Fiedler Vector Approximation via Interacting Random Walks

VISHWARAJ DOSHI, North Carolina State University, USA
DO YOUNG EUN, North Carolina State University, USA

The Fiedler vector of a graph, namely the eigenvector corresponding to the second smallest eigenvalue of a graph Laplacian matrix, plays an important role in spectral graph theory with applications in problems such as graph bi-partitioning and envelope reduction. Algorithms designed to estimate this quantity usually rely on a priori knowledge of the entire graph, and employ techniques such as graph sparsification and power iterations, which have obvious shortcomings in cases where the graph is unknown, or changing dynamically. In this paper, we develop a framework in which we construct a stochastic process based on a set of interacting random walks on a graph and show that a suitably scaled version of our stochastic process converges to the Fiedler vector for a sufficiently large number of walks. Like other techniques based on exploratory random walks and on-the-fly computations, such as Markov Chain Monte Carlo (MCMC), our algorithm overcomes challenges typically faced by power iteration based approaches. But, unlike any existing random walk based method such as MCMCs where the focus is on the leading eigenvector, our framework with interacting random walks converges to the Fiedler vector (second eigenvector). We also provide numerical results to confirm our theoretical findings on different graphs, and show that our algorithm performs well over a wide range of parameters and the number of random walks. Simulations results over time varying dynamic graphs are also provided to show the efficacy of our random walk based technique in such settings. As an important contribution, we extend our results and show that our framework is applicable for approximating not just the Fiedler vector of graph Laplacians, but also the second eigenvector of any time reversible Markov Chain kernel via interacting random walks. To the best of our knowledge, our attempt to approximate the second eigenvector of any time reversible Markov Chain using random walks is the first of its kind, opening up possibilities to achieving approximations of higher level eigenvectors using random walks on graphs.

CCS Concepts: • **Mathematics of computing → Probabilistic algorithms; Stochastic processes; Approximation algorithms; Spectra of graphs**;

Additional Key Words and Phrases: Fiedler vector, graph partitioning, spectral clustering, reversible Markov chains, interacting particle systems.

ACM Reference Format:
Vishwaraj Doshi and Do Young Eun. 2020. Fiedler Vector Approximation via Interacting Random Walks. 1, 1 (January 2020), 28 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The eigenvector corresponding to the second smallest eigenvalue of a graph Laplacian matrix is usually referred to as the Fiedler vector. Originally introduced by Miroslav Fiedler in his works on algebraic connectivity [19, 20], the Fiedler vector has found applications in areas such as graph partitioning and clustering [2, 14, 16, 37, 38, 41, 42, 46, 54], graph drawing [27], graph colouring [3],

Authors' addresses: Vishwaraj Doshi, Operations Research Graduate Program, North Carolina State University, Raleigh, NC, 27695, USA, vdoshi@ncsu.edu; Do Young Eun, Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, 27695, USA, dyeun@ncsu.edu.

XXXX-XXXX/2020/1-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

envelope reduction [5] and the analysis of proteins [28, 43]. It plays an important role in spectral graph theory [11, 13], providing powerful heuristics by solving relaxations of NP-hard, integer problems on graph bi-partitioning [21, 44, 53].

Over the years, a number of algorithms have been implemented to approximate the Fiedler vector. The most notable ones are techniques based on graph sparsification [48–50], and multi-level/multi-grid techniques [5, 6, 10, 18]. Both of these focus on pruning edges of the graph to obtain a 'sparser' or 'coarser' subgraph. Fiedler vector computation is then performed on these subgraphs to obtain approximations to the Fiedler vector of the original graph. While the pruning can be done via innovative probabilistic rules based on measures such as effective resistance [48], or using greedy, deterministic rules [6], the core computation of the Fiedler vector is still carried out by various kinds of power methods. The Fiedler vector is also directly related to the mixing time[1, 9, 32] via the corresponding second eigenvalue $\lambda_2(\mathbf{Q})$, also known as the spectral gap, and algorithms to approximate this quantity [12, 22, 26] are fundamentally different from those estimating the Fiedler vector. Other techniques include [8], which approximates the Fiedler vector in a distributed fashion over ad-hoc networks, where 'ad-hoc' refers to the nodes having the ability to process information locally and exchange information with neighbors in a synchronous fashion, and 'distributed' refers to each node estimating it's own component of the Fiedler vector using a local version of power iterations; and [51], which uses techniques such as matrix deflation to develop another power method to numerically compute the Fiedler vector using the dominant eigenvector of a slightly smaller matrix.

While deterministic power method based techniques for approximating the Fiedler vector can have their advantages under the relevant settings (entire state space known beforehand, ad hoc networks with computational capability at each node and message passing), they face challenges when the state space may be unknown in the beginning, or the graph can only be explored via edge traversal mechanisms and direct access to any arbitrary node is not available. Moreover, they do not adapt well to dynamic graphs. This can especially be seen in [8] where recurring small changes in the graph topology can break the important mean-preserving property of their power method, which is corrected via a mechanism only at every N^{th} iteration (N being the number of nodes in the graph) of the algorithm. In other words, given a change in graph topology, the algorithm may not correct its trajectory till almost N many steps, making the case worse for larger graphs. Random walk based methods, on the other hand, provide a way to deal with the above challenges by performing *in situ* computations as they explore the graph on-the-fly. As exhibited time and again in the field of Markov Chain Monte Carlo (MCMC) [23, 33, 40], they can be used robustly to estimate target quantities on graphs without really feeling the repercussions of scale, lack of knowledge of the state space or the effect of dynamically changing graph topology. While MCMC techniques, by employing various versions of random walks and via the ergodic theorem, are successful in estimating π (or sampling according to π) - the first/leading/principal eigenvector of the kernel $\mathbf{Q}$, no similar techniques have provided extensions to the second eigenvector of the kernel.

In this paper, we fill this void by developing a framework based on (interacting) random walks to approximate the Fiedler vector of graph Laplacian matrices. We do this by constructing a stochastic process employing multiple *interacting* random walkers and showing that a properly scaled version of this process converges to the Fiedler vector. Specifically, these random walkers traverse an undirected, connected graph $\mathcal{G}$ according to a Continuous Time Markov Chain (CTMC) with kernel given by $\mathbf{Q} = -\mathbf{L}$, where the matrix $\mathbf{L} \triangleq \mathbf{D} - \mathbf{A}$ is the combinatorial Laplacian of $\mathcal{G}$ ($\mathbf{A}$ is the adjacency matrix and $\mathbf{D}$ is the degree matrix). Walkers are divided equally into two groups, that compete with each other over the network. If a walker encounters another one from the other competing group, it causes the walker to relocate to the location of another randomly selected walker of the competing group. Such competitive interactions are mutual, and for a sufficiently large number of walkers, lead to a natural bi-partition of the graph over time. By analyzing a closely related deterministic process (the fluid limit) we show that the relative density of walkers over the graph serves as a good approximation of the Fiedler vector. We then extend our results to show that our method based on interacting random walks applies to other commonly used graph Laplacians, as well as for estimating the second eigenvector of any time reversible Markov chain kernel. While algorithms based on random walks successfully achieve, via ergodic theorems, knowledge about the *first eigenvector* of matrics, our paper takes a step forward and provides an interacting random walk based algorithm to achieve the *second eigenvector* of a class of matrices. To the best of our knowledge, our framework is the first one to do so, and opens up possibilities of estimating higher order eigenvectors by using such interacting random walk based techniques.

In the remainder of the paper, we begin by giving the basic notations and an introduction to the Fiedler vector via its application in graph partitioning in Section 2. The main theoretical results are distributed among Sections 3, 4 and 5. In Section 3, we detail the construction of our stochastic process and via Theorem 3.4 we relate it to a system of ordinary differential equations (ODEs) as its fluid limit. We show that over a finite time horizon, the stochastic process rarely deviates from the solution of the ODE when the number of walkers is sufficiently large. Section 4 is devoted to the stability analysis of the resulting deterministic ODE system, where we show using a Lyapunov function that the Fiedler vector is the only asymptotically stable fixed point of a suitably scaled version of the system, while all others being unstable. In Section 5 we bring together our results from Sections 3 and 5 to formally show that for sufficiently large number of walkers, our stochastic process spends most of its time in the long run around the asymptotically stable Fiedler vector, while never getting stuck around an unstable fixed point. In Section 6, we provide numerical results to support our theoretical findings and also simulations over time varying dynamic graphs to show the robustness of our framework in that setting. In Section 7, we extend all our results to include various graph Laplacians and time reversible Markov chain kernels. Section 8 provides our concluding remarks.

2 PRELIMINARIES

2.1 Basic notations

Let $\mathcal{G}(\mathcal{N},\mathcal{E})$ denote a general, undirected, connected graph, with $\mathcal{N}$ being the set of nodes and $\mathcal{E}$ being the set of edges, represented by pairs (i,j) for $i,j \in \mathcal{N}$. Let the cardinality of $\mathcal{N}$ be given by a natural number N (i.e. $|\mathcal{N}| = N$). The mathematical quantities best capturing all the information of $\mathcal{G}(\mathcal{N},\mathcal{E})$ are the 'adjacency matrix' $\mathbf{A}$[1] defined as $\mathbf{A}_{ij} \triangleq 1,$ if $(i,j) \in \mathcal{E}$ and $0,$ otherwise $\forall i,j \in \mathcal{N}$, and the diagonal 'degree matrix' $\mathbf{D}$ of the graph defined as $\mathbf{D}_{ii} \triangleq \sum_{j\in\mathcal{N}} A_{ij} \ \ \forall i \in \mathcal{N}$. We call $\mathbf{D}_{ii}$ the 'degree' of node $i \in \mathcal{N}$, and alternatively represent it as $d(i) \triangleq \mathbf{D}_{ii}$.

