Advection on Graphs

0.1 Overview

Advection on graphs is a natural translation of continuous advection, a partial differential equation that describes the propogation of mass in a vector field, to continuous time and discrete space. The continuous-to-discrete transition is implemented by assigning time-dependent real-valued masses to the graph's vertices, and orienting (and weighting) the edges on the graph to serve as the flow field. As far as I am aware, it has primarily been given attention by Chapman [1], who approach the subject from the perspective of coordinated control of multi-agent systems. (They mention a few independent employments of a similar construction in the context of specific models of disease spread [2], population migration [3], and input-output models in economics [4]). It is also briefly discussed by Grady and Polimeni in their *Discrete Calculus* [5]. The construction of advection on graphs is a first-order linear system $\dot{u} = -L_{adv}u$, whose matrix L_{adv} is a modified version of the standard graph Laplacian. It has asymptotic long-term behavior, and can be interpreted as the process of diffusion on a flow field.

Chapman [1] introduces advection as a modification of consensus, a system for cooperative control that has been well developed in the past twenty years (see [6],[7],[8]). Consensus is very similar to diffusion on a directed graph in that each connected component will obtain constant equilibrium values across nodes. In contrast, advection has limiting behavior that is not constant across nodes within connected components. As Chapman emphasizes, advection on graphs conserves mass. Unmentioned by Chapman but mentioned by Grady et al. when they relate advection to PageRank, this conservation property actually hints at a connection we will spend some time articulating: the relation between advection on graphs and Markov chain theory.

First, we will give some background on undirected graph diffusion (the heat equation) and the matrix differential operators connected to it. Next, we will introduce advection and observe a few preliminary properties. Third, we will articulate the relationship between the solution to advection and the stationary distribution of a Markov chain, and give an overview of relevant concepts from Markov chain theory. We then tap on Markov chain theory to describe the categories and structure of advection's limiting behavior, as well as transient behavior. In so doing, we translate the applications of Perron-Frobenius theory developed in the body of work on consensus to the advection setup. Last, we will introduce a some topological graph theory, and discuss the behavior of advection on covering graphs.

0.2 Graphs and the Graph Heat Equation

We will work with directed graphs.

Definition 0.2.0.1. A *graph* $\mathcal{G} = (V, E)$ will be a set of vertices (nodes) $V = \{1, 2, \ldots, n\}$, and a set of edges $E = \{e_1, e_2, \ldots, e_m\}$, where each edge is named by an ordered pair of edges $(i \to j)$. [1]

Definition 0.2.0.2. We define an orientation function ω whose input is a node a and an edge $e = (i, j)$ and whose output is an integer as follows

$$\omega(a, (i, j)) := \begin{cases} 1 & \text{if } a = j \\ -1 & \text{if } a = i \\ 0 & \text{otherwise} \end{cases}$$

Before we introduce the setup of advection, there are some objects in the discrete calculus on graphs setup that are useful to be acquainted with.

0.2.1 Gradient Operator

If we consider a function on the graph vertices, $u : V \to \mathbb{R}^n$, how would we describe the gradient of this function? We'd want the gradient of u to return information about the change in u with respect to the degrees of freedom, which are the edges. The gradient operator will be a $m \times n$ matrix, where m is the number of edges and n is the number of vertices. It is a map that takes u to its gradient; $D : u \to \nabla u$. The output of performing the gradient operator on u will be a vector where each element corresponds to an edge. [2]

Definition 0.2.1.1. On an oriented graph $\mathcal{G}$ we define the *gradient operator* D as $D_{i,j} = \omega(j, e_i)$.

Consider as an example the graph below and its matrix D.

[1] We avoid using v to refer to vertices since we're saving the letter to denote the velocity field in advection.

[2] What's different about the gradient operator of a graph is that it doesn't return a vector for each "locality" it operates on. In 2D continuous space, for example, the gradient at a specific point will be a vector of derivatives with respect to two directions, x and y, since the space is 2D. In the graph case, the smallest "locality" we can compute a difference at is an edge, and this can only be with respect to one dimension. In this sense the operator is more like a derivative operator.

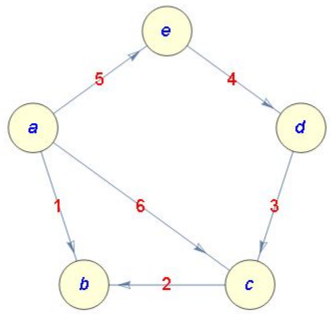


Figure 1: Example graph.

$$D = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 1 \\ -1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

0.2.2 Divergence operator.

The divergence operator is the transpose of the gradient operator. Intuitively, we want the divergence to tell us for each vertex, the sum of the flow on the edges connecting to that node.

Definition 0.2.2.1. We define the divergence operator as $D_{i,j} := \omega(i, e_j)$, and it is $n \times m$, a map from the edges to the nodes.

For the graph in 1 the divergence operator is

$$D^* = \begin{bmatrix} -1 & 0 & 0 & 0 & -1 & -1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \end{bmatrix}$$

0.2.3 Laplacian

The Laplacian operator is the divergence of the gradient:

$$L_{ij} := \sum_k D^*_{i,k} D_{k,j}$$

For $i = j$ we have (from above definitions)

$$\begin{aligned} L_{ii} &= \sum_k \omega(i, e_k)\omega(i, e_k) \\ &= N(k), \text{ the number of edges connecting to node } k \end{aligned}$$

For $i \neq j$ we have

$$L_{ij} = \sum_k \omega(i, e_k)\omega(j, e_k)$$

This is zero unless i and j both share edge e_k, in which case $\omega(i, e_k)$ and $\omega(j, e_k)$ must have opposite signs, meaning

$$L_{ij} = \begin{cases} -1 & \text{if } (i,j) \in E \\ N(i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

For the example graph above in 1, we have

$$L = \begin{bmatrix} 3 & -1 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}$$

0.2.4 The Heat Equation

For a function $u(t)$ on vertices of a graph, the heat equation is described by

$$\frac{\partial u}{\partial t} = \kappa L u$$

where κ is any constant, and L is the Laplacian defined directly above. This is an analogue of the continuous heat equation, which has been transformed from an partial differential equation to a system of ordinary differential equations.

We can solve it through the eigenvalue method: Since L is symmetric, it has an orthonormal eigenbasis; we can write the solution as a linear combination of eigenvectors:

$$\begin{aligned} \frac{\partial u}{\partial t} + \kappa L u = 0 \\ \frac{\partial}{\partial t}\Big(\sum_i b_i w_i\Big) + \kappa L\Big(\sum_i b_i w_i\Big) = 0 \end{aligned}$$

where the eigenvectors w_i have unit norm, and b_i, like u, are functions of time

$$\begin{aligned} \sum_i \frac{\partial b_i}{\partial t} w_i + \kappa \sum_i \lambda_i b_i w_i = 0 \\ \sum_i \big(\frac{\partial b_i}{\partial t} + \kappa \lambda_i b_i\big) w_i = 0 \\ \frac{\partial b_i}{\partial t} = -\kappa \lambda_i b_i \end{aligned}$$

This means the solution u(t)

$$u(t) = \sum_i b_i(t) w_i = \sum_i b_i(0) \exp\left(-\kappa \lambda_i t\right) w_i$$

$b_i(0)$ is the initial condition $u(0)$ projected onto the eigenvector w_i, ie

$$b(0) = \begin{bmatrix} | & | & \dots & | \\ w_1 & w_2 & \dots & w_n \\ | & | & \dots & | \end{bmatrix} u(0)$$

If λ_i is positive, the solution will decay to 0 along the corresponding eigenvector w_i. Since L is positive semi-definite [9], all λ_i are nonnegative. So the non-decaying solutions occur for $\lambda_i = 0$, when the corresponding

w_i is in the kernel of L. The nullity of L is in fact the space of functions on nodes where each connected component has constant values throughout. This follows from the fact that the rows of L sum to 0.

We could also circumvent the eigenvalues and eigenvectors via the matrix exponential method:

$u(t) = \exp[-\kappa L t]u(0)$

0.3 Introduction to Advection and Consensus

We introduce a discrete notion of advection by following the principle of flux from which the continuous formulation is derived. In its original continuous formulation the advection equation is

$$\frac{du}{dt} = -\nabla \cdot (\bar{v}u)$$

where $\nabla\cdot$ is the divergence operator, u is a conserved scalar quantity moving through $\bar{v}$, a time-invariant velocity field on the domain of u. This formulation sets the change in mass at a point equal to the flux through the point.

We start with a graph that has an orientation $(i \to j)$ defined on each edge. $\bar{v}$ in the continuous case becomes a vector in R^m, the space of edges. We restrict the entries of v to positive values, so that the flow is always in the direction of the orientation of the edge.

We adopt the formulation of advection on graphs proposed by Chapman and Mesbahi. [1]. We'll set up the discrete analogue of the above formulation by requiring

$$\frac{du_i}{dt} = \sum_{(j\to i)} u_j v_{(j\to i)} - \sum_{(i\to k)} u_i v_{(i\to k)} \tag{2}$$

ie, the change in mass at vertex i is the sum over edges terminating at i of the product of the velocity value at the edge and the mass at the vertex from which the edge originates, minus the sum over edges originating at i of the product of the velocity value at the edge and the mass at the i.

In matrix form we can rewrite 2 as:

$$\frac{du}{dt} = D^* V D^o u \tag{3}$$

The role of D^o, (o for originate) is to assign the masses at the vertices u to the edges originating at them. It is defined as

$$D^o(i,j) := \begin{cases} -D(i,j) & \text{if } D(i,j) < 0 \\ 0 & \text{for } D(i,j) \geq 0 \end{cases}$$

and as such is a modified gradient operator.

