

mkchung@wisc.edu

Abstract. Images in both two and three dimensions are inherently constructed by connecting squares or cubes, and can therefore be naturally modeled using a cell complex data structure. In this paper, we introduce the concept of a cell complex, with a particular focus on two-dimensional cubical complexes, as a principled mathematical representation for image data. This representation explicitly encodes nodes, edges, and faces together with their incidence and orientation relationships, providing a natural foundation for discrete geometric, topological, and flow-based analysis. MATLAB codes and sample data used in this paper are available at https://github.com/laplcebeltrami/BMI768/tree/main/cublical-complex.

1 Cell Complex

A *cell complex* is a combinatorial data structure that represents a space by gluing together building blocks of different dimensions. In a two-dimensional setting, the basic blocks are 0-cells (nodes), 1-cells (edges connecting nodes), and 2-cells (faces whose boundaries are cycles of edges) (Fig 1). Geometry is stored only through *incidence relations*: which nodes an edge connects, and which edges form the boundary of a face, together with a consistent choice of *orientation*. This representation is particularly convenient for defining discrete differential operators (incidence/boundary matrices) (Anand & Chung 2023), discrete flows on edges.

We first construct a two-dimensional cubical cell complex on a rectangular $n \times m$ grid of square faces using the MATLAB function

```
S= cellcomplex_build_checkerboard(n,m)
```

The resulting structured array `S` stores the cubical cell complex in the form

```
S =

  struct with fields:

    nodes: [25×2 double]
    edges: [40×2 double]
    faces: [1×1 struct]
```

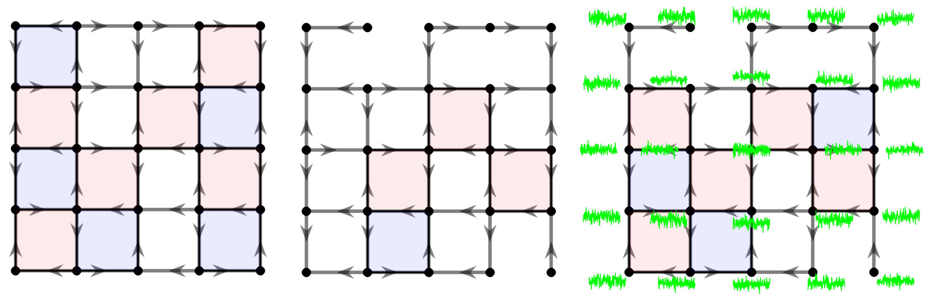

Fig. 1: Left: A 4×4 cubical cell complex `S` with a checkerboard pattern of oriented square faces, illustrating the regular grid structure of nodes, edges, and faces. Middle: A randomly perturbed cubical cell complex `Sr` obtained by randomly flipping edge orientations and removing a subset of edges according to an Erdős–Rényi type model, resulting in an irregular topology with missing faces and disrupted cycles. Right: Multivariate time series generated at each node of the perturbed complex using a VAR(p) model, where the directed edge flows serve as the ground-truth interaction structure driving the temporal dynamics.

Here, `S.nodes` contains the coordinates of the vertices, which form the $(n+1)\times(m+1)$ lattice points (i,j) and represent the 0-cells of the complex. Adjacent vertices are connected by edges stored in `S.edges`, defining the 1-cells. Each unit square determined by four neighboring vertices constitutes a 2-cell (face), recorded in `S.faces` together with its boundary information. The checkerboard construction assigns a consistent orientation to the boundary of each face, alternating between clockwise and counterclockwise orientations across the grid. As a result, every square face forms a coherently oriented directed cycle, which is essential for network flow analysis and topological analyses (Anand & Chung 2023).

```
S =

  struct with fields:

    nodes: [25×2 double]
    edges: [40×2 double]
    faces: [1×1 struct]
```