Since vectors and matrices will be used throughout the paper, we standardize their notation. Lower case, bold faced letters will be used to represent vectors (e.g. $\mathbf{v} \in \mathbb{R}^N$), while upper case, bold faced letters will be used to represent matrices (e.g. $\mathbf{M} \in \mathbb{R}^{N\times N}$), unless clarified otherwise. The i^{th} (ij^{th}) entry of vector $\mathbf{v}$ (matrix $\mathbf{M}$) will be denoted by v_i or $[\mathbf{v}]_i$ (M_{ij} or $[\mathbf{M}]_{ij}$), depending on the situation. We let $\mathbf{D}_{\mathbf{v}} := \text{diag}(\mathbf{v})$ represent the diagonal matrix with $[\mathbf{D}_{\mathbf{v}}]_{ii} = v_i$. Also denote by $\mathbf{1} = [1,1,\cdots,1]^T$ and $\mathbf{0} = [0,0,\cdots,0]^T$, the N-dimensional vectors of all ones and zeros

[1] In the case of a weighted graph, we replace $\mathbf{A}$ by the weighted adjacency matrix $\mathbf{W}$, where the ij^{th} entries represent the weights assigned to each edge. All other equations remain the same. However, we will safely exclude any further, separate mention of weighted graphs because the scope of our results is broad enough to cover not just weighted graphs, but also similar quantities such as kernels of time reversible Markov chains, as we shall observe later in Section 7.

respectively, and $\mathbf{e}_k$ for all $k \in \{1, \cdots, N\}$ represent the canonical basis vectors in R^N, which take value 1 at their k^{th} entry, 0 at every other entry.

The use of '$\mathbf{Q}$' will be reserved exclusively for representing transition rate matrices of continuous time Markov chains (CTMCs). '$\mathbf{P}(\cdot)$' will be used to denote the probability measure, while simple '$\mathbf{P}$' will be used exclusively for transition probability matrices of discrete time Markov chains (DTMCs). Note that throughout the paper, we shall often refer to these matrices using the umbrella term *kernel* of Markov chains. Vectors such as $\mathbf{x}(t)$ and $\mathbf{y}(t)$, with t being time, will be used to denote (deterministic) solutions to systems of ordinary differential equations (ODEs). However, if indexed by a parameter, for example $\mathbf{x}^n(t)$ and $\mathbf{y}^n(t)$, they will be used to denote stochastic processes (also parameterized by n). This distinction between notations of similar deterministic and stochastic quantities shall be reiterated when we define such quantities later on.

Finally, we let $\|\cdot\|$ denote the Euclidean norm for any Euclidean space (i.e. with the appropriate dimensions implicitly understood), and use $\tilde{\cdot}$ to denote normalized versions of vectors, or sets containing normalized vectors.

2.2 Graph Laplacians

Consider an undirected, connected graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$. The quantity of interest throughout the paper will be the *Combinatorial Laplacian*, which is defined as

$$\mathbf{L} \triangleq \mathbf{D} - \mathbf{A}.$$

$\mathbf{L}$ is a symmetric (due to the graph being undirected), positive-semidefinite matrix with non negative eigenvalues. Since the graph is connected, $\mathbf{A}$, and as a result $\mathbf{L}$, are irreducible matrices. Thus, the Perron Frobenius (PF) [25, 35] theorem applies and the smallest eigenvalue 0 (with eigenvector being $\mathbf{1}$) has multiplicity 1. We denote by $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_N$, the spectrum of $\mathbf{L}$, with $\mathbf{1} = \mathbf{v}_1,\ \mathbf{v}_2, \cdots,\ \mathbf{v}_N$ being the corresponding eigenvectors.

Closely related to the combinatorial Laplacian is the *normalized or symmetric Laplacian*, which goes by

$$\mathcal{L} \triangleq \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}.$$

Also a symmetric positive semi-definite matrix, it's smallest eigenvalue is 0 (with corresponding eigenvalue being $\mathbf{D}^{1/2}\mathbf{1}$). Denote by $0 = \bar{\lambda}_1 < \bar{\lambda}_2 \leq \cdots \leq \bar{\lambda}_N$, the spectrum of $\mathbf{L}$, with $\mathbf{D}^{1/2}\mathbf{1} = \bar{\mathbf{v}}_1,\ \bar{\mathbf{v}}_2, \cdots,\ \bar{\mathbf{v}}_N$ being the corresponding eigenvectors.

The third graph Laplacian we introduce is the *random walk Laplacian*, denoted by $\mathbf{L}^{rw}$. We shall define it using the combinatorial and symmetric Laplacians as

$$\mathbf{L}^{rw} \triangleq \mathbf{D}^{-1}\mathbf{L} = \mathbf{D}^{-1/2}\mathcal{L}\mathbf{D}^{1/2} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}.$$

It gets its name because $\mathbf{P} = \mathbf{D}^{-1}A$ is a stochastic matrix (all the rows add up to one) which defines a simple random walk in discrete time on the graph. $\mathbf{L}^{rw}$ is not a symmetric matrix. However, note that due to the similarity transformation (second equality), it shares the same spectrum as $\mathcal{L}$with its left eigenvectors given by $\mathbf{v}_i^{rw} = \mathbf{D}^{1/2}\bar{\mathbf{v}}_i$ where $\bar{\mathbf{v}}_i$ is an eigenvector of $\mathcal{L}$. The PF eigenvector of $\mathbf{L}^{rw}$ is therefore given by $\mathbf{v}_1^{rw} = \mathbf{D}\mathbf{1}$, the vector with its entries being the degree of the respective node.

2.3 Graph bi-partitioning and Fiedler vector

The second eigenvalue of $\mathbf{L}$, i.e. λ_2, is known as the algebraic connectivity of a graph, and the corresponding eigenvector is commonly referred to as the *Fiedler vector*. We shall however use the term *Fiedler vector* to refer to second eigenvectors of any graph Laplacian matrix. To convey the relevance of the Fiedler vector, we look at the problem of bi-partitioning a graph, whose objective is to partition a connected graph into two connected subgraphs in the most 'natural' way possible.

This is often interpreted as partitioning the graph into two subgraphs S and S^c with the fewest number of edges interlinking them, but also not having meaningless solutions such as partitioning the graph at leaf nodes. This train of thought has evolved into the study of problems such as the *Ratio-cut* problem (*RCut*) [21, 53] and the *Normalized-cut* problem (*NCut*) [44, 53], among others. Define $\text{Cut}(S) \triangleq (1/2) \sum_{i \in S} \sum_{j \in S^c} A_{ij}$, for any $S \subset \mathcal{N}$. The *RCut* problem for graph bi-partitioning is given by

$$\text{RCut}(\mathcal{G}) = \min_{S \subset \mathcal{N}} \text{RCut}(S) = \min_{S \subset \mathcal{N}} \left(\frac{\text{Cut}(S)}{|S|} + \frac{\text{Cut}(S^c)}{|S^c|} \right), \tag{1}$$

Unfortunately, minimizing the *RCut* over all subsets of $\mathcal{N}$ is an NP hard problem. [21] showed that by enlarging the integer valued domain into the real valued domain, approximate solutions can be found efficiently. For the *RCut* problem, this real valued relaxation is given by

$$\text{RCut}(\mathcal{G}) \approx \min_{\mathbf{f} \perp \mathbf{1},\ \mathbf{f} \neq 0} \frac{\mathbf{f}^T \mathbf{L} \mathbf{f}}{\mathbf{f}^T \mathbf{f}},$$

the solution to which is well known, and is the Fiedler vector $\mathbf{v}_2$ of $\mathbf{L}$. Likewise, a relaxation of the similar *NCut*[2] problem is solved by $\bar{\mathbf{v}}_2$, the Fiedler vector of the normalized Laplacian $\mathcal{L}$ (or equivalently, the Fiedler vector of $\mathbf{L}^{rw}$, since they share the same signs for the entries, leading to the same partition). Partitioning according to the signed entries of these second eigenvectors therefore solves natural relaxations to well-defined but difficult to solve bi-partitioning problems, and these observations have paved the way for spectral clustering[11, 13–15, 21, 37, 41, 44, 45, 54] as a powerful data analysis tool.

3 A MULTI-WALK, INTERACTING STOCHASTIC PROCESS WITH A DETERMINISTIC LIMIT

We aim to construct a random walk based process which, in the long run, can approximate the Fiedler vector (second eigenvector of $\mathbf{L} = \mathbf{D} - \mathbf{A}$). Our process consists of multiple random walks interacting with each other in a prescribed manner. This can also be written as a density dependent process (with parameter 'n' proportional to the number of random walks) that has a deterministic fluid limit. In this section we provide construction of our stochastic process, and the result on its convergence to a deterministic process (its fluid limit).