Since D^*VD^o is in fact a modified Laplacian, we can alternately say

$$\frac{du}{dt} = D^*VD^o u = -L_{adv}u$$

where

$$L_{adv} = \mathcal{D}_{out}(\mathcal{G}) - \mathcal{A}_{in}(\mathcal{G}) \tag{4}$$

is the advection Laplacian as defined in Chapman: The (i,i)th entry of the weighted diagonal matrix $\mathcal{D}_{out}(\mathcal{G})$ is defined as $\mathcal{D}_{i,i} := \sum\limits_{(i\to k)} v_{(i\to k)}\ \forall i$, that is, the sum of the weights of the edges going out of the ith entry, and the (i,j)th entry of the weighted adjacency matrix $\mathcal{A}_{in}$ is defined as $\mathcal{A}_{i,j} := v_{(j\to i)}$, ie the weight of the edge going from j to i.

Here is the intuition for why 2 and 3 are equivalent: Moving right to left in 3, $D^o u$ is a vector in edge space where each entry is the value of u at the vertex where that edge originates; $VD^o u$ is the same with each edge weighted by its velocity. When we apply D^* and obtain $D^*VD^o u$, we send to each vertex these weighted entries on the edges connected to them, summed and given signs according to whether the edge is entering or exiting the vertex. This is equivalent to the effect of 2, which for each vertex adds the vertex's own value weighted by its outgoing edges ($\mathcal{D}_{out}$) to the negative of the vertices feeding into it ($\mathcal{A}_{in}$). We mention the formulation in 3 to illustrate the parallel to the standard Laplacian as the gradient of the divergence.

The advection Laplacian for the previous graph, if all edge weights are set to 1, is given by

$$L_{adv} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 \\ -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Now we effectively have a modification of the heat equation, still a first order linear differential equation, where the Laplacian matrix is weighted, and is no longer symmetric.

Chapman and Mesbahi present advection as a modified verion of consensus dynamics, which are described by

$L_{cons} = \mathcal{D}_{in}(\mathcal{G}) - \mathcal{A}_{in}(\mathcal{G})$

where (i, i)th entry of the weighted diagonal matrix $\mathcal{D}_{in}(\mathcal{G})$ is defined as the sum of the weights of the edges going into the ith vertex, and $\mathcal{A}_{in}(\mathcal{G})$ stays the same.

The consensus Laplacian for the previous graph is given by

$$L_{cons} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

As with the heat equation, the consensus Laplacian has a kernel of constant vectors, as is clear from the example above. It does not, however, conserve mass as advection does, as we are about to show. First, we note a useful relation between L_{cons} and L_{adv}:

Definition 0.3.0.1. [1] The *reverse graph* of $\mathcal{G} = (V, E, W)$ is the graph $\mathcal{G}^{\mathcal{T}} = (V, E^{\mathcal{T}}, W^{\mathcal{T}})$ defined by preserving the original vertex set, and creating a new reversely oriented weighted edge set $(E^{\mathcal{T}}, W^{\mathcal{T}})$ by adding an edge $(j, i) \in E$ with weight w_{ij} if there exists $(i, j) \in E$ with weight w_{ij}.

Proposition 0.3.0.1. $L_{adv}(\mathcal{G}) = (L_{cons}(\mathcal{G}^{\mathcal{T}}))^T$

It follows from the definition that $\mathcal{D}_{in}(G) = \mathcal{D}_{out}(\mathcal{G}^{\mathcal{T}})$. Since $L_{cons}(\mathcal{G}^{\mathcal{T}}) = \mathcal{D}_{in}(\mathcal{G}^{\mathcal{T}}) - \mathcal{A}_{in}(\mathcal{G}^{\mathcal{T}})$, and (b) $\mathcal{A}_{in}(\mathcal{G}^{\mathcal{T}}) = (\mathcal{A}_{in}(\mathcal{G}))^T$, we have

$$
\begin{aligned}
(L_{cons}(\mathcal{G}^{\mathcal{T}})^T &= (\mathcal{D}_{in}(\mathcal{G}^{\mathcal{T}}) - \mathcal{A}_{in}(\mathcal{G}^{\mathcal{T}}))^T \\
&= \mathcal{D}_{in}(\mathcal{G}^{\mathcal{T}}) - (\mathcal{A}_{in}(\mathcal{G}^{\mathcal{T}}))^T \\
&= \mathcal{D}_{out}(\mathcal{G}) - \mathcal{A}_{in}(\mathcal{G}) = L_{adv}(\mathcal{G})
\end{aligned}
$$

An implication of this is that $L_{adv}(\mathcal{G})$ and $(L_{cons}(\mathcal{G}^{\mathcal{T}}))^T$ have the same rank, which we will use in 0.5.2.1

0.3.1 Solution

Equilibrium Behavior

Since both the symmetric, unweighted L and the consensus Laplacian L_{cons} have a constant kernel, their limiting behavior is to approach a constant, but the kernel of L_{adv} allows for more interesting limiting behavior. How does the structure of a graph relate to the kernel of L_{adv}? This is the question we seek to provide insight on. (We are able to directly compute the solutions, but we want to get insight beyond this).

As we had before with the heat equation, our advection equation is a first order system of linear equations, and has solutions of the form

$$
u(t) = e^{-[L_{adv}]t}u(0)
$$

Unlike for diffusion, L_{adv} is not necessarily symmetric or diagonalizable. Still, solving advection is concerned with the 0-eigenspace of L_{adv}, which we will demonstrate always exists. We will mostly be considering contexts where the vector field we impose on the edges is identically valued for all edges—the case where V is diagonal ones.

Positive semidefinite

Even though it is not symmetric, L_{adv} is positive semidefinite in the sense that for any vector $w \in \mathbb{C}^n, Re(w^* L_{adv} w) \geq 0$. (We include complex numbers since as we will see, L_{adv} can have complex eigenvectors and eigenvalues.) The proof is extremely similar to the one in [10]:

$$
\begin{aligned}
w^* L_{adv} w &= w^*(\mathcal{D}_{out} - \mathcal{A}_{\in})w \\
&= w^*(\mathcal{D} - \mathcal{A})w \\
&= w^*\mathcal{D}w - w^*\mathcal{A}w \\
&= \sum_{i=1}^{n} \mathcal{D}_{i,i}(\overline{w_i})w_i - \sum_{i,j=1}^{n} (\overline{w_i})w_j\mathcal{A}_{i,j} \\
&= \frac{1}{2}\Big(\sum_{i=1}^{n} \mathcal{D}_{i,i}(\overline{w_i})w_i - 2\sum_{i,j=1}^{n} (\overline{w_i})w_j\mathcal{A}_{i,j} + \sum_{j=1}^{n} \mathcal{D}_{j,j}(\overline{w_j})w_j\Big)
\end{aligned}
$$

Since $\mathcal{D}_{i,i} = \sum\limits_{j=1}^{n} \mathcal{A}_{i,j}$

$$
\begin{aligned}
&= \frac{1}{2}\Big(\sum_{i=1}^{n}(\sum_{j=1}^{n} \mathcal{A}_{i,j})(\overline{w_i})w_i - 2\sum_{i,j=1}^{n} (\overline{w_i})w_j\mathcal{A}_{i,j} + \sum_{j=1}^{n}(\sum_{i=1}^{n} \mathcal{A}_{i,j})(\overline{w_j})w_j\Big) \\
&= \frac{1}{2}\Big(\sum_{i,j=1}^{n} \mathcal{A}_{i,j}|w_i|^2 - 2\sum_{i,j=1}^{n} (\overline{w_i})w_j\mathcal{A}_{i,j} + \sum_{i,j=1}^{n} \mathcal{A}_{i,j}|w_j|^2\Big) \\
&= \frac{1}{2}\Big(\sum_{i,j=1}^{n} \mathcal{A}_{i,j}|w_i|^2 - \sum_{i,j=1}^{n} (\overline{w_i})w_j\mathcal{A}_{i,j} - \sum_{i,j=1}^{n} (\overline{w_j})w_i\mathcal{A}_{i,j} + \sum_{i,j=1}^{n} \mathcal{A}_{i,j}|w_j|^2\Big) \\
&= \frac{1}{2}\Big(\sum_{i,j=1}^{n} \mathcal{A}_{i,j}(w_i - w_j)(\overline{w_i} - \overline{w_j})\Big) \\
w^* L w &= \frac{1}{2}\Big(\sum_{i,j=1}^{n} \mathcal{A}_{i,j}(w_i - w_j)(\overline{w_i - w_j})\Big) = \frac{1}{2}\Big(\sum_{i,j=1}^{n} \mathcal{A}_{i,j}|w_i - w_j|^2\Big) \geq 0
\end{aligned}
$$

since $\mathcal{A}_{i,j}$ is 0 or 1.

Having established that $Re(w^* L_{adv} w) \geq 0 \ \forall w \in \mathbb{C}$, consider an eigenvector v of L_{adv} with eigenvalue λ.

Since

$$
\begin{aligned}
Re(v^* L_{adv} v) &\geq 0 \\
Re(v^* \lambda v) &\geq 0 \\
Re(\lambda v^* v) &\geq 0 \\
Re(\lambda |v|^2) &\geq 0 \\
Re(\lambda)|v|^2 &\geq 0 \\
Re(\lambda) &\geq 0
\end{aligned}
$$

all eigenvalues of L_{adv} have nonnegative real part.