Finally, the face structure `S.faces` is stored in matrix form to enable efficient downstream computation. `S.faces` encodes the 2-cells (square faces) of the cubical complex and is itself a structured array with fields

```
S.faces

  struct with fields:
    vertices: [16×4 double]
       edges: [16×4 double]
        sign: [16×1 double]
```

Each row of `S.faces.vertices` lists the four vertex indices $[\mathrm{TL}, \mathrm{TR}, \mathrm{BR}, \mathrm{BL}]$ that define a unit square face, ordered according to the fixed boundary convention. The corresponding row of `S.faces.edges` stores the indices of the four edges that form the boundary of that face, aligned with the clockwise traversal order $\mathrm{TL} \to \mathrm{TR} \to \mathrm{BR} \to \mathrm{BL} \to \mathrm{TL}$. The field `S.faces.sign` records the orientation consistency of each face: a value of $+1$ indicates that all boundary edges follow the clockwise orientation, a value of -1 indicates a uniformly counterclockwise orientation, and a value of 0 indicates that the boundary edges are not consistently oriented. In the checkerboard construction, the orientations alternate across faces, ensuring that each square face forms a coherent directed cycle while sharing edges consistently with its neighbors.

2 Random Cell Complex

For simulations, we need to introduce randomness into the cubical cell complex through two successive perturbation steps. First, we perturb the orientation of edges with a fixed probability p. Specifically, for each edge in the complex, its direction is independently flipped with probability p, while the underlying connectivity of the grid is preserved. This operation is implemented by the function

```
Sp = cellcomplex_perturb_edges(S, p, seed)
```

where the random seed ensures reproducibility. The resulting complex `Sp` retains the same nodes and edges as the original complex, but with randomly perturbed edge orientations, thereby disrupting the original checkerboard circulation pattern.

Next, we perform an edge deletion operation to further increase structural complexity. This step follows an Erdős–Rényi (ER) type random graph model (Erdos & Rényi 1960), in which randomness is introduced by independently removing each edge with a fixed probability. In the classical $G(n, p)$ ER-model, any pair of nodes among n nodes is connected independently with probability p. In contrast, our setting starts from a fixed grid-based cell complex, and randomness is introduced by independently deleting each existing edge with probability p. This operation is carried out using

```
Spr = cellcomplex_remove_edges(Sp, p, seed),
```

producing a perturbed complex `Spr` with missing edges and, consequently, missing faces. Together, edge orientation perturbation and ER-type edge removal generate irregular cell complexes that deviate substantially from the original regular grid, providing challenging test cases for subsequent flow, cycle, and topological analyses.

3 Multivariate Time Series on Cell Complex

We next generate multivariate time series data that follow the prescribed edge flow pattern on the perturbed cell complex. To this end, we adopt a Vector Autoregressive (VAR) model (Gorrostieta et al. 2012, Melnyk & Banerjee 2016), which provides a standard framework for modeling linear dynamical interactions among multiple nodes. Given the directed edge set `Sprf.edges` and the associated edge flows `Sprf.flows`, the function `VAR_graph` simulates a VAR(p) process in which the coupling structure and interaction strengths are determined by the underlying cell complex:

```
p = 4;
noise_level = 1;
edges = Sprf.edges;
flows = Sprf.flows;
Z = VAR_graph(edges, flows, p, noise_level);
```

The parameter `p` specifies the temporal order of the vector autoregressive (VAR) model, determining how many past time points influence the current state. The parameter `noise_level` controls the variance of the additive stochastic noise injected at each time step, thereby regulating the signal-to-noise ratio of the simulated dynamics. The directed edge set `edges` together with the associated edge flows `flows` defines the coupling structure and interaction strengths among nodes, ensuring that the generated time series Z is consistent with the prescribed flow pattern on the underlying cell complex.