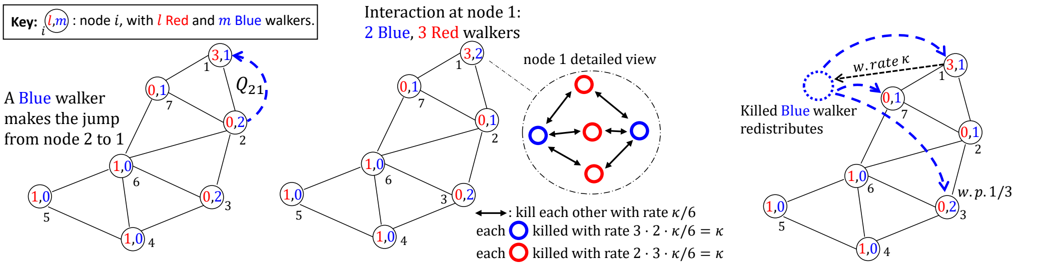
3.1 The interacting stochastic process

Let $\mathcal{G}(\mathcal{N}, \mathcal{E})$ be any undirected, connected graph, with $N = |\mathcal{N}|$ as before. For any $n \in \mathbb{N}$, consider $2n$-many random walkers traversing $\mathcal{G}(\mathcal{N}, \mathcal{E})$ according to a CTMC generated by the kernel $\mathbf{Q} = -\mathbf{L} = \mathbf{A} - \mathbf{D}$. These $2n$ walkers are split into two groups, of n walkers each, by labeling them as either 'type-x' or 'type-y'. Define the stochastic process $\big(X(t), Y(t)\big)_{t \geq 0}$, where $X(t) \in \mathbb{N}_0^N$ ($Y(t) \in \mathbb{N}_0^N$) is a vector such that $X(t)_i$ ($Y(t)_i$) indicates the number of type-x (type-y) walkers present at node $i \in \mathcal{N}$ at time $t \geq 0$. Since there are n walkers of each type, we have $\sum_{i \in \mathcal{N}} X(t)_i = \sum_{i \in \mathcal{N}} Y(t)_i = n$ for all $t \geq 0$. The state space is given by

$$\Gamma_n \triangleq \left\{ (X, Y) \in \mathbb{N}_0^N \times \mathbb{N}_0^N \;\middle|\; \sum_{i \in \mathcal{N}} X_i = \sum_{i \in \mathcal{N}} Y_i = n \right\}. \tag{2}$$

Ever so often, a type-x walker traversing the graph may find itself at a node with type-y walkers present. Such an event is what we call an 'interaction'. When an interaction occurs, every type-y walker present 'kills' each type-x walker present with rate κ/n, for some scalar $\kappa \in (0, +\infty)$. In

[2] The *NCut* problem is similar to the *RCut* problem with the terms $|S|$ and $|S^c|$ in the denominator replaced by $\text{Vol}(S)$ and $\text{Vol}(S^c)$, where $\text{Vol}(S) \triangleq \sum_{i \in S} d(i)$.


Key: (l,m) i : node i, with l Red and m Blue walkers.
A Blue walker makes the jump from node 2 to 1
Q_{21}
Interaction at node 1:
2 Blue, 3 Red walkers
node 1 detailed view
: kill each other with rate $\kappa/6$
each ○ killed with rate $3 \cdot 2 \cdot \kappa/6 = \kappa$
each ○ killed with rate $2 \cdot 3 \cdot \kappa/6 = \kappa$
$w.rate\ \kappa$
Killed Blue walker redistributes
$w.p. 1/3$
3,1
0,1
0,2
1,0
1,0
0,2
1,0
3,2
0,1
0,1
1,0
1,0
0,2
1,0
3,1
0,1
0,1
1,0
1,0
0,2
1,0
1
2
3
4
5
6
7

(a) Walk according to ergodic CTMC $\mathbf{Q}$ (b)Killed with rate $\left(\frac{\kappa}{n}\right) X(t)_i Y(t)_i$ at node i (c)Killed $\longrightarrow$ Redistribute to node $j, w.p. X(t)_j/n$

Fig. 1. The interaction and redistribution mechanism.

return, every type-x walker present 'kills' each type-y walker present with rate κ/n. Another way to describe this interaction is in a pairwise manner. Given any possible 'pair' of one type-x and one type-y walker present at node j, they both kill each other with rate κ/n. At every node $j \in \mathcal{N}$ there are $X(t)_j Y(t)_j$ such pairs at any time $t \geq 0$. Thus, overall, type-x or type-y walkers die with rate $\frac{\kappa}{n} X(t)_j Y(t)_j$. Upon being killed, the 'dead' walker relocates to the position of another randomly selected walker of the same type. At time $t > 0$, this is equivalent to saying that a killed type-x walker will redistribute to some node $i \in \mathcal{N}$ with probability $X^n(t)_i/n$, which is proportional to the number of type-x walkers present at node i. Similarly, a killed type-y walker will redistribute to node $i \in \mathcal{N}$ with probability $Y^n(t)_i/n$.

Consider Figure 1 as an example of an interaction event. We consider $n = 6$. The colour blue is used for type-x walkers, while red is used for type-y walkers. At each node, the number of *blue* and *red* walkers present is given by the respectively coloured numeric entry. Figure 1(a) shows a *blue* walker moving from its initial position at node 2, to node 1, with rate Q_{21}. Figure 1(b) shows a snapshot of the system at a time right after the event in Figure 1(a). Node 1 is now the site of interactions between 2 *blue* and 3 *red* walkers. A detailed view of the interactions at node 1 is shown, where an arrow represents a pair of walkers killing each other with rate κ/n. Let $r \in \mathbb{N}$ count the number of arrows connected to any particular walker. Then, that walker is killed by the group of the other type walkers with the total rate $\frac{\kappa}{n} \cdot r$. Following this logic, each *blue* walker in Figure 1(b) is killed with rate $\frac{\kappa}{6} \cdot 3 = \kappa/2$. Since there are two *blue* walkers present, one of them is killed with rate $\frac{\kappa}{2} \cdot 2 = \kappa$. Similarly, each *red* walker is killed individually with rate $\kappa/3$, and since there are three *red* walkers, one of them is killed with rate κ. Therefore the rate with which a *blue* walker dies is the same at which a *red* walker dies at node 1. Finally, Figure 1(c) shows an event in which a *blue* walker is killed at node 1. Upon death, it instantaneously relocates to the position of another *blue* walker chosen uniformly at random. That is, it randomly redistributes to node i with probability $X(t)_j/n$. A similar redistribution would occur if a *red* walker was to die instead of a *blue* walker, in which case it would redistribute to node j with probability $Y(t)_j/n$.

Events of the process $\left(X(t), Y(t)\right)_{t \geq 0}$ involve type-x or type-y walkers going from a node j to another node i. This corresponds to jumps of size $(\mathbf{e}_i - \mathbf{e}_j, 0)$ for type-x and $(0, \mathbf{e}_i - \mathbf{e}_j)$ for type-y walkers. At any state $(X, Y) \in \Gamma_n$, jumps can be caused by walking from $j \longrightarrow i$ according to the base CTMC kernel $\mathbf{Q}$, which happens with rate $Q_{ji} X_j$ for type-x walkers and $Q_{ji} Y_j$ for type-y walkers. The jumps can also occur as a result of being killed at node j and redistributed to node i. As mentioned earlier, one of the type-x walkers at node j is killed with rate $\frac{\kappa}{n} X(t)_j Y(t)_j$, upon which it redistributes to i with probability $\frac{X_i}{n}$. Therefore, the jump $j \longrightarrow i$ occurs with the overall rate $\frac{\kappa}{n} X(t)_j Y(t)_j \frac{X_i}{n}$ for type-x walkers. Similarly, type-y are killed at j and redistributed to i with overall rate $\frac{\kappa}{n} X(t)_j Y(t)_j \frac{Y_i}{n}$.

Hereafter, for any state $(X,Y)\in\Gamma_n$, we use $\mathbf{x}\in\mathbb{R}^N$ to denote the density distribution of type-x walkers over the graph, with $x_i \triangleq \frac{X_i}{N}$. Similarly we use $\mathbf{y}\in\mathbb{R}^N$ to denote the density distribution of type-y walkers over the graph, with $y_i \triangleq \frac{Y_i}{N}$. Let $\bar{\mathbf{Q}}^n_{x:j\to i}(X,Y)$ denote the total rate with which jumps of type $(\mathbf{e}_i-\mathbf{e}_j,0)$ occur, and similarly let $\bar{\mathbf{Q}}^n_{y:j\to i}(X,Y)$ denote the total rate with which jumps of size $(0,\mathbf{e}_i-\mathbf{e}_j)$ occur. From all of the above, these two quantities can be written as

$$\bar{Q}^n_{x:j\to i}(X,Y) = Q_{ji}X_j + (\frac{\kappa}{n}Y_jX_j)x_i = n\big(Q_{ji}x_j + (\kappa y_jx_j)x_i\big) \tag{3}$$

$$\bar{Q}^n_{y:j\to i}(X,Y) = Q_{ji}Y_j + (\frac{\kappa}{n}X_jY_j)y_i = n\big(Q_{ji}y_j + (\kappa x_jy_j)y_i\big). \tag{4}$$

We can think of $\bar{Q}^n_{x:j\to i}$ and $\bar{Q}^n_{y:j\to i}$ for all $i,j\in\mathcal{N}, i\neq j$ as the off-diagonal entries of a $2N\times 2N$ dimensional matrix $\bar{\mathbf{Q}}(X,Y)$. This matrix is then the transition rate matrix of the CTMC $\big(X(t),Y(t)\big)_{t\geq 0}$ on a finite state space Γ_n as in (2), and we have now fully characterized our stochastic process.