Conservation of Mass

The advection system conserves mass [1]. To show this, we need to demonstrate that $\sum\limits_i \frac{du_i}{dt} = 0$.

$$
\begin{aligned}
\sum_i \frac{du_i}{dt} &= \sum_i \Big(\sum_{(j \to i)} v_{(j \to i)} u_j - \sum_{(i \to k)} v_{(i \to k)} u_i \Big) \\
&= \sum_{(j \to i) \in E} v_{(j \to i)} u_j - \sum_{(i \to j \in E)} v_{(i \to j)} u_i \\
&= 0
\end{aligned}
$$

This result is related to the result that we show in 0.4.1.

0.3.2 A Few Illustrations

Advection on a planar grid graph

Visualizing advection on a planar graph is a good starting example to demonstrate its properties and how it imitates the continuous case.

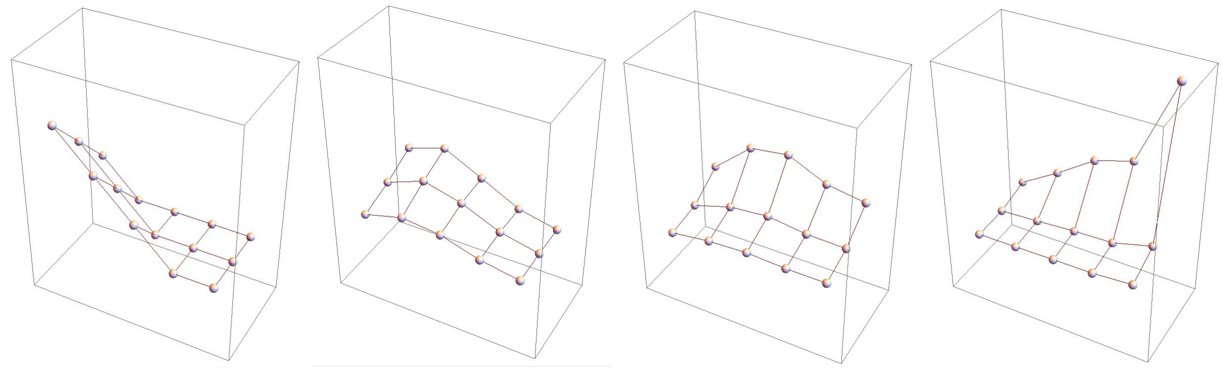


Figure 2: Advection on a grid graph with edges oriented north and east. A mass initialized in the southwest corner proprogates over to the northeast corner and stays there.

Advection on a cycle

Consider $\mathcal{G}_R(n)$, a ring graph (aka circle graph, cycle graph) with n vertices and n edges. We will orient the edges to point around the cycle $(1 \to 2, 2 \to 3, \ldots, k \to k+1, \ldots, n \to 1)$.

With this setup the matrix L_{adv} takes the form

$$L_{adv} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & -1 \\ -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{pmatrix}$$

The stereotyped structure of this matrix makes it clear that its kernel is the constant vector $c\mathbb{1}$.

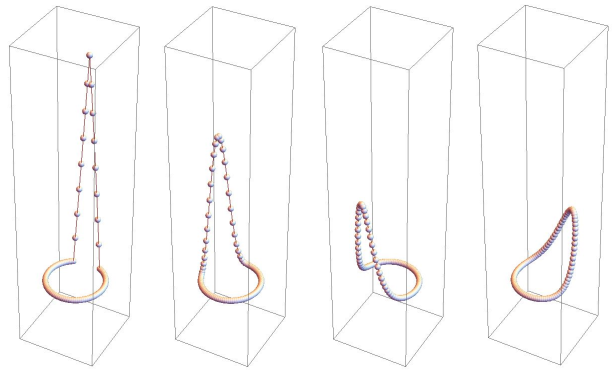


Figure 3: Advection on a cycle graph with edges oriented counterclockwise. The mass propogates around the circle, spreading out as it does so, and eventually dies down to a constant.

As it turns out, simple and highly symmetric graphs such as the torus and the circle graphs obtain a constant value in the equilibrium of advection. In fact, it follows from the definition of L_{adv} that any graph where all vertices have the same number of ingoing and outgoing edges will obtain a constant in the equilibrium.

We seek to put together a language for describing the equilibrium values of advection for more complex and asymmetric graphs.

0.4 Markov Chains and Perron-Frobenius

Characterizing the equilibrium behavior of advection on arbitrary graphs is done efficiently by recognizing this object as a cousin of a Markov chain.

0.4.1 Relation between Advection and Markov Chains

Recall our formulation of $L_{adv} = \mathcal{D}_{in}(\mathcal{G}) - \mathcal{A}_{in}(\mathcal{G})$. The solution of advection is the solution of the system $L_{adv}x = 0$. Assume for now that $\mathcal{D}_{in}$ (we'll shorten it to $\mathcal{D}$) is invertible. Now consider a related system,

$L_{adv}\mathcal{D}^{-1}y = 0$ [5]

$$\begin{aligned}
L_{adv}\mathcal{D}^{-1}y &= 0\\
(\mathcal{D}-\mathcal{A})\mathcal{D}^{-1}y &= 0\\
(\mathcal{D}\mathcal{D}^{-1}-\mathcal{A}\mathcal{D}^{-1})y &= 0\\
(I-\mathcal{A}\mathcal{D}^{-1})y &= 0\\
(\mathcal{A}\mathcal{D}^{-1}-I)y &= 0\\
\mathcal{A}\mathcal{D}^{-1}y &= y
\end{aligned}$$

$\mathcal{A}\mathcal{D}^{-1}$ is a *left stochastic matrix* in that each column sums to 1:

Recall $\mathcal{D}_{i,i} := \sum\limits_{(i\to k)} v_{(i\to k)}$, meaning $\mathcal{D}^{-1}_{i,i} := \frac{1}{\sum\limits_{(i\to k)} v_{(i\to k)}}$

and $\mathcal{A}_{i,j} := v_{(j\to i)}$, so

$$(\mathcal{A}\mathcal{D}^{-1})_{i,j} = \frac{v_{(j\to i)}}{\sum\limits_{(j\to k)} v_{(j\to k)}}$$

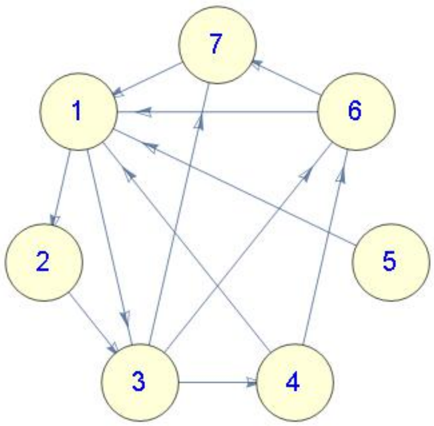
and summing across rows:

$$\sum_i (\mathcal{A}\mathcal{D}^{-1})_{i,j} = \sum_i \frac{v_{(j\to i)}}{\sum\limits_{(j\to k)} v_{(j\to k)}} = 1$$

This means that $\mathcal{A}\mathcal{D}^{-1}y = y$ can describe a Markov chain, as we will soon dive into.

If x^* is the solution to the advection system $L_{adv}x = 0$ and y^* is the solution to the Markov system $\mathcal{A}\mathcal{D}^{-1}y = y$, then we can obtain x^* by solving the Markov chain and computing $x^* = \mathcal{D}^{-1}y^*$.

Here is an illustration:



$$L_{adv}(\mathcal{G}) = \begin{bmatrix} 2 & 0 & 0 & -1 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 2 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 1 \end{bmatrix}$$

$$\mathcal{A}\mathcal{D}^{-1}(\mathcal{G}) = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} & 1 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{2} & 0 \end{bmatrix}$$

Figure 4: Advection/Markov chain connection example

0.4.2 Markov Chains

Definition 0.4.2.1. A *finite state-space discrete Markov chain* with initial distribution λ and transition matrix $P = p_{ij}$ (a stochastic matrix [3] as defined in 0.4.1) is a finite sequence $(X_0, \dots, X_n)$ where

(1) X_0 has distribution λ; ie $\mathbb{P}(X_0 = i_0) = \lambda_{i_0}$

(2) $P(X_{n+1} = i_{n+1} | X_0 = i_0, \dots, X_n = i_n) = p_{i_{n+1} i_n}$.

This definition follows [11], pg. 2 with a few modifications ($p_{i_{n+1} i_n}$ instead of $p_{i_n i_{n+1}}$ switches P to being left stochastic). $I = \{i_0, i_1, i_2, \dots, i_n\}$ is the state space of the process and corresponds to the set of vertices in a graph. We can think of a Markov chain as a object going on a random walk across the vertices of a graph. (1) says that the vertex the object starts at is chosen according to some distribution λ (which assigns a probability to each vertex). (2) says that once the starting vertex is chosen, the object proceeds to navigate the vertex space according to probabilities assigned to the edges connecting to that vertex, which is the entries $\{p_{ji}\}$.

[3]Note: we diverge from the standard convention used in statistics to express transition matrices as *right stochastic* and distributions as row vectors; to maintain consistency with our advection setups we use column vectors and left stochastic matrices.

It is then somewhat intuitive that this process should be connected to our notion of advection. As mentioned in 0.4.1, the solution to the system $L_{adv}x = 0$ and the solution to the system $\mathcal{A}\mathcal{D}^{-1}y = y$ (a Markov chain) are related via $x^* = \mathcal{D}^{-1}y^*$, so the two are tightly related. We'll discuss how the concepts of communicating classes and expected visits, developed for Markov chain theory, translate to the advection system.