The VAR(p) process in `VAR_graph` built as follows:

$$Z(t) \;=\; \sum_{\ell=1}^{p} A^{(\ell)} Z(t-\ell) \;+\; \varepsilon(t),$$

where $\varepsilon(t) \sim \mathcal{N}(0, \sigma^2 I_n)$. Each lag matrix $A^{(\ell)} \in \mathbb{R}^{n\times n}$ is constructed directly from the edge flow matrix W as

$$A^{(\ell)} \;=\; \rho\, \mathbf{1}_{\{\ell=1\}} I_n \;+\; \gamma_\ell\, W^\top, \qquad \gamma_\ell = \texttt{coup} \cdot \texttt{decay}^{\,\ell-1},$$

where ρ is a small self-retention coefficient, `coup` controls the overall coupling strength between nodes, and `decay` $\in (0,1)$ imposes a geometric decay across lags. The transpose $W^\top$ ensures that an edge $i \to j$ contributes to the dynamics of node j, consistent with incoming influences. The initial states are set as

$$Z(1), \ldots, Z(p) \;\sim\; \mathcal{N}(0, \sigma^2 I_n).$$

As a result, the generated time series explicitly encode the prescribed directed flow structure of the underlying cell complex, while stochastic noise controls the signal-to-noise ratio of the observed dynamics.

PROJECT 1: Detecting Cycles from Complex Data

Goal. The goal of this project is to recover coherent cyclic structures from multivariate data and to study the robustness of these cycles under random perturbations of directionality and connectivity. While synthetic data may be used for validation, students are encouraged to apply the methodology to diverse datasets distributed in class.

Description. Students will work with data that can be represented as a network or higher-order discrete structure, such as simulated dynamical systems (DATA 1), spatially embedded networks, or real-world multivariate time series. When appropriate, the underlying structure may be modeled as a cubical or graph-based cell complex, but alternative representations such as triangle-based cell complex are also acceptable. Randomness may be introduced through perturbations of interaction direction, edge weights, or connectivity patterns, including Erdős–Rényi (ER) type edge removal or other stochastic modifications.

Multivariate time series will be analyzed using vector autoregressive VAR(p) or related linear dynamical models to estimate directed interactions from time-lagged dependencies. The inferred interactions will then be examined using tools from Topological Data Analysis (TDA) (Chung 2023), and network analysis (Chung 2019), such as cycle detection, cycle spaces, or Hodge decomposition (Anand & Chung 2023) on graphs or cell complexes. The project will focus on identifying which cyclic structures remain detectable as noise levels, perturbation strength, or data sparsity increase. Beyond cycle detection, students can investigate how topological summaries—particularly the first Betti number β_1 or related cycle-count measures—vary as a function of perturbation parameters and noise. The emphasis is on empirical exploration and interpretation rather than on formal theoretical justification, which is optional.

Learning Outcomes. Upon completion, students will gain practical experience in representing complex data using network- or cell-complex–based models, analyzing cycle structures and topological features, and relating multivariate time-series dynamics to underlying topological organization. Students will also develop intuition about the robustness and limitations of cycle-based representations in noisy and heterogeneous data settings, and learn how to quantify statistical variability and uncertainty in multivariate settings.

Bibliography

Anand, D. & Chung, M. (2023), 'Hodge-Laplacian of brain networks', *IEEE Transactions on Medical Imaging* **42**, 1563–1473.

Chung, M. (2019), *Brain Network Analysis*, Cambridge University Press, London.

Chung, M. (2023), 'PH-STAT: Statistical Inference on Persistent Homology', *arXiv:2304.05912* .

Erdos, P. & Rényi, A. (1960), 'On the evolution of random graphs', *Publicationes Mathematicae, Institute of Mathematics, Hungarian Academy of Sciences* **5**, 17–60.

Gorrostieta, C., Ombao, H., Bedard, P. & Sanes, J. (2012), 'Investigating stimulus-induced changes in connectivity using mixed effects vector autoregressive models', *NeuroImage* **59**, 3347–3355.

Melnyk, I. & Banerjee, A. (2016), Estimating structured vector autoregressive models, *in* 'International Conference on Machine Learning', PMLR, pp. 830–839.