3.2 A closely related deterministic system

Define the density dependent version of $\big(X(t)/n, Y(t)/n\big)_{t\geq 0}$ as $\big(\mathbf{x}^n(t),\mathbf{y}^n(t)\big)_{t\geq 0} \triangleq \big(X(t)/n, Y(t)/n\big)_{t\geq 0}$. Its state space Θ_n is a version of Γ_n where each entry is scaled by $\frac{1}{n}$, or more precisely,

$$\Theta_n = \Sigma_n \times \Sigma_n, \tag{5}$$

where $\Sigma_n = \Sigma \cap \{\frac{1}{n}\mathbf{u} \mid \mathbf{u}\in\mathbb{Z}^N\}$, with $\Sigma \triangleq \{\mathbf{u}\in\mathbb{R}^N \mid \mathbf{u}^T\mathbf{1}=1, \mathbf{u}\geq 0\}$ and $\mathbb{Z}^N$ being the N-dimensional grid. Jumps of size $\frac{1}{n}(\mathbf{e}_i-\mathbf{e}_j,0)$ and $\frac{1}{n}(0,\mathbf{e}_i-\mathbf{e}_j)$ for the process $\big(\mathbf{x}^n(t),\mathbf{y}^n(t)\big)_{t\geq 0}$ occur with rates (3),(4). Using this information, we can define a vector field $F:\Theta_n\longrightarrow\mathbb{R}^{2N}$. For any $(\mathbf{x},\mathbf{y})\in\Theta_n$, $F(\mathbf{x},\mathbf{y})$ captures the average change in $(\mathbf{x},\mathbf{y})\in\Theta_n$ per unit time, and is written as

$$F(\mathbf{x},\mathbf{y}) = \big(F_x(\mathbf{x},\mathbf{y}), F_y(\mathbf{x},\mathbf{y})\big), \tag{6}$$

where

$$F_x(\mathbf{x},\mathbf{y}) \triangleq \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N},j\neq i}\frac{(\mathbf{e}_i-\mathbf{e}_j)}{n}\bar{Q}^n_{x:j,i}(\mathbf{x},\mathbf{y}) \tag{7}$$

$$F_y(\mathbf{x},\mathbf{y}) \triangleq \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N},j\neq i}\frac{(\mathbf{e}_i-\mathbf{e}_j)}{n}\bar{Q}^n_{y:j,i}(\mathbf{x},\mathbf{y}). \tag{8}$$

The following result helps us write the above two equations in a more compact form.

Proposition 3.1. *(Proof in Appendix A.1.) For any* $(\mathbf{x},\mathbf{y})\in\Theta_n$, *we have*

$$F_x(\mathbf{x},\mathbf{y}) = \mathbf{Q}^T\mathbf{x} + [\kappa\mathbf{x}^T\mathbf{y}]\mathbf{x} - \kappa\mathbf{D}_{\mathbf{y}}\mathbf{x} \tag{9}$$

$$F_y(\mathbf{x},\mathbf{y}) = \mathbf{Q}^T\mathbf{y} + [\kappa\mathbf{x}^T\mathbf{y}]\mathbf{y} - \kappa\mathbf{D}_{\mathbf{x}}\mathbf{y} \tag{10}$$

Proposition 3.1 allows us to consider the following system of ordinary differential equations (ODEs):

$$\begin{aligned}\frac{d}{dt}\mathbf{x}(t) &= F_x\big(\mathbf{x}(t),\mathbf{y}(t)\big)\\ \frac{d}{dt}\mathbf{y}(t) &= F_y\big(\mathbf{x}(t),\mathbf{y}(t)\big)\\ \mathbf{x}(0)^T\mathbf{1} &= \mathbf{y}(0)^T\mathbf{1} = 1,\end{aligned} \tag{11}$$

where we let $\big(\mathbf{x}(t),\mathbf{y}(t)\big)_{t\geq 0}$ denote solutions to (11) for $t \geq 0$ (this is sometimes referred to as the semi-flow of F). Note the distinction between the notations $\big(\mathbf{x}^n(t),\mathbf{y}^n(t)\big)$ and $\big(\mathbf{x}(t),\mathbf{y}(t)\big)$, where the former is a stochastic process (parameterized by n), while the latter is the deterministic solution to an ODE system. For the next result, let $\|\cdot\|$ denote the Euclidean norm in $\mathbb{R}^{2N}$, and S be the subset of $\mathbb{R}^{2N}$ given by

$$S \triangleq \{(\mathbf{x},\mathbf{y}) \in \mathbb{R}^{2N} \mid \mathbf{x}^T\mathbf{1} = \mathbf{y}^T\mathbf{1} = 1\}. \tag{12}$$

Proposition 3.2. *(Proof in Appendix A.2.) $F : S \longrightarrow \mathbb{R}^{2N}$ is Lipschitz continuous with Lipschitz constant $M < \infty$, and for every initial point $\big(\mathbf{x}(0),\mathbf{y}(0)\big) \in S$, there exists a unique solution to the system* (11).

The set S is an invariant set for the ODE system (11). Indeed, observe that for any $(\mathbf{x},\mathbf{y} \in S)$ we have $d(\mathbf{1}^T\mathbf{x})/dt = \mathbf{1}^T F(\mathbf{x},\mathbf{y}) = \mathbf{1}^T\mathbf{Q}^T\mathbf{x} + \kappa\mathbf{x}^T\mathbf{y} - \kappa\mathbf{y}^T\mathbf{x} = 0$ since $\mathbf{Q1} = 0$. Similarly, $d(\mathbf{1}^T\mathbf{y})/dt = 0$, and any solution starting in S stays in S. We will only consider such solutions in our analysis of the ODE. To ensure that this is the case, we rewrite (11) as an equivalent system given by

$$\begin{aligned} \frac{d}{dt}\mathbf{x}(t) &= \mathbf{Q}^T\mathbf{x}(t) + \Lambda_x(t)\mathbf{x}(t) - \kappa\mathbf{D}_{\mathbf{y}(t)}\mathbf{x}(t) \\ \frac{d}{dt}\mathbf{y}(t) &= \mathbf{Q}^T\mathbf{y}(t) + \Lambda_y(t)\mathbf{y}(t) - \kappa\mathbf{D}_{\mathbf{x}(t)}\mathbf{y}(t) \\ \mathbf{x}(t)^T\mathbf{1} &= \mathbf{y}(t)^T\mathbf{1} = 1 \ \big(\forall t \geq 0\big), \end{aligned} \tag{13}$$

where $\Lambda_x(t)$ and $\Lambda_y(t)$ are real valued scalars. By summing up all the entries of $\frac{d}{dt}\mathbf{x}(t)$ and $\frac{d}{dt}\mathbf{x}(t)$ and substituting $\mathbf{x}(t)^T\mathbf{1} = \mathbf{y}(t)^T\mathbf{1} = 1$, it can be easily seen that $\Lambda_x(t) = \Lambda_y(t) = \kappa\mathbf{x}(t)^T\mathbf{y}(t)$. Thus, we retrieve the original system and hence the equivalence. Moving forwards, we will use $\Lambda(t) \triangleq \Lambda_x(t) = \Lambda_y(t) = \kappa\mathbf{x}(t)^T\mathbf{y}(t)$ in our equations. Before proceeding with our first important result, we make an observation about fixed points $(\mathbf{x}^*,\mathbf{y}^*)$ of (13).

Remark 3.3. *Any fixed point $(\mathbf{x}^*,\mathbf{y}^*) \in S$ of* (13) *has all strictly positive entries. In other words, there exists no $i \in \mathcal{N}$ such that $x_i^* = 0$ or $y_i^* = 0$.*

Proof. (Remark 3.3) Consider $(\mathbf{x}^*,\mathbf{y}^*) \in S$ to be fixed points of (13). Then, the i^{th} entry of $\mathbf{x}^*$ satisfies the equation

$$0 = \sum_{j\in\mathcal{N}} Q_{ji}x_j^* + [\kappa{\mathbf{x}^*}^T\mathbf{y}^*]x_i - \kappa x_i^* y_i^*.$$

Suppose $x_i^* = 0$. Then the above equation becomes $0 = \sum_{j\in\mathcal{N},\ j\neq i} Q_{ji}x_j^*$. This means that for any $j \in \mathcal{N}$ such that $Q_{ji} > 0$ (implying that node j is a neighbor of node i), the corresponding entry $x_j^* = 0$. This in turn leads to $x_k^* = 0$, for all neighbors k of node j. Since our graph is connected, there is always a path connecting node i to any other node of the graph, implying $\mathbf{x} = \mathbf{0}$. Similarly, if $y_i^* = 0$, following the same steps as before gives us $\mathbf{y}^* = \mathbf{0}$. This is in violation of the third equation in (13), giving us a contradiction. This completes the proof. □

We are now ready to state the main result connecting the stochastic process from Section 3.1 and the deterministic ODE system from Section 3.2.

3.3 From stochastic to deterministic dynamics

In this section we show that the stochastic process from Section 3.1, indexed by n, almost surely converges to the deterministic ODE system from Section 3.2. Before we state the theorem, we make the following assumption.

A1: For any $n \in \mathbb{N}$, $\big(\mathbf{x}(0),\mathbf{y}(0)\big) = \big(\mathbf{x}^n(0),\mathbf{y}^n(0)\big)$.

Theorem 3.4. *(Proof in Appendix A.3.) Consider the family of stochastic processes (indexed by n)* $\{(\mathbf{x}^n(t), \mathbf{y}^n(t))_{t\geq 0}\}_{n\in\mathbb{N}}$, *with kernels* $(\bar{\mathbf{Q}}^n)_{n\in\mathbb{N}}$ *defined as in* (3) *and* (4). *Let* $(\mathbf{x}(t), \mathbf{y}(t))$ *be solutions to the ODE system* (11) *that satisfy (A1). Then, for all* $T \in (0, \infty)$, *we have*

$$\lim_{n\to\infty} \sup_{0\leq t\leq T} \|(\mathbf{x}^n(t), \mathbf{y}^n(t)) - (\mathbf{x}(t), \mathbf{y}(t))\| = 0 \quad a.s. \tag{14}$$

More precisely, for any $\epsilon > 0$ *and* $T \in (0, \infty)$, *we have*

$$\begin{aligned} &\mathbf{P}\Big(\sup_{0\leq t\leq T} \big\|(\mathbf{x}^n(t), \mathbf{y}^n(t)) - (\mathbf{x}(t), \mathbf{y}(t))\big\| \geq \epsilon\Big) \\ &\leq 4N(N-1)\exp\left(-n(1+\kappa)T \cdot h\Big(\frac{\epsilon e^{-MT}}{\sqrt{2}N(N-1)(1+\kappa)T}\Big)\right), \end{aligned} \tag{15}$$

where $h(x) \triangleq (1+x)\log(1+x) - x$, *M is the Lipschitz constant from Proposition 3.2, and* $N = |\mathcal{N}|$ *is the size of the graph.*

(15) gives an upper bound on the probability that, uniformly over a finite time horizon $[0, T]$, the stochastic process indexed by $n \in \mathbb{N}$ deviates from the solution to the deterministic process by at least $\epsilon > 0$, provided they start at the same initial point. A discussion on the effect of parameters n, κ and N on the bound (15) and the long run behavior (as $T \longrightarrow \infty$) is deferred to the end of Section 5.