0.4.3 Communicating Classes

Definition 0.4.3.1 (Communicating Classes)**.** For a Markov chain, state i leads to state j (or, $i \to j$) if $P(X_n = j \text{ for some } n \geq 0 | X_0 = i) > 0$. If $i \to j$ and $j \to i$, then the states i and j communicate and we write $i \leftrightarrow j$([11], pg. 10, 11).

Since $i \to j$ and $j \to k$ implies $i \to i$, and $i \to i\ \forall i$, the relation $\leftrightarrow$ imposes an equivalence relation on the set of vertices I, and partitions the graph $\mathcal{G}$. Each partitioned section—each set of vertices mutually reachable to each other—is called a *communicating class*.

Definition 0.4.3.2 (Irreducible)**.** A Markov chain is said to be *irreducible* if all of its states are in a single communicating class.

Definition 0.4.3.3 (Closed, recurrent, transient)**.** A communicating class C is said to be *closed* if for a given state $i \in C$, $i \to j \implies j \in C$. A closed class is a communicating class that there's no way to leave. Intuitively, for a finite Markov chain, any state in a closed class is *recurrent* in the sense that $P(X_n = i \text{ for infinitely many } n | X_0 = i) = 1$.

On the other hand, if a state is called *transient* if $P(X_n = i \text{ for infinitely many } n | X_0 = i) = 0$. This is a state that it is possible to leave and never come back to.

Theorem 0.4.3.1. *For any given communicating class C in a Markov chain, all states in C are either recurrent or transient; that is, all communicating classes are either recurrent or transient.*

Proof in Norris, [11], pg. 26.

0.4.4 Expected Visits and Stationary Distributions

Definition 0.4.4.1. A distribution y (a vector in vertex space) is said to be *invariant* if $Py = y$.

First Hits and Expected Visits

Definition 0.4.4.2. The *hitting time* (given by T_k) the first time (an integer) that the Markov chain visits state k.

The expected number of visits to a vertex i between visits to vertex k is

$y_i^k := \mathbb{E}_k \sum_{n=0}^{T_k-1} 1_{\{X_n=i\}} = \mathbb{E}\Big(\sum_{n=0}^{T_k-1} 1_{\{X_n=i\}}|X_0 = k\Big)$

where 1_A is the indicator function of whether A happened.

Definition 0.4.4.3. A matrix A is said to be *reducible* if there exists a permutation matrix P: P^TAP is of the form $\begin{bmatrix} X & Y \\ 0 & Z \end{bmatrix}$, where X, Z are square, and 0 is a 0-matrix. A matrix that is not reducible is *irreducible.*

Theorem 0.4.4.1. *([11], pg. 35-36, modified)*

If P is an irreducible matrix, then

$y^k = (y_i^k : i \in I)$ *satisfies* $Py^k = y^k$*; ie it is an invariant distribution.*

Proof

$$y_j^k = \mathbb{E}_k \sum_{n=1}^{T_k-1} 1_{\{X_n=j\}}$$

Because the state-space is finite, and because the irreducibility of P ensures that every state will eventually get to any other state (we prove this in 0.5.1.1), we can say $\mathbb{P}(T_k < \infty) = 1$, or in other words

$$y_j^k = \mathbb{E}_k \sum_{n=1}^{\infty} 1_{\{X_n=j, n \le T_k\}}$$

Equivalently

$$= \sum_{n=1}^{\infty} \mathbb{P}_k(X_n = j, n \le T_k)$$

Rephrased by summing over all possible states previous to X_n:

$$
\begin{aligned}
&= \sum_{i \in I} \sum_{n=1}^{\infty} \mathbb{P}_k(X_{n-1} = i, X_n = j, n \leq T_k) \\
&= \sum_{i \in I} \mathbb{P}_k(X_n = j | X_{n-1} = i) \sum_{n=1}^{\infty} \mathbb{P}_k(X_{n-1} = i, n \leq T_k) \\
&= \sum_{i \in I} p_{ij} \sum_{n=1}^{\infty} \mathbb{P}_k(X_{n-1} = i, n \leq T_k) \\
&= \sum_{i \in I} p_{ji} \sum_{m=0}^{\infty} \mathbb{P}_k(X_m = i, m \leq T_k - 1)
\end{aligned}
$$

We perform some reverse moves

$$
\begin{aligned}
&= \sum_{i \in I} p_{ji} \mathbb{E}_k \sum_{m=0}^{T_k - 1} 1_{\{X_m = i\}} \\
y_j^k &= \sum_{i \in I} p_{ji} y_i^k, \text{so } y^k = P y^k
\end{aligned}
$$

This theorem gives us some insight into the nature of the stationary distribution: it is induced by the effect of the graph connectivity on the expected visits to any given vertex. We will examine this further in 0.5.2.

0.4.5 Perron-Frobenius

Theorem 0.4.5.1. *(Perron-Frobenius, [12], pg. 673) If $A_{n \times n}$ is an irreducible matrix, then the following hold:*

(1) Let $\sigma(A)$ be the set of eigenvalues of A. Their maximum, $r = \rho(A) \in \sigma(A)$, exists and satisfies $r > 0$.

(2) The algebraic multiplicity of r is 1.

(3) There exists an eigenvector $x > 0 : Ax = rx$.

(4) The vector that satisfies $Ap = rp, p > 0, \|p\|_1 = 1$ is called the Perron vector. There are no nonnegative eigenvectors of A except for positive multiples of p, regardless of the eigenvalue.

We will also here mention a relevant result: if A is (left) stochastic, the largest eigenvalue $\rho(A)$ of A is 1 ([12]).

0.5 Implications for Advection

To re-emphasize, the solutions to advection are easily computable; our goal in this section is to articulate qualitative features of the solution to advection, building from the Markov chain theory and the theory on the consensus. We'll first establish some results relating the connectivity of $\mathcal{G}$ to properties of L_{adv}.

0.5.1 Graph Connectivity

Definition 0.5.1.1. On a directed graph $\mathcal{G}$, a *globally reachable* node is a node that can be reached from any other vertex by a directed path.

Definition 0.5.1.2. On a directed graph $\mathcal{G}$, a *strongly connected component* is a subgraph for which all nodes are globally reachable. A graph that is entirely one strongly connected component is called strongly connected.

A strongly connected component is equivalent to a communicating class.

Definition 0.5.1.3. A graph with at least one node that can reach every other node via a directed path we will call *weakly connected*.

This is equivalent to the notion of rooted in-branching introduced in [1].

The concepts of irreducibility (as defined in 0.4.5) and strongly connected components are linked:

Theorem 0.5.1.1. *For any strongly connected graph $\mathcal{G}$, $\mathcal{A}\mathcal{D}^{-1}(\mathcal{G})$ is irreducible.([12]).*

Proof. We'll show that $\mathcal{A}(\mathcal{G})$ for a strongly connected graph is irreducible, which implies $\mathcal{A}\mathcal{D}^{-1}(\mathcal{G})$ is irreducible.

Remark 0.5.1.1. If $\mathcal{A}$ is an adjacency matrix for some graph $\mathcal{G}$, and P is a permutation matrix, then $B = P^T\mathcal{A}P$ is the adjacency matrix for the same graph $\mathcal{G}$ with its vertices relabeled according to the permutation π given in P: $P^T\mathcal{A}P$ is $\mathcal{A}$ with its columns and rows rearranged according to the permutation.

This follows since $B_{ij} = 1 \iff \mathcal{A}_{\pi^{-1}(i)\pi^{-1}(j)} = 1$, meaning (i, j) is an edge in B's graph precisely if $(\pi^{-1}(i), \pi^{-1}(j)$ is an edge in $\mathcal{A}$'s graph, so (since π is a bijection) $\mathcal{A}$'s graph and B's graph are the same. Having established this, we proceed.

(1) $\mathcal{A}(\mathcal{G})$ reducible $\implies$ $\mathcal{G}(\mathcal{A})$ not strongly connected

By definition, $\mathcal{A}$ reducible means there is a permutation matrix P where $B = P^T \mathcal{A} P$ is of the form $\begin{bmatrix} X & Y \\ 0 & Z \end{bmatrix}$, where X, Z are square, and 0 is a 0-matrix. If B is $n \times n$, let X be $r \times r$ and Z be $(n-r) \times (n-r)$. If α is the set of vertices $\{1, 2, 3 \ldots, r\}$ and β is the set of vertices $\{r+1, \ldots, n\}$, the bottom left 0-matrix means that no vertex in α is connected to a vertex in β, so the graph with adjacency matrix B is not strongly connected, and since it's isomorphic to the graph with adjacency matrix $\mathcal{A}$ (by the above remark), $\mathcal{G}(\mathcal{A})$ is not strongly connected.

(2) $\mathcal{G}(\mathcal{A})$ not strongly connected $\implies$ $\mathcal{A}(\mathcal{G})$ reducible

If $\mathcal{G}(\mathcal{A})$ is not strongly connected, then there is a way to label the vertices so that some subset of vertices $\alpha = \{1, 2, 3 \ldots, r\}$ has no connections to another set of vertices $\beta = \{r+1, \ldots, n\}$. Then with respect to this labeling we have an adjacency matrix B of the form $\begin{bmatrix} X & Y \\ 0 & Z \end{bmatrix}$, where X, Z are square, and 0 is a 0-matrix. Let P be the permutation matrix that gives a re-labeling of the vertices from whatever their original labeling was to the labelling B uses. Then if $\mathcal{A}$ is the adjacency matrix according to the original labeling, we have $B = P^T \mathcal{A} P$, and $\mathcal{A}$ is reducible. $\square$

Lemma 0.5.1.1. *If $\mathcal{A}\mathcal{D}^{-1}(\mathcal{G})$ is irreducible, rank($L_{adv}(\mathcal{G})$) = rank($L_{cons}(\mathcal{G}) = \mathcal{D} - \mathcal{A}$) $= n-1$.*

Proof. If $\mathcal{A}\mathcal{D}^{-1}$ is irreducible, by 0.4.5 there exists an eigenvalue $\lambda_1 = \rho(\mathcal{A}\mathcal{D}^{-1})$, with algebraic multiplicity 1. As a row stochastic matrix, $\rho(\mathcal{A}\mathcal{D}^{-1}) = 1$ with algebraic multiplicity 1. This means $\dim(\ker[I - \mathcal{A}\mathcal{D}^{-1}]) = 1$, and $\text{rank}[I - \mathcal{A}\mathcal{D}^{-1}] = n-1$. [12]. And since $(I - \mathcal{A}\mathcal{D}^{-1}) = L\mathcal{D}^{-1}$, $\text{rank}(L) = n-1$ as well.

Lemma 0.5.1.2. *$\mathcal{G}$ has a globally reachable vertex if and only if rank($L_{cons}(\mathcal{G})$) $= n-1$:*

Proof in [13].

0.5.2 Graphs with Unique Equilibria

Most commonly studied in the context of consensus are cases where the Laplacian has a one-dimensional kernel and solutions are unique. We comment on the properties of advection in this case.

Lemma 0.5.2.1. *$\mathcal{G}$ is weakly connected if and only if rank($L_{adv}(\mathcal{G})$) $= n-1$:*

This follows from 0.5.1.2 and 0.3.0.1.

Lemma 0.5.2.2. *If $\mathcal{G}$ is strongly connected, the kernel of L_{adv} has strictly positive values in all its entries.*

Proof cited in [1], given in [2].

Definition 0.5.2.1. Let an *sink* be a strongly connected component $\mathcal{S}$ for which no vertex in $\mathcal{S}$ has an edge leading out of $\mathcal{S}$.

A sink is a subgraph whose vertices are contained within an closed recurrent communicating class.

Proposition 0.5.2.1. *If $\mathcal{G}$ is weakly connected and has a proper subgraph $\mathcal{S}$ that is a sink, the basis of the one-dimensional kernel of $L_{adv}(\mathcal{G})$ will have nonzero values only at the vertices within the sink.*

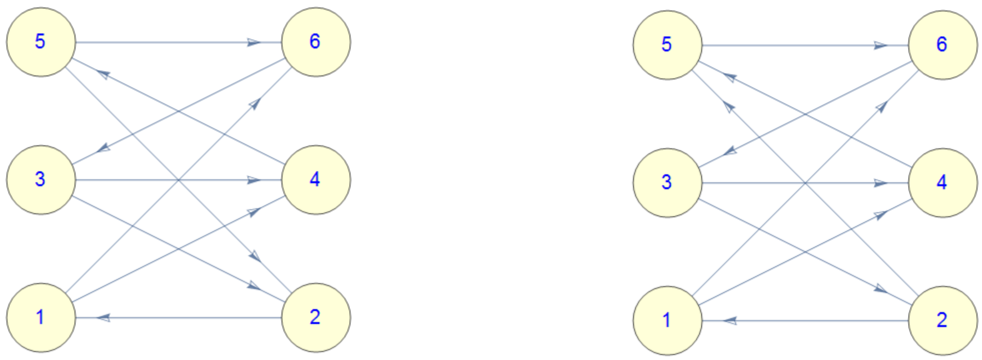
Proof. Begin with the sink subgraph alone and suppose it has l vertices. Let the basis vector for the kernel of $L_{adv}(\mathcal{S})$ be $u_{\mathcal{S}} = (u_1 u_2 \ldots u_l)^T$. By 0.5.2.2 we have $u_i > 0 \ \forall i$. By definition $\forall i, \sum_{j=1}^{l} L_{adv}(\mathcal{S})_{i,j} u_j = 0$. We showed in 0.5.2.1 that the kernel of $L_{adv}(\mathcal{G})$ must be one-dimensional. If $\alpha_{\mathcal{S}}$ is the set of vertices in $\mathcal{S}$, and $\alpha_{\mathcal{G}}$ is the set of vertices in $\mathcal{G}$, by the definition of the sink there is no vertex in $\alpha_{\mathcal{S}}$ that has an edge leading into a vertex in $\alpha_{\mathcal{G}}$. This means $L_{adv}(\mathcal{G})$ is of the form $\begin{bmatrix} L_{adv}(\mathcal{G}) & A \\ 0 & B \end{bmatrix}$, where B is $(n-l) \times (n-l)$ and 0 is a 0-matrix. Then clearly if we add $n - l$ 0's to $u_{\mathcal{S}}$ we will have a vector $(u_1 u_2 \ldots u_l \ldots 0 \ldots 0)^T$ that is the basis of $\ker(L_{adv}(\mathcal{G}))$.

So we've established that the $\dim(\ker(L_{adv})) = 1$ situation occurs when $\mathcal{G}$ is weakly connected, and that the strongly connected subgraph, if it exists, will collect the mass. Within the strongly connected subgraph, the cycle structure provides explanatory insight on the limiting solution, if we employ the results on expected visits we covered in 0.4.4.

We'll consider an example of how a small tweak to graph structure dramatically alters the solution form, and demonstrate the connection to expected visits.

$K_{3,3}$: A Demonstration of Cycle Effects

Consider the bipartite graph $\mathcal{G} = K_{3,3}$. The edges on $\mathcal{G}$ can be oriented such that half of them have two outgoing edges and one incoming, and the other three vertices have two incoming edges and one outgoing (this is $\mathcal{G}_1$, left):



(a) $\mathcal{G}_1$: A "balanced" bipartite digraph.

(b) $\mathcal{G}_2$: A bipartite digraph with a single edge reversed.

Figure 5: Two versions of $K_{3,3}$

In this case, if the edges are unweighted (all given weight 1), $\ker(L_{adv}) = c\begin{bmatrix}1 & 2 & 1 & 2 & 1 & 2\end{bmatrix}^T$—vertices on the same side have the same value.

If we switch the orientation of a single edge (leaving all edges unweighted), it is still the case that three nodes have two outgoing edges and one incoming, and three vice versa, but the equilibrium changes to $c\begin{bmatrix}1 & 2 & 4 & 5 & 7 & 8\end{bmatrix}^T$.

The theory developed for limiting distributions of irreducible Markov chains allows us to explain this difference in terms of the nodes' recurrence times. Recall from 0.4.4 that for a Markov chain with transition matrix P, the vector y^k that satisfies $Py^k = y^k$ has its entries defined by

$y_i^k =$ the number of visits to i between visits to k

(The choice of k does not change y^k except up to a constant). And recall from 0.4.1 that the advection solution is related to the Markov chain with transition matrix $\mathcal{AD}^{-1}$. When the edges are unweighted, $\mathcal{AD}^{-1}$ will be the transition matrix that assigns equal probabilities to each outgoing edge at a vertex.

We can explicitly compute the entries of y^k for $\mathcal{G}_1$ by visualizing the possible walks that begin at vertex 2, and see how long they take to return to vertex 2:

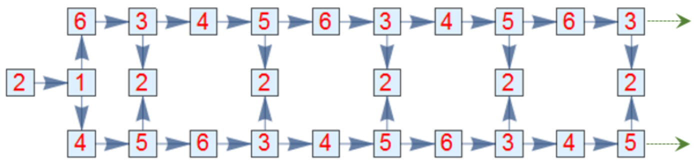


Figure 6: A walk diagram for $\mathcal{G}_1$

Consider the walks that begin and end at vertex 2. What is the expected number of times we hit vertex 3 across all these possible walks?

We can compute y_3^2, the expected visits to vertex 3 between visits to vertex 2 as

$$\sum_{\substack{\text{walks that hit}\\ \text{vertex 3 once}}} \mathrm{P(walk)}(1) + \sum_{\substack{\text{walks that hit}\\ \text{vertex 3 2x}}} \mathrm{P(walk)}(2) + \sum_{\substack{\text{walks that hit}\\ \text{vertex 3 3x}}} \mathrm{P(walk)}(3) + \dots$$

Upon examination, we can see that there are exactly 4 routes that begin and end at 2 and pass through vertex 3 once. The first takes 2 forks, the second and third take 3 forks, and the fourth takes 4 forks, so

$$\sum_{\substack{\text{walks that hit}\\ \text{vertex 3 once}}} \mathrm{P(walk)}(1) = (\tfrac{1}{2})^2(1) + (\tfrac{1}{2})^3(1) + (\tfrac{1}{2})^3(1) + (\tfrac{1}{2})^4(1)$$

Next, there are again exactly 4 routes that begin and end at 2 and pass through vertex 3 twice. The first takes 4 forks, the second and third take 5 forks, and the fourth takes 6 forks, so

$$\sum_{\substack{\text{walks that hit}\\ \text{vertex 3 2x}}} \mathrm{P(walk)}(1) = (\tfrac{1}{2})^4(2) + (\tfrac{1}{2})^5(2) + (\tfrac{1}{2})^5(2) + (\tfrac{1}{2})^6(2)$$

This establishes the pattern for each hit count, and we can write the total sum:

$$
\begin{aligned}
y_3^2 &= [(\tfrac{1}{2})^2 + 2(\tfrac{1}{2})^3 + (\tfrac{1}{2})^4][1 \cdot \frac{1}{2^2} + 2 \cdot \frac{1}{2^4} + 3 \cdot \frac{1}{2^6} + \dots] \\
&= [1 + \tfrac{1}{2} + \tfrac{1}{2} + \tfrac{1}{4}] \sum_{n=1}^{\infty} \frac{n}{2^{2n}} \\
&= \left(\frac{9}{4}\right)\frac{4}{9} \\
y_3^2 &= 1
\end{aligned}
$$

By a symmetry argument, we can see that $y_5^2 = 1$. Further, Figure 6 can convince us that every walk that hits 3 a certain number of times before returning to 2, will hit 6 exactly the same number of times. The same goes for 5 and 4. Last, the expected hits at 1 between visits to 2 is trivially 1, since any path leaving 2 must hit 1 first, and cannot reach 1 from any other vertex than 2.