4 ANALYSIS OF THE DETERMINISTIC FLOW AND CONVERGENCE TO FV

From Theorem 3.4, the solutions $(\mathbf{x}(t), \mathbf{y}(t))_{t\geq 0}$ of the ODE system (13) serve as a deterministic approximation for the CTMC $(\mathbf{x}^n(t), \mathbf{y}^n(t))_{t\geq 0}$ for larger values of $n \in \mathbb{N}$. Then, it makes sense to analyze the trajectories of the ODE system for its convergence properties and the nature of its fixed points. For the rest of the paper, we use the notation $\Lambda(t) = \Lambda_x(t) = \Lambda_y(t)$ mentioned earlier.

4.1 Fixed points of the system

We first state a consequence of the Courant-Fischer min-max theorem [25, 35] (specific to our case), which we shall refer to later in this Section.

Lemma 4.1. *Let* $\tilde{S} \triangleq \{\mathbf{w} \in \mathbb{R}^N \mid \mathbf{w}^T\mathbf{1} = 0, \|\mathbf{w}\| = 1\}$. *Given a CTMC kernel* $\mathbf{Q} = -\mathbf{L}$, *where* $\mathbf{L}$ *is a Laplacian matrix with eigenvalues ordered as* $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_N$, *we have*

$$\lambda_2 = \min_{\mathbf{u}\in\tilde{S}} \mathbf{u}^T[-\mathbf{Q}]^T\mathbf{u}, \tag{16}$$

with $\tilde{\mathbf{v}}_2 = \mathbf{v}_2/\|\mathbf{v}_2\|$ *being the minimizer, and*

$$\lambda_k = \max_{\mathbf{u}\in\tilde{S},\ \mathbf{u}\perp\{\mathbf{v}_{k+1},\cdots,\mathbf{v}_N\}} \mathbf{u}^T[-\mathbf{Q}]^T\mathbf{u}, \tag{17}$$

with $\tilde{\mathbf{v}}_k = \mathbf{v}_k/\|\mathbf{v}_k\|$ *being the maximizer.*

Let $\mathbf{z}(t) \triangleq \mathbf{x}(t) - \mathbf{y}(t)$ for all $t \geq 0$, and consider an (implicit) system of equations obtained by subtracting the second equation of (13) from the first one, which then reads as

$$\begin{aligned} \frac{d}{dt}\mathbf{z}(t) &= \mathbf{Q}^T\mathbf{z}(t) + \Lambda(t)\mathbf{z}(t) \\ \mathbf{z}(t)^T\mathbf{1} &= 0 \ (\forall t \geq 0). \end{aligned} \tag{18}$$

Let Ω denote the set of fixed points of (18). Any fixed point $\mathbf{z}^* \in \Omega$ satisfies

$$-\mathbf{Q}^T\mathbf{z}^* = \mathbf{L}^T\mathbf{z}^* = \Lambda^*\mathbf{z}^* \tag{19}$$

$$\mathbf{z}^{*T}\mathbf{1} = 0 \tag{20}$$

for some $\Lambda^* \in (0, \infty)$, and could therefore be a (left) eigenvector of $\mathbf{L}$ (up to a scalar multiple), or the *zero* vector (which we shall often refer to as the *origin*). Since, for any $t \geq 0$, $\mathbf{z}(t) = \mathbf{0}$ if and only if $\mathbf{x}(t) = \mathbf{y}(t)$, the fixed point $\mathbf{0} \in \Omega$ of (18) corresponds to the invariant set S_0 of (13), defined as

$$S_0 \triangleq \{(\mathbf{x}, \mathbf{y}) \in S \mid \mathbf{x} = \mathbf{y},\ \mathbf{x}^T = \mathbf{y}^T = 1\}. \tag{21}$$

To exclude, from our analysis, the case where trajectories of (18) might hit *zero* after some finite time, or equivalently trajectories of (13) might enter set S_0 and stay there for all future times, we show the following result.

Proposition 4.2. *For all sufficiently large κ, the invariant set S_0 defined in* (21) *is an unstable set for the system* (13).

Proof. Recall the form of the Jacobian of $F : \mathbb{R}^{2N} \to \mathbb{R}^{2N}$ from Appendix A.2. When evaluated at $\mathbf{x} = \mathbf{y}$ (which we shall use here in place of $\mathbf{x}^*$ and $\mathbf{y}^*$ to denote fixed points), we can write it as

$$\mathbf{J}_F(\mathbf{x}, \mathbf{x}) = \begin{bmatrix} \mathbf{Q} & 0 \\ 0 & \mathbf{Q} \end{bmatrix} + \kappa \begin{bmatrix} \mathbf{x}\mathbf{x}^T + \mathbf{D}_\mathbf{x}\mathbf{D}_\mathbf{x} - \mathbf{D}_\mathbf{x} & \mathbf{x}\mathbf{x}^T - \mathbf{D}_\mathbf{x} \\ \mathbf{x}\mathbf{x}^T - \mathbf{D}_\mathbf{x} & \mathbf{x}\mathbf{x}^T + \mathbf{D}_\mathbf{x}\mathbf{D}_\mathbf{x} - \mathbf{D}_\mathbf{x} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{Q}^T & 0 \\ 0 & \mathbf{Q}^T \end{bmatrix} + \kappa \begin{bmatrix} \mathbf{D}_\mathbf{x}^2 & 0 \\ 0 & \mathbf{D}_\mathbf{x}^2 \end{bmatrix} + \kappa \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \otimes \left[\mathbf{x}\mathbf{x}^T - \mathbf{D}_\mathbf{x}\right],$$

Here, $\mathbf{D} \otimes \mathbf{E}$ denotes the Kronecker product of two square matrices $\mathbf{D}$ and $\mathbf{E}$. To simplify further analysis, we write down the above matrix as $\mathbf{J}_F(\mathbf{x}, \mathbf{x}) = \mathbf{A} + \mathbf{B} + \mathbf{C}$, where

$$\mathbf{A} = \begin{bmatrix} \mathbf{Q}^T & 0 \\ 0 & \mathbf{Q}^T \end{bmatrix}, \quad \mathbf{B} = \kappa \begin{bmatrix} \mathbf{D}_\mathbf{x}^2 & 0 \\ 0 & \mathbf{D}_\mathbf{x}^2 \end{bmatrix}$$

$$\mathbf{C} = \kappa \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \otimes \left[\mathbf{x}\mathbf{x}^T - \mathbf{D}_\mathbf{x}\right]$$

For any symmetric matrix $\mathbf{M} \in \mathbb{R}^{N \times N}$, let $\lambda_1(\mathbf{M}) \leq \lambda_2(\mathbf{M}) \leq \cdots \leq \lambda_N(\mathbf{M})$ denote the ordering of its eigenvalues. Observe that the eigenvalues of $\mathbf{A}$ are the same as those of $\mathbf{Q}$ with double the multiplicity, implying $\lambda_{2N}(\mathbf{A}) = 0$, and $\lambda_{2k}(\mathbf{A}) = \lambda_k(\mathbf{Q})$ for all $k \in \{1, \cdots, N\}$. The eigenvalues of $\mathbf{B}$ are the diagonal elements. Thus, $\lambda_1(B) = \kappa \min_{i \in \mathcal{N}} x_i^2$ and $\lambda_{2N}(B) = \kappa \max_{i \in \mathcal{N}} x_i^2$.

For matrix $\mathbf{C}$, observe that $\mathbf{x}\mathbf{x}^T - \mathbf{D}_\mathbf{x}$ has a *zero* row sum, with negative diagonal entries and non-negative off-diagonal entries. It therefore defines a CTMC transition rate matrix, and we have $\lambda_{2N}(\mathbf{x}\mathbf{x}^T - \mathbf{D}_\mathbf{x}) = 0$, with the other eigenvalues being strictly negative. It can also be checked that the matrix of all ones on the left of the Kronecker product has the spectrum $\{2, 0\}$. The eigenvalues of the Kronecker product are equal to eigenvalues of the two involved matrices cross multiplied. We therefore obtain $2N$ eigenvalues of $\mathbf{C}$, with $\lambda_{2N}(\mathbf{C}) = \cdots = \lambda_{N-1}(\mathbf{C}) = 0$, and the others being strictly negative.

The Weyl's inequality[25, 35] for real symmetric matrices $\mathbf{D}, \mathbf{E} \in \mathbb{R}^{N \times N}$ is given, for any $k \in \{1, \cdots, N\}$, as

$$\lambda_1(\mathbf{D}) + \lambda_k(\mathbf{E}) \leq \lambda_k(\mathbf{D} + \mathbf{E}) \leq \lambda_N(\mathbf{D}) + \lambda_k(\mathbf{E})$$

Applying the lower bound of Weyl's inequality on $\lambda_{N-1}(\mathbf{A} + \mathbf{B} + \mathbf{C})$, we get

$$\lambda_1(\mathbf{A}) + \lambda_{N-1}(\mathbf{B} + \mathbf{C}) \leq \lambda_{N-1}(\mathbf{A} + \mathbf{B} + \mathbf{C}).$$

Applying the lower bound of Weyl's inequality once again, this time on $\lambda_{N-1}(\mathbf{B} + \mathbf{C})$, we get

$$\lambda_1(\mathbf{A}) + \lambda_1(\mathbf{B}) + \lambda_{N-1}(\mathbf{C}) \leq \lambda_{N-1}(\mathbf{A} + \mathbf{B} + \mathbf{C}).$$

Now since $\lambda_1(\mathbf{A}) = \lambda_1(\mathbf{Q}) = -\lambda_N(\mathbf{L})$, $\lambda_{N-1}(\mathbf{C}) = 0$ and $\lambda_1(\mathbf{B}) = \kappa \min_{i \in \mathcal{N}} x_i^2$, we get

$$-\lambda_N(\mathbf{L}) + \kappa \min_{i \in \mathcal{N}} x_i^2 \leq \lambda_{N-1}(\mathbf{J}_F(\mathbf{x}, \mathbf{x})).$$

As a consequence of Remark 3.3 in Section 3.1, $x_i > 0$ for every $i \in \mathcal{N}$. This means the $N + 1$ largest eigenvalues of the Jacobian become positive for all sufficiently large $\kappa > 0$, and each fixed point in S_0 has an unstable space[3] of minimum dimension $N + 1$ associated with the linearized system at the point [39]. Therefore the set S_0 is linearly unstable in S. This concludes the proof. □

The above result is equivalent to saying that the fixed point $\mathbf{0} \in \Omega$ of (18) is unstable. This allows us to consider trajectories starting from $\big(\mathbf{x}(0), \mathbf{y}(0)\big) \in S \setminus S_0$, or equivalently, from $\mathbf{z}(0) \neq 0$, which will be useful in Section 4.2.

4.2 Convergence to the Fiedler vector

From Section 4.1, we know that the set Ω of fixed points of (18) contains any scalar multiples $c\mathbf{v}_k$ of the (left) eigenvectors $\mathbf{v}_k$ of $\mathbf{L}$ for $k \geq 2$. This includes $\mathbf{0}$, which we showed to be an unstable fixed point in Proposition 4.2. The next question is about the convergence properties of the ODE system, and we have the following.

Theorem 4.3. *Trajectories of the system* (18) *always converge to a fixed point. Furthermore, the Fiedler vector* $\mathbf{v}_2$ *is an asymptotically stable fixed point of the ODE system (18), with all others being unstable.*

The rest of this section is devoted to the proof of Theorem 4.3. First, we provide a 'Lyapunov function' $V : \mathbb{R}^N \longrightarrow [0, +\infty)$ with strictly negative slope along trajectories of (13), at any point which is not in Ω (i.e. not a fixed point). This will subsequently be shown in Proposition 4.4. We then put together our proof of Theorem 4.3, where we show convergence to a fixed point using the LaSalle invariance principle (Theorem B.2 in Appendix B). This will prove the first statement of Theorem 4.3. To prove the second statement, we use the Lyapunov stability theory to show that all the fixed points of type $c\mathbf{v}_k$ ($k \geq 3, c \neq 0$) are unstable, and the only possible candidate left for convergence, i.e., $c\mathbf{v}_2$, is asymptotically stable. For the sake of completeness, we provide relevant definitions and results from the theory surrounding LaSalle invariance principle and Lyapunov stability in Appendix B.

Recall that the set S_0 is unstable and we only consider trajectories starting from $S \setminus S_0$. For such trajectories, since $\mathbf{z}(t) \neq \mathbf{0}$ in $S \setminus S_0$, we can construct another (implicit) system whose solutions are

$$\tilde{\mathbf{z}}(t) \triangleq \frac{\mathbf{z}(t)}{\|\mathbf{z}(t)\|} \text{ for all } t \geq 0. \tag{22}$$

Trajectories of (22) are always contained in the set $\tilde{S} = \{\mathbf{w} \in \mathbb{R}^N \mid \mathbf{w}^T \mathbf{1} = 0, \|\mathbf{w}\| = 1\}$. Non-zero fixed points of (18) will also be fixed points of this new system. However, this time, they are normalized, and isolated in space, unique up to only a sign. This will make it easier to apply the LaSalle principle later to show convergence. Since we are now working in the normalized space, we let $\{\tilde{\mathbf{v}}_2, \cdots, \tilde{\mathbf{v}}_N\}$ denote the normalized eigenvectors of $\mathbf{L}$, and denote the set of fixed points as $\tilde{\Omega} = \{\tilde{\mathbf{v}}_2, \cdots, \tilde{\mathbf{v}}_N\}$.[4]

[3] An unstable space of a point is the set where trajectories move away from the point. For individual fixed points, it is usually enough to show that this unstable space is at least 1-dimensional to guarantee instability. This can be done by showing that one of the eigenvalues of the Jacobian evaluated at the point is positive (for linear instability)[39]. However when showing instability of all points associated to an invariant, k-dimensional set given by K, we need to show that the unstable space of these points is of dimension at least $k + 1$, in order to rule out the unstable spaces made of vectors pointing simply inside set K. Showing the existence of an unstable space of dimension $k + 1$ hence shows instability of the set K itself.

[4] Technically, it should be $\tilde{\Omega} = \{\pm\tilde{\mathbf{v}}_2, \cdots, \pm\tilde{\mathbf{v}}_N\}$, but we use $\tilde{\mathbf{v}}_k$ to refer to $\pm\tilde{\mathbf{v}}_k$ irrespective of the sign.

Define the Lyapunov function $V : \mathbb{R}^N \longrightarrow [0,\infty)$ as

$$V(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T[-\mathbf{Q}]^T\mathbf{u} = \frac{1}{2}\mathbf{u}^T\mathbf{L}\mathbf{u}, \quad \text{for any } \mathbf{u} \in \mathbb{R}^N. \tag{23}$$

We shall now analyze the function $V(\mathbf{u})$ over trajectories $\tilde{\mathbf{z}}(t)$ as defined in (22).

Proposition 4.4. *For trajectories of (13) starting from any* $(\mathbf{x}(0), \mathbf{y}(0)) \in S\backslash S_0$, *we have* $\frac{d}{dt}V(\tilde{\mathbf{z}}(t)) \leq 0$, *with equality only at the fixed points* $\tilde{\Omega}$ *that are left eigenvectors of* $\mathbf{L}$.

Proof. For the rest of this section, we suppress the '(t)' notation and assume it implicitly. We also use $\dot{f}$ to mean $\frac{d}{dt}f$ whenever convenient. Observe that

$$V(\tilde{\mathbf{z}}(t)) = \frac{1}{2}\tilde{\mathbf{z}}^T[-\mathbf{Q}]^T\tilde{\mathbf{z}} = \frac{1}{2}\frac{\mathbf{z}^T[-\mathbf{Q}]^T\mathbf{z}}{\mathbf{z}^T\mathbf{z}}.$$

Differentiating V along the trajectories of (18) gives

$$\frac{d}{dt}V(\tilde{\mathbf{z}}(t)) = \frac{1}{2(\mathbf{z}^T\mathbf{z})^2}\Big([\mathbf{z}^T\mathbf{Q}^T\mathbf{z}]\big[\frac{d}{dt}\mathbf{z}^T\mathbf{z}\big] - [\mathbf{z}^T\mathbf{z}]\big[\frac{d}{dt}\mathbf{z}^T\mathbf{Q}^T\mathbf{z}\big]\Big). \tag{24}$$

First, we have

$$\frac{d}{dt}\Big(\mathbf{z}^T\mathbf{z}\Big) = 2\mathbf{z}^T\dot{\mathbf{z}}(t) = 2\big(\mathbf{z}^T\mathbf{Q}^T\mathbf{z} + \Lambda(t)\mathbf{z}^T\mathbf{z}\big), \tag{25}$$

and similarly we have

$$\frac{d}{dt}\Big(\mathbf{z}^T[-\mathbf{L}]\mathbf{z}\Big) = \dot{\mathbf{z}}^T\mathbf{Q}^T\mathbf{z} + \mathbf{z}^T\mathbf{Q}^T\dot{\mathbf{z}} = 2\big(\Lambda(t)\mathbf{z}^T\mathbf{Q}^T\mathbf{z} + \mathbf{z}^T[\mathbf{Q}^2]^T\mathbf{z}\big). \tag{26}$$

By substituting (25) and (26) into (24), we get

$$\frac{d}{dt}V(\tilde{\mathbf{z}}(t)) = \frac{\big(\mathbf{z}\mathbf{Q}^T\mathbf{z}\big)^2 - \big(\mathbf{z}^T[\mathbf{Q}^2]^T\mathbf{z}\big)(\mathbf{z}^T\mathbf{z})}{(\mathbf{z}^T\mathbf{z})^2} \tag{27}$$

To show (27) is non-positive, we leverage the property of $\mathbf{Q} = -\mathbf{L}$ being a symmetric matrix whose eigenvectors $\{\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \cdots \tilde{\mathbf{v}}_N\}$ form an orthonormal basis for $\mathbb{R}^N$. Thus, any vector $\mathbf{z} \in \mathbb{R}^N$ can be written as a linear combination of these orthonormal eigenvectors. Since the trajectory $\mathbf{z}$ always satisfies $\mathbf{z}^T\mathbf{v}_1 = \mathbf{z}^T\mathbf{1} = 0$, we can write $\mathbf{z} = \sum_{k=2}^N c_k\tilde{\mathbf{v}}_k$, where $c_k = \mathbf{z}^T\tilde{\mathbf{v}}_k$ and similarly,

$$\mathbf{z}^T\mathbf{z} = \sum_{k=2}^N c_k^2, \ \mathbf{z}^T[-\mathbf{Q}^T]\mathbf{z} = \sum_{k=2}^N \lambda_k c_k^2, \ \mathbf{z}^T[\mathbf{Q}^2]^T\mathbf{z} = \sum_{k=2}^N \lambda_k^2 c_k^2. \tag{28}$$