We have established, then, that $y^2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T$. We said in 0.4.1 that if x^* is the solution to the advection system $L_{adv}x = 0$ and y^* is the solution to the Markov system $\mathcal{A}\mathcal{D}^{-1}y = y$, then we can obtain x^* by solving the Markov chain and computing $x^* = \mathcal{D}^{-1}y^*$. Since $(d_{ii})^{-1} = 1/2$ for the odd vertices (since they have 2 outgoing edges), and 1 for the even vertices, we obtain $\ker(L_{adv}) = c \begin{bmatrix} 1 & 2 & 1 & 2 & 1 & 2 \end{bmatrix}^T$ as desired.

We can construct a similar diagram for $\mathcal{G}_2$ and see that it is less symmetric:

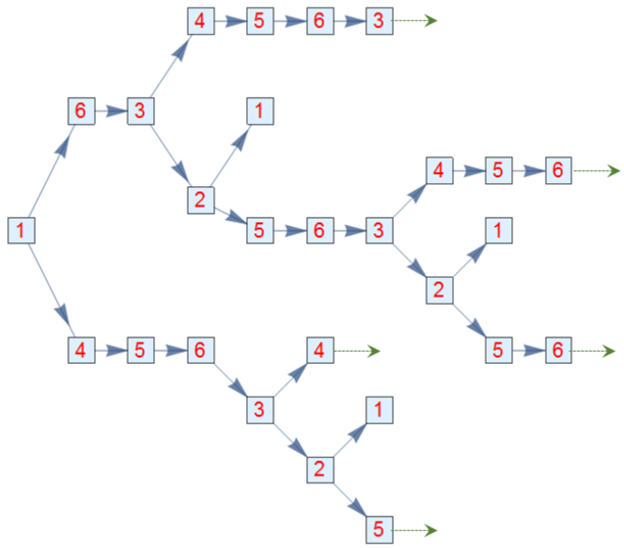


Figure 7: A walk diagram for $\mathcal{G}_2$

We said we computed the advection solution to be $\ker(L_{adv}) = x^* = \begin{bmatrix} 1 & 2 & 4 & 5 & 7 & 8 \end{bmatrix}^T$. So $y^* = \mathcal{D}x^* = \begin{bmatrix} 2\cdot 1 & 2\cdot 2 & 2\cdot 4 & 1\cdot 5 & 1\cdot 7 & 1\cdot 8 \end{bmatrix}^T = \begin{bmatrix} 2 & 4 & 8 & 5 & 7 & 8 \end{bmatrix}^T$ (up to a normalizing constant).

Without getting into as much in-depth calculation as we did for $\mathcal{G}_1$, just by looking at Figure 7, it is clear that every walk that hits 3 a certain number of times before returning to 1, will hit 6 exactly the same number of times. And indeed, we have just computed that $y_3^k = y_6^k$. Further, we can see that the walks that hit 3 n times before returning to 1 outnumber the walks that hit 2 n times before returning to 1, by a factor of 2:1. And indeed, $y_3^k = 2y_2^k$.

0.5.3 Rank of L_{adv} and Strongly Connected Components

Now that we've used Markov chain theory to provide some better intuition on the 1D solutions to advection, we turn our attention to the multi-dimensional case.

Note that a graph can have overlapping weakly connected components. Note also that the number of sinks can be less than the number of strongly connected components. Here is an example of a graph that has two strongly connected components (vertices $\{2, 3, 4\}$ and vertices $\{5, 6\}$), and one sink:

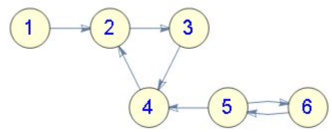


Figure 8: A graph with overlapping weakly connected components.

The subgraph with the vertices $\{1, 2, 3, 4\}$ is a weakly connected component that overlaps with the subgraph with the vertices $\{2, 3, 4, 5, 6\}$, which is another weakly connected component.

Theorem 0.5.3.1. *If $\mathcal{G}$ has c sinks $\mathcal{S}_1, \mathcal{S}_2, \ldots \mathcal{S}_k, \ldots, \mathcal{S}_c$, the rank of L_{adv} is equal to $n - c$, and the kernel of L_{adv} is spanned by the vectors $u_k = \ker(L_{adv}(\mathcal{S}_k))$.*

Proof. A graph with c sinks can have its nodes relabeled into groups α_1 as the vertices of $\mathcal{S}_1$, α_2 as the vertices of $\mathcal{S}_2$, etc., and β as the set of vertices of $\mathcal{G}$ that are in no sink, so that $L_{adv}(\mathcal{G})$ is of the form

$$L_{adv}(\mathcal{G}) = \begin{bmatrix} B_0 & 0 & 0 & \cdots & 0 \\ B_1 & L_{adv}(\mathcal{S}_1) & 0 & \cdots & 0 \\ B_2 & 0 & L_{adv}(\mathcal{S}_2) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ B_c & 0 & 0 & \cdots & L_{adv}(\mathcal{S}_c) \end{bmatrix}$$

where B_0 is the portion of $L_{adv}(\mathcal{G})$ that describes the interconnections between vertices in β, and B_k is the portion of $L_{adv}(\mathcal{G})$ that describes the interconnections between vertices in β and in $\mathcal{S}_k$. Past B_0, the top row (of matrices) is 0 matrices because no node in one of the sinks has an edge leading out into a node in β. (Recall the definition $\mathcal{A}_{i,j} := v_{(j \to i)}$, the weight of the edge going from j to i). Further, all entries above and below $L_{adv}(\mathcal{S}_k)$ are 0 since there is no node in one of the sinks that has an edge leading out into a node in a different sink. Each submatrix $L_{adv}(\mathcal{S}_k)$ is of the same form as it would be as if it were a standalone graph because the advection Laplacian is unchanged locally, since the diagonal only concerns outgoing nodes.

(The trivial case is where the graph is entirely a single sink, which has a one-dimensional kernel as established in 0.5.1.1.)

Now let $u_k = \ker(L_{adv}(\mathcal{S}_k))$. The set w_k^* of vectors that are defined as u_k for the vertices within $\mathcal{S}_k$ and 0 otherwise are a set of c linearly independent vectors in $\ker(L_{adv}(\mathcal{G}))$. There are no other vectors (not a linear combination of these) in the kernel by the following argument:

Since Lemma 0.5.1.1 established that each $L_{adv}(\mathcal{S}_k)$ individually has a kernel of dimension 1, then the w_k^* clearly span the kernel of $L_{adv}(\cup\mathcal{S}_i) = \begin{bmatrix} L_{adv}(\mathcal{S}_2) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & L_{adv}(\mathcal{S}_c) \end{bmatrix}$. So any other vector in the kernel can only have values that are not a linear combination of w_k^* at vertices in β. But this would mean, for some k, that $\begin{bmatrix} B_0 & 0 \\ B_k & L_{adv}(\mathcal{S}_k) \end{bmatrix}$ has two (linearly independent) vectors in its kernel, and as the advection Laplacian for a weakly connected graph, this violates 0.5.2.1. $\square$

0.5.4 Graphs with Multi-dimensional Limiting Behavior

In 0.5.2 we discussed the case where there is a unique limiting solution to the advection system $L_{adv}x = 0$ (the nullspace of L_{adv} is one dimensional), which results when $\mathcal{G}$ has a single sink. The result we just established in the previous section allows us to address the remaining situations, which occur when $\mathcal{G}$ has multiple sinks, and the limiting solution to the advection system is not unique.

We just established that the kernel of L_{adv} is spanned by the vectors $w_k^* = \ker(L_{adv}(\mathcal{S}_k))$ (with zeros on all nodes outside $\mathcal{S}_k$). According to the general scheme we introduced for solving first order linear systems, we can project our initial mass along the 0-eigenvectors, corresponding to the communicating classes, to determine the final distribution of mass in advection. To rephrase what we discussed in 0.2.4, the solution $u(t)$ of a first order linear system is of the form

$u(t) = \sum_i b_i(t) w_i = \sum_i c_i(0) \exp(-\lambda_i t) w_i$

where $b_i(0)$ is the initial condition $u(0)$ projected onto the eigenvector w_i, i.e.

$$b(0) = \begin{bmatrix} | & | & \dots & | \\ w_1 & w_2 & \dots & w_n \\ | & | & \dots & | \end{bmatrix} u(0)$$

The solution will decay to 0 along all eigenvectors except those with 0-eigenvalues. If the c w_k^* span the kernel, we have

$u(t) = \sum_{k=1}^{c} b_k(t) w_k^* = \sum_{k=1}^{c} b_k(0) \exp(-\lambda_i t) w_k^*$

where

$$b(0) = \begin{bmatrix} | & | & \dots & | \\ w_1 & w_2 & \dots & w_c \\ | & | & \dots & | \end{bmatrix} u(0)$$

In fact, if we replaced each sink with a single node, and computed the allocation of mass to that node, we could then recursively solve advection on the sink as its own strongly connected graph using the techniques from the previous section.

0.5.5 Random Graphs

If we consider the behavior of advection on a randomly generated large graph (with randomly oriented edges), it would make sense to expect there to be multiple sinks. Here is an example of the progression of advection on a randomly generated oriented graph with 35 nodes and 100 edges.

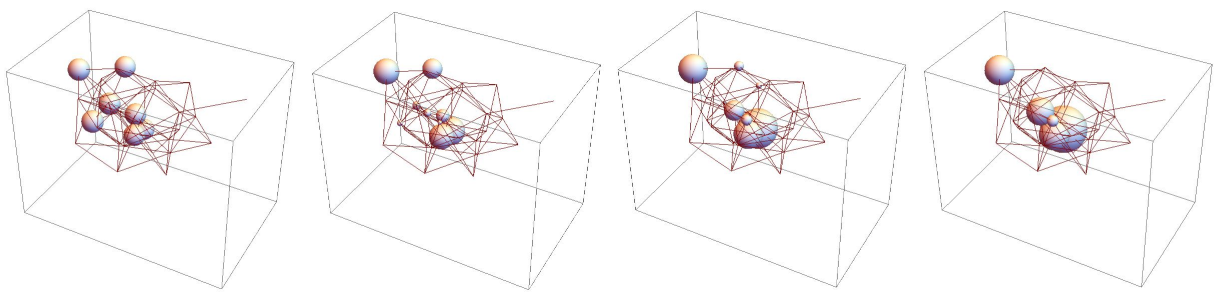


Figure 9: Advection on a random digraph with 35 nodes and 100 edges.

This graph has 5 single-node sinks, which accumulate the mass over time (and form a 5-dimensional kernel for L_{adv}). There is a good amount of room for potential future research exploring the statistics of patterns of sinks in randomly generated graphs.

0.5.6 A Few Comments on Short Term Behavior

By now we have provided a fairly thorough treatment of the qualitative features of the long-term solutions to advection. We'll make a few short comments here on the spectral properties of L_{adv} that affect the short-term behavior of advection.

We established in 0.3.1 that all eigenvalues have positive real part, which is important since it confirms that advection won't explode. This is connected to a larger set of results on the spectra of L_{cons}:

Definition 0.5.6.1. The following properties are equivalent for a non-negative square matrix A termed a *primitive* matrix:

(1) there is a k such that all entries of A^k are positive. [14]

(2) A only has one eigenvalue $r = \rho(A)$ on its spectral circle ([12],pg. 674).

(3) The graph with adjacency matrix A has the property that the set of all cycle lengths has common divisor 1. [6]

Matrices that are not primitive are called *imprimitive*, and a graph with an imprimitive adjacency matrix has the property that the set of all cycle lengths has a common divisor $k > 1$ (the graph is called k-periodic).[6]

Proposition 0.5.6.1. *All eigenvalues of L_{cons} lie in a disk of radius 1 centered at the point $1 + 0j$ in the complex plane. If $A(\mathcal{G})$ is primitive, then all the nonzero eigenvalues of L_{cons} lie on the interior of this disk, and if it is imprimitive, then its eigenvalues are distributed along the border of the disk. ([6])*

The result in 0.3.0.1 connecting the consensus and advection Laplacian of graphs and reverse graphs, $L_{adv}(\mathcal{G}) = (L_{cons}(\mathcal{G}^{\mathcal{T}}))^T$, can be useful in translating these results on the spectra of L_{cons} to the spectra of L_{adv}. For now we will stay content with the result on positive semidefinite-ness, and note one more simple property:

Proposition 0.5.6.2. *If a graph $\mathcal{G}$ has no cycles in it, $L_{adv}(\mathcal{G})$ has entirely real eigenvalues.*

This is because the nodes can be labelled such that $L_{adv}(\mathcal{G})$ is lower triangular, and its diagonal entries are positive real numbers, so the eigenvalues are all real. We can see this in the lack of oscillatory behavior in the grid graph example in Figure 2.

0.6 Covering Graphs

The theory of covering graphs discussed in Gross and Tucker [15] and Sunada [16] has some interesting implications on advection on a graph: given a covering graph and a base graph, the vertices in the fiber $\{a_k\}$ in the covering graph have the same equilibrium values as the base graph had at its vertex a. [4] (Theorem

[4]Conversation with Dennis Tseng helped clarify this idea.

0.6.0.1).

First we'll define a covering graph as according to Sunada:

Let $\mathcal{G} = (V, E)$ and $\mathcal{G}_0 = (V_0, E_0)$ be connected graphs.

Definition 0.6.0.1. [16] A *morphism* from $\mathcal{G}$ to G_0 is a set of a vertex map and an edge map $f = (f_V : V \to V_0, f_E : E \to E_0)$ that satisfies

$\omega(a, (i, j) = \omega(f_V(a), f_E((i, j))$

with ω is as defined in 0.2, which is to say that f_V and f_E preserve the oriented adjacency relations between the vertices.

A covering graph is a special type of morphism. Here is an example of a morphism that fails to be a covering projection, which we define next:

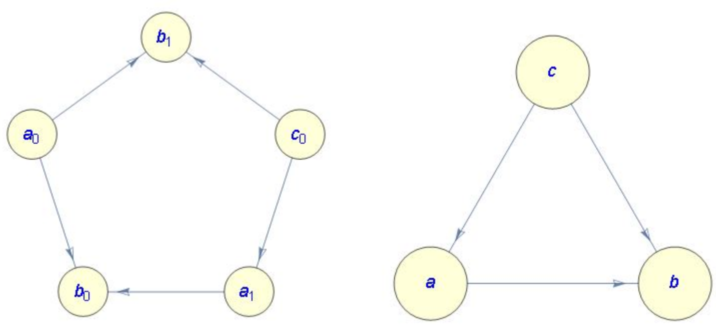


Figure 10: We can define a morphism from $\mathcal{G}$, left, to $\mathcal{G}_0$, right, by sending $a_0, a_1 \to a; b_0, b_1 \to b, c_0 \to c$, and $(a_i, b_i) \to (a, b), (b_i, c_i) \to (b, c), (a_i, c_i) \to (a, c)$. This satisfies the conditions of a morphism, but is not a covering map.

Definition 0.6.0.2 (Covering map)**.** [16] A morphism $f : \mathcal{G} \to \mathcal{G}_0$ is a *covering map* if it preserves local adjacency relations between vertices and edges, that is, it satsifies:

(1) $f : V \to V_0$ is surjective.

(2) $\forall a \in V$ the restriction $f|_{\{e \in E : \omega(a,e) \neq 0\}} : \{e \in E : \omega(a, e) \neq 0\} \to \{e \in E_0 : \omega(f(a), f(e)) \neq 0\}$ is a bijection. In other words, there is an orientation-preserving bijection between the edges attached to a vertex a in the and the edges attached to $f(a)$.

We call $\mathcal{G}$ the *covering graph* over the *base graph* $\mathcal{G}_0$. For a vertex $a \in V_0$, its *fiber* is the set of vertices mapped to it, $f^{-1}(a) = \{b \in V : f(b) = a\}$.

Here is an example of a covering projection.

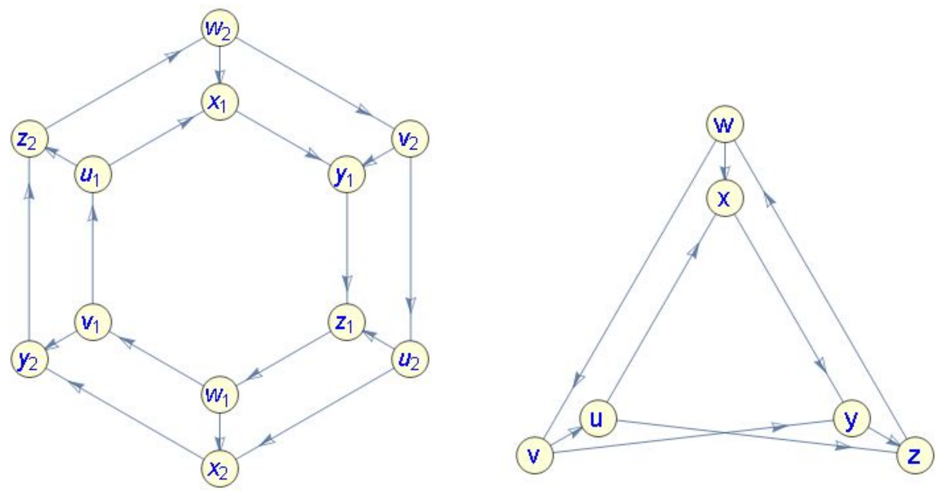


Figure 11: We can define a covering map from $\mathcal{G}$, left, to $\mathcal{G}_0$, right, by sending $a_i, \rightarrow a\ \forall a \in V(\mathcal{G}_0)$, and $(a_i, b_i) \rightarrow (a, b)\ \forall (a, b) \in E(\mathcal{G}_0)$. This satisfies the conditions of a covering map. Example from Gross and Tucker [15], pg. 59.