Substituting these into (27) yields

$$\frac{d}{dt}V(\tilde{\mathbf{z}}(t)) = \left[\sum_{k=2}^N \lambda_k \frac{c_k(t)^2}{\Big(\sum_{k=2}^N c_k(t)^2\Big)}\right]^2 - \left[\sum_{k=2}^N \lambda_k^2 \frac{c_k(t)^2}{\Big(\sum_{k=2}^N c_k(t)^2\Big)}\right]. \tag{29}$$

Now, for each $t > 0$, define a discrete random variable $R(t)$ which takes values λ_k with probability $\frac{c_k(t)^2}{\sum_{k=2}^N c_k(t)^2}$ for $k \in \{2, 3, \cdots, N\}$. We can then rewrite (29) as

$$\frac{d}{dt}V(\tilde{\mathbf{z}}(t)) = E[R(t)]^2 - E[R(t)^2] = -\text{Var}[R(t)] \leq 0, \tag{30}$$

where the equality $\frac{d}{dt}V(\mathbf{z}(t)) = 0$ holds when the random variable $R(t)$ is constant, i.e., when only one of the $c_k(t)$'s is non-zero. In other words, we have zero derivative only when $\tilde{\mathbf{z}}(t)$ hits an eigenvector $\tilde{\mathbf{v}}_k$ for some $k = 2, 3, \cdots, N$. This completes the proof. □

Using the above results and Theorem B.4 from Appendix B, which is regarding the stability of fixed points, we are now ready to prove Theorem 4.3.

Proof. (Theorem 4.3) First, we use the LaSalle invariance principle (Theorem B.2) to prove convergence to *a* fixed point. Note that any trajectory originating from $\tilde{S} = \{\mathbf{w} \in \mathbb{R}^N \mid \mathbf{w}^T\mathbf{1} = 0, \|\mathbf{w}\| = 1\}$ (and as a result never leaving $\tilde{S}$) is relatively compact. This is because $\tilde{S}$ is a closed and bounded subset of a finite dimensional Euclidean space $\mathbb{R}^N$, and hence compact. Also, the fixed points $\{\tilde{\mathbf{v}}_2, \tilde{\mathbf{v}}_3, \cdots, \tilde{\mathbf{v}}_N\}$ of $\left(\tilde{\mathbf{z}}(t)\right)_{t\geq 0}$ are isolated in $\tilde{S}$. Then, Theorem B.2 implies that the solutions given by (22) have all their trajectories starting from $\tilde{\mathbf{z}}(0) \in \tilde{S}$ converge to *a* fixed point (i.e. no limit cycles exist and convergence to a point is guaranteed).

Now, we use Proposition B.4(i) to show that the Fiedler vector $\tilde{\mathbf{v}}_2$ is an asymptotically stable fixed point. We already know from (16) in Lemma 4.1 that $\tilde{\mathbf{v}}_2$ is the global minimizer of the Lyapunov function V, and by Proposition B.4(i), it is asymptotically stable.

To see why the eigenvectors $\tilde{\mathbf{v}}_k \in \tilde{\Omega} \setminus \{\tilde{\mathbf{v}}_2\}$ are unstable, we show that they can never be local minima. Indeed, for any $\alpha \in (0, 1)$ and $k > 2$, let $\mathbf{p} = \tilde{\mathbf{v}}_k + \alpha(\tilde{\mathbf{v}}_2 - \tilde{\mathbf{v}}_k) = (1 - \alpha)\tilde{\mathbf{v}}_k + \alpha\tilde{\mathbf{v}}_2$. Then, we have again as a consequence of (17) in Lemma 4.1 that $V(\mathbf{p}) < V(\tilde{\mathbf{v}}_k)$. Moreover, if $\tilde{\mathbf{z}}(0) = \mathbf{p}$ we have $\dot{V}(\tilde{z}(0)) < 0$ by Proposition 4.4. Therefore, for $k > 2$, $\tilde{\mathbf{v}}_k$ is definitely not a local minimizer, and is thus unstable in view of Proposition B.4(ii).

In summary, we have shown that the Fiedler vector is the only asymptotically stable fixed point of (18), with all other fixed points (eigenvectors) being unstable. This completes the proof. □

5 FROM DETERMINISTIC BACK TO STOCHASTIC DYNAMICS

In this section, we take the opportunity to reflect upon the results in Sections 3 and 4, and their influence on the long run behaviour of the stochastic process constructed in Section 3.1. Theorem 3.4 made the first connection between stochastic processes $(\mathbf{x}^n(t), \mathbf{y}^n(t))$ for $n \in \mathbb{N}$, and the solution $(\mathbf{x}(t), \mathbf{y}(t))$ of the deterministic ODE system (11) (or, consequently, between $\mathbf{z}^n(t) \triangleq \mathbf{x}^n(t) - \mathbf{y}^n(t)$ and $\mathbf{z}(t)$ in (18)) by showing that uniformly over a finite time horizon (and sufficiently large n) the stochastic process rarely deviates from the solution of the ODE system. This prompted us to analyze system (18) and in Theorem 4.3, we gave stability results on all the fixed points of (18). The Fiedler vector $\mathbf{v}_2$ (up to a constant multiple) turned out to be an asymptotically stable fixed point, with all the higher eigenvectors $\mathbf{v}_k$, $k \geq 3$ (up to constant multiples) being unstable. This gives us reason to believe that a working algorithm simulating the interacting stochastic process $(\mathbf{x}^n(t), \mathbf{y}^n(t))$ should have $\mathbf{z}^n(t)$ converge to $c\mathbf{v}_2$ in some sense, provided it never gets stuck around other, unstable fixed points forever. In this section, we make this intuition precise by resorting to results in [7], Sections 3.2 and 3.3.

For any set $B \subset \mathbb{R}^N$ and given $\mathbf{z}^n(0) \in B$ (stochastic process starts in B), let

$$T^n(B) = \inf\{t \geq 0 \mid \mathbf{z}^n(t) \notin B\}$$

define the *exit time* from set B. Similarly, let

$$H^n(B, T) = \frac{1}{T}\int_0^T \mathbb{1}_{\{\mathbf{z}^n(s)\in B\}} ds$$

denote the (random) fraction of time the stochastic process $\mathbf{z}^n(t)$ spends inside set B in the interval $[0, T]$. Note that $\mathbf{z}^n(t)$ takes values on the N-dimensional grid $(\mathbb{Z}^N)$ scaled by the factor $1/n$. Denote this scaled grid by $\frac{1}{n}\mathbb{Z}^N$. Then, $\mathbf{z}^n(t) \in B$ for some $B \in R^N$ and $t \geq 0$ is possible only if $B \cap \frac{1}{n}\mathbb{Z}^N \neq \varnothing$. Using this, define for every $k \geq 2$ the set $B(\mathbf{v}_k, m)$ as the smallest open ball containing $c\mathbf{v}_k$, but not the *origin*, such that $B(\mathbf{v}_k, m) \cap \frac{1}{m}\mathbb{Z}^N \neq \varnothing$. Note that as m increases, $B(\mathbf{v}_k, m)$ gets smaller and smaller for each $k \geq 0$ (since resolution of the grid $\frac{1}{m}\mathbb{Z}^N$ gets finer), until the sets do not intersect anymore, and $\lim_{m\to\infty} B(\mathbf{v}_k, m) = \{c\mathbf{v}_k, c \in \mathbb{R}\}$[5]. Therefore for large values of m, the sets $B(\mathbf{v}_k, m)$ for $k \geq 2$ are disjoint and never contain two or more different eigenvectors as fixed points, thereby isolating them. Also, for any set $B \subset \mathbb{R}^N$ containing $\mathbf{v}_k$, $k \geq 3$, let $\bar{B}$ represent all points in B such that starting from those points, the ODE trajectories $\mathbf{z}(t)$ leave B in finite time (i.e. subset of the unstable space of $\mathbf{v}_k$ which is contained in B). Then, we can prove the following by adapting Propositions 3 and 4 in [7] to our system.