Proposition 0.6.0.1. *If $\mathcal{G}$ covers $\mathcal{G}_0$ the size of $f^{-1}(a)$ is the same across all $a \in \mathcal{G}_0$:[16]*

First, observe that for an edge $(a, b) \in E_0$, $|f^{-1}((a, b))| = |f^{-1}(a)|$. This follows directly from the bijection requirement in the definition of the covering map: If we pick a vertex $a' \in V : f(a') = a$, and there is a local bijection between the edges E_a and $f(E_a)$, then that particular a' contributes one element to the set $f^{-1}(a)$, and contributes one element to the set $f^{-1}((a, b))$. Adding across all a' we'll get the same number of elements in $\{f^{-1}((a, b))\}$ and $\{f^{-1}(a)\}$.

It follows that for any edge (a, b), since $|f^{-1}(a)| = |f^{-1}((a, b))| = |f^{-1}(b)|$, we have $|f^{-1}(a)| = |f^{-1}(b)|$. As long as the graph is connected, $|f^{-1}(a)|$ must be the same for all a. If it is a finite number k, then f is said to be k-fold.

We will now prove the following:

Theorem 0.6.0.1. *For a graph $\mathcal{G}$ that covers a graph $\mathcal{G}_0$ via a k-fold map f, let $u_0^* = \ker(L_{adv}(\mathcal{G}_0))$ and $u^* = \ker(L_{adv}(\mathcal{G}))$. For any vertex $a \in V_0$, the entry of u^* corresponding to any $a_i \in f^{-1}(a)$ satisfies $u_0^*(a) = ku^*(a_i)$, where the k is independent of the choice of a.*

An equivalent way of stating this is to say if we consider a covering graph as the state space of a Markov chain, the values of the stationary distribution at all a_i in the fiber of a are equal (and they are equal to a specific fraction of the value at a of the stationary distribution of the covering graph).

This result means that covering graphs can be used as a tool for constructing distributed repeating patterns of advection equilibrium on a graph. We can demonstrate this in the mobius strip graph example (Figure 11). $\mathcal{G}_0$ in the example in Figure 11 is the same as the $\mathcal{G}_2$ in Figure 5, and has kernel $\ker(L_{\mathcal{G}_0}) = c\begin{bmatrix}1 & 2 & 4 & 5 & 7 & 8\end{bmatrix}^T$ as we discussed in 0.5.2. Theorem 0.6.0.1 tells us that if we label the vertices of $\mathcal{G}$ accordingly, $\ker(L_{\mathcal{G}}) = c\begin{bmatrix}1 & 2 & 4 & 5 & 7 & 8 & 1 & 2 & 4 & 5 & 7 & 8\end{bmatrix}^T$.

Proof. Name the vertices of $\mathcal{G}_0$ $a_1, a_2, \ldots$, and let vector functions u_0 on the vertices of $\mathcal{G}_0$ be ordered this way, and as well let the entries of $L_{adv}(\mathcal{G}_0)$ (we'll shorten this to $L_{\mathcal{G}_0}$) be constructed with respect that order. If $k = |f^{-1}(a)|$ is the fold of the covering map, and $\{a_n^1, a_n^2, \ldots, a_n^k\} = f^{-1}(a_1)$ is the fiber of a_1 we can choose to order the entries of a vector function u on the vertices of $\mathcal{G}$ according to $\{a_1^1, a_1^2, a_1^3, \ldots, a_1^k, a_2^1, a_2^2, a_2^3, \ldots, a_2^k, a_n^1, a_n^2, a_n^3, \ldots, a_n^k, \ldots\}$, and we can let the entries of $L_{\mathcal{G}}$ be constructed with respect that order.

Let $u_0^* \in \ker[L_{\mathcal{G}_0}]$. Expand u_0^* to a function u^* on the vertices of $\mathcal{G}$ by repeating each entry of u_0^* k times, that is

$$\begin{bmatrix}a_1 & a_2 & \cdots & a_n & \vdots\end{bmatrix}^T \rightarrow$$
$$\begin{bmatrix}a_1 & a_1 & \cdots & a_1 & a_2 & a_2 & \cdots & a_2 & a_3 & \cdots & a_{n-1} & a_n & a_n & \cdots & a_n & a_{n+1} & \cdots\end{bmatrix}^T$$

We claim $L_{\mathcal{G}}u^* = 0$.

Consider the jth entry of $L_{\mathcal{G}}u^*$. It is given by

$$\begin{aligned}
L_{\mathcal{G}}u^*(j) &= \sum_i L_{\mathcal{G}}(j,i)u^*(i) \\
&= \sum_{i \in V : L_{\mathcal{G}}(j,i) \neq 0} L_{\mathcal{G}}(j,i)u^*(i) \\
&= \sum_{i \in V : D_{\mathcal{G}}(j,i) \neq 0} D_{\mathcal{G}}(j,i)u^*(i) - \sum_{i \in V : A_{\mathcal{G}}(j,i) \neq 0} A_{\mathcal{G}}(j,i)u^*(i) \\
&= D_{\mathcal{G}}(j,j)u^*(j) - \sum_{i \in V : A_{\mathcal{G}}(j,i) \neq 0} A_{\mathcal{G}}(j,i)u^*(i)
\end{aligned}$$

By the local bijection property of the covering map,

$$= D_{\mathcal{G}_0}(f(j), f(j))u^*(j) - \sum_{i \in V : A_{\mathcal{G}}(j,i) \neq 0} A_{\mathcal{G}_0}(f(j), f(i))u^*(i)$$

By the construction of u_0^*,

$$= D_{\mathcal{G}_0}(f(j), f(j))u_0^*(f(j)) - \sum_{i \in V : A_{\mathcal{G}}(j,i) \neq 0} A_{\mathcal{G}_0}(f(j), f(i))u_0^*(f(i))$$

$$= \sum_i D_{\mathcal{G}_0}(f(j), f(i))u_0^*(f(i))$$

$$L_{\mathcal{G}} u^*(j) = 0 \quad \square$$

This result has an implication for expected visits as discussed in 0.4.4: if we start at some vertex b in a covering graph $\mathcal{G}$ and make a sojourn back to b, the structure of $\mathcal{G}$ produces the result that the expected number of times we hit some a_m in the fiber of a is precisely the expected number of times we hit any other a_n in the same fiber.

This result also has potential implications for cooperative control. At the end of their paper, Chapman and Mesbahi [1] discuss a few implementations of advection. In their first example, they connect a multi-agent team via a cycle graph, and use the edge weights to control the geometry of the limiting configuration. Specifically, since the graph is a cycle, the equilibrium value at a vertex is inversely proportional to the weight of the edge projecting onto it. Were some application to arise where the connecting graph needed to be unweighted or could not be weighted, the covering graph allows for the construction of a stereotyped pattern in the limiting values.

0.7 Conclusion

The goal of this report was to present advection on graphs as a mathematical object considered more broadly and generally than simply as a construction for cooperative control as it was in [1]. While the limiting solutions of advection are easily computable, viewing advection as a cousin of a Markov chain provides further intuition and insight on the role of cycles in determining patterns in the solutions, as well as on the character of solutions for graphs that are not irreducible, an area that has been largely sidestepped by the cooperative control literature. Further, we hope to have sparked a possibly useful line of inquiry into the employment of covering graphs for advection dynamics, and related constructions such as Markov chains.

This report mostly focused on the limiting solutions of advection, but there is substantial room for further work on characterizing the spectra of advection Laplacians, beginning with translating existing results on the spectra of consensus Laplacians. In particular, there is certainly more to be worked out on the spectral analysis of covering graphs. There is also a good amount of work to be done analyzing advection on large random graphs: (a) in understanding how the sink structure of a graph depends on the probabilistic parameters generating its oriented edges, and (b) characterizing the limiting behavior, as well as the spectra of the advection Laplacian, in the large n limit (as the number of vertices approaches infinity).

Bibliography

[1] Airlie Chapman and Mehran Mesbahi. Advection on graphs. *Proceedings of the IEEE Conference on Decision and Control*, pages 1461–1466, 2011.

[2] Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[3] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[4] Abraham Berman and Robert J Plemmons. Nonnegative matrices. *The Mathematical Sciences, Classics in Applied Mathematics*, 9, 1979.

[5] Leo J Grady and Jonathan Polimeni. *Discrete calculus: Applied analysis on graphs for computational science*. Springer Science & Business Media, 2010.

[6] J Alexander Fax and Richard M Murray. Information flow and cooperative control of vehicle formations. *IEEE transactions on automatic control*, 49(9):1465–1476, 2004.

[7] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[8] Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control*, 49(9):1520–1533, 2004.

[9] RB Bapat. The laplacian matrix of a graph. *Mathematics Student-India*, 65(1):214–223, 1996.

[10] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[11] James R Norris. *Markov chains*. Cambridge university press, 1998.

[12] Carl D Meyer. *Matrix analysis and applied linear algebra*, volume 2. Siam, 2000.

[13] Francesco Bullo, Jorge Cortes, and Sonia Martinez. *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009.

[14] Shlomo Sternberg. *Dynamical systems*. Dover Publications, Inc., Mineola, NY, 2010.

[15] Jonathan L Gross and Thomas W Tucker. *Topological graph theory*. Courier Corporation, 1987.

[16] T Sunada. Topological crystallography with a view towards discrete geometric analysis, surveys and tutorials in the applied mathematical sciences, vol. 6, 2012.