Proposition 5.1. *Consider m sufficiently large such that $B(\mathbf{v}_k, m)$ for all $k \geq 2$ are disjoint. Then,*

(i) *for any $k \geq 3$ and sufficiently large n, the stochastic process originating from $\mathbf{z}^n(0) \in \bar{B}(\mathbf{v}_k, m)$ leaves the neighborhood $B(\mathbf{v}_k, m)$ of $\mathbf{v}_k$ in finite time with high probability. More precisely,*

$$\mathbf{P}\left(\limsup_{n\to\infty} T^n(B(\mathbf{v}_k, m)) < +\infty\right) = 1.$$

(ii) *for n sufficiently large and any $B(\mathbf{v}_2, m)$ where $m \leq n$, with high probability the process $\mathbf{z}^n(t)$ spends almost all its time, in the long run, in the neighborhood $B(\mathbf{v}_2, m)$. More precisely, for every $m \in \mathbb{N}$,we have*

$$\lim_{n\to\infty} [\liminf_{T\to\infty} H^n(B(\mathbf{v}_2, m), T)] = 1 \quad a.s.$$

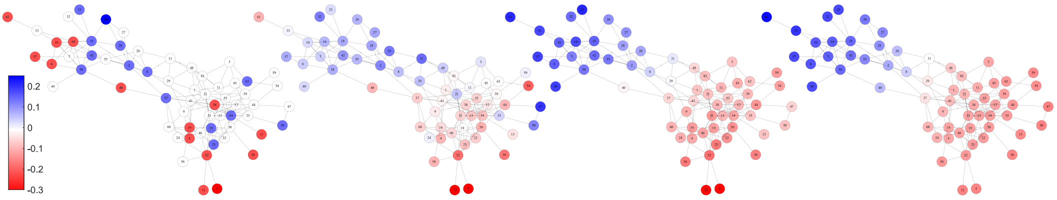
The above result states that for all sufficiently large n, the stochastic process spends only finite time in a neighborhood of the unstable fixed points $\mathbf{v}_k$, $k \geq 3$. It then proceeds to spend almost all its time, in the long run, in a neighborhood of the asymptotically stable fixed point $\mathbf{v}_2$. The second statement of Proposition 5.1 is especially helpful to us. Since it implies that for all sufficiently large n, the stochastic process spends most of its time in a small neighborhood of $\mathbf{v}_2$ with high probability, we expect to observe that $\frac{1}{T}\int_0^T \mathbf{z}^n(s)ds \approx \mathbf{v}_2$ via simulations, for large time horizon T. In the next section, we provide numerical results to show that this is indeed the case for interacting random walks over different values of n.

Lastly, recall the bound (15) in Theorem 3.4. (15) only gives information on how closely the stochastic process follows the solution to the ODE system, and is not by any means a measure of how quickly $\mathbf{z}^n(t)$ might converge to $\mathbf{v}_2$. That being said, the bound in (15) increases with κ and N, but decreases in n. Therefore, theoretically, increases in κ and N need to be compensated by increasing n if we want (15) to ensure that the stochastic process $\left(\mathbf{x}^n(t), \mathbf{y}^n(t)\right)_{t\geq 0}$ closely follows the solution $\left(\mathbf{x}(t), \mathbf{y}(t)\right)_{t\geq 0}$ of the ODE system (11). However in practice our framework turns out to be forgiving and requires only moderately large n to work well with a wide range of N and κ, as we shall observe in the next section.

6 NUMERICAL RESULTS

We begin by providing our simulation setup in Section 6.1. In Section 6.2 we present simulation results over a range of parameters and different graphs in order to confirm our main theoretical results from Sections 3, 4 and 5. In Section 6.3 we consider the setting of dynamic graphs, and present numerical results that show the robustness of the performance of framework over a dynamic topology.

[5] The above can also be done for the normalized version $\tilde{\mathbf{z}}(t)$, in which case the fixed points are $\tilde{\mathbf{v}}_k$ and truly isolated in space, making construction of the isolating simple. However, we construct $B(\mathbf{v}_k, m)$ the way we do to emphasize that for larger n, $B(\mathbf{v}_k, m)$ can be taken to be tighter around $c\mathbf{v}_k$ for larger m.



(a) Normalized $\hat{\mathbf{z}}^n(t)\,;n=15,t=0$ (b) $\hat{\mathbf{z}}^n(t)\,;n=15,t=10$ (c) $\hat{\mathbf{z}}^n(t)\,;n=15,t=50$ (d) Normalized FV $(\tilde{\mathbf{v}}_2)$

Fig. 2. Colormap showing progression of the algorithm over a short time. Red and blue colors denote which group of walkers has majority at the corresponding node, while intensity of the color depends on the value of the corresponding FV estimate. (d) shows partition using the actual Fiedler vector.

6.1 Simulation setup

We source two real world datasets, one is about a social network of dolphins (62 nodes, 159 edges) [34] and the other is about the Facebook social network (4039 nodes, 88234 edges) from the SNAP repository [31]. They are undirected, connected graphs, each having its own combinatorial Laplacian matrix $\mathbf{L}$. Walking according to $\mathbf{Q}=-\mathbf{L}$ on any graph is very natural, since it requires only local information.[6] For a specifically chosen n, we keep track of the quantities $\hat{\mathbf{x}}^n(t) \triangleq \frac{1}{t}\int_0^t \mathbf{x}^n(s)ds$ and $\hat{\mathbf{y}}^n(t) \triangleq \frac{1}{t}\int_0^t \mathbf{y}^n(s)ds$, the empirical density distributions of type-x and type-y walkers; and our Fiedler vector estimator $\hat{\mathbf{z}}^n(t) \triangleq \hat{\mathbf{x}}^n(t) - \hat{\mathbf{y}}^n(t)$ for all $t \geq 0$.

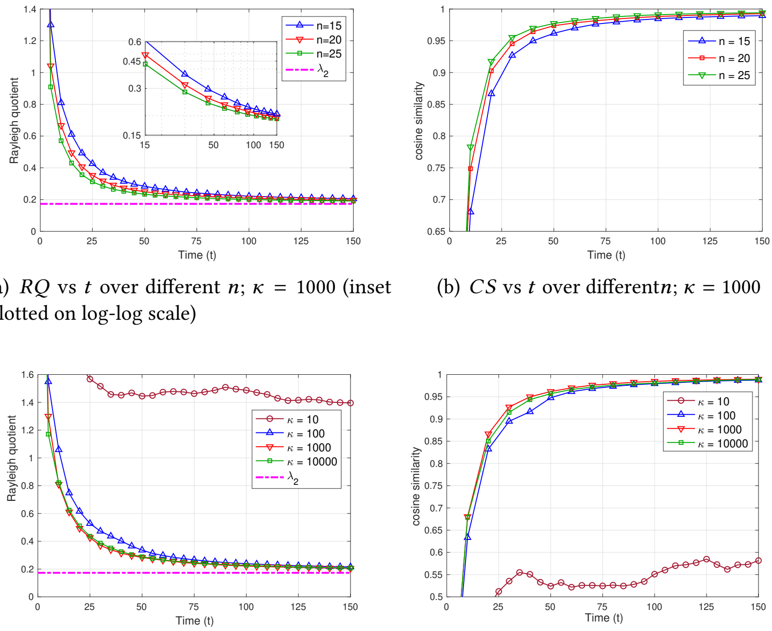
To show how our framework approximates the Fiedler vector and leads to a natural bi-partition of the graph, we illustrate in Figure 2 the progression of a single simulation run over the *Dolphins* graph, with $n=15$ walkers in each group and $\kappa=1000$ kept constant throughout the run. Figures 2(a), (b) and are 'snapshots' taken at times $t=0$ (initial configuration), 10 and 50 respectively. The color of a node represents which group of walkers hold majority at that node, and its intensity depends on the corresponding (normalized) value of $\hat{\mathbf{z}}^n(t)$. Figure 2(d) is an illustration of what the spectral bi-partitioning using the true value (computed offline) of $\mathbf{v}_2$ would be. As the simulation progresses, we can observe the bi-partition taking shape and Figure 2(c) and (d) end up closely resembling each other.

We wish to make these observations from Figure 2 more concrete, and show quantitatively that the quantity $\hat{\mathbf{z}}^n(t)$ converges to the Fiedler vector $\mathbf{v}_2$ over time. For this purpose we use two metrics to help visualize the convergence, namely the Rayleigh quotient (RQ) and the cosine similarity (CS), which are given respectively by

$$RQ(\hat{\mathbf{z}}^n(t)) \triangleq \frac{\hat{\mathbf{z}}^n(t)^T\mathbf{L}\hat{\mathbf{z}}^n(t)}{\|\hat{\mathbf{z}}^n(t)\|^2}, \qquad CS(\hat{\mathbf{z}}^n(t)) \triangleq \frac{|\hat{\mathbf{z}}^n(t)^T\mathbf{v}_2|}{\|\hat{\mathbf{z}}^n(t)\|\cdot\|\mathbf{v}_2\|}.$$

From Lemma 4.1, $RQ(\cdot)$ achieves its minimum, λ_2 at the Fiedler vector $\mathbf{v}_2$. Therefore $RQ(\hat{\mathbf{z}}^n(t))$ approaching λ_2 signifies that $\hat{\mathbf{z}}^n(t)$ aproaches $\mathbf{v}_2$. The cosine similarity $CS(\hat{\mathbf{z}}^n(t))$ tracks the angle between $\mathbf{v}_2$ and $\hat{\mathbf{z}}^n(t)$. $CS(\hat{\mathbf{z}}^n(t))=1$ if $\hat{\mathbf{z}}^n(t)$ aligns with $\mathbf{v}_2$, implying its convergence, and $CS(\hat{\mathbf{z}}^n(t))=0$ if it is orthogonal to $\mathbf{v}_2$. $CS(\hat{\mathbf{z}}^n(t)$ does not just track the convergence, but also tests Proposition 5.1(i), i.e. whether $\hat{\mathbf{z}}^n(t)$ ever gets stuck near one of the unstable fixed points, $\mathbf{v}_k$ for $k \geq 3$. Such a case would cause $CS(\hat{\mathbf{z}}^n(t))=0$, and will be fairly visible in our results if it is a common occurrence. For each simulation run, every random walker starts at an initial position on the graph selected uniformly at random. We use 100 different runs for each simulation setup and display results as averaged over the 100 runs. To test these simulations, we pre-compute values of λ_2 and $\mathbf{v}_2$ for

[6] A walker at node i, will stay there for a random exponential time with rate $d(i)$, after which it will move to a neighbor of i chosen uniformly at random. Hence, the walker uses only local information to walk according to CTMC $\mathbf{Q}=-\mathbf{L}$.



(a) RQ vs t over different n; $\kappa = 1000$ (inset plotted on log-log scale)

(b) CS vs t over different$n$; $\kappa = 1000$

(c) RQ vs t over different κ; $n = 15$

(d) CS vs t over different κ; $n = 15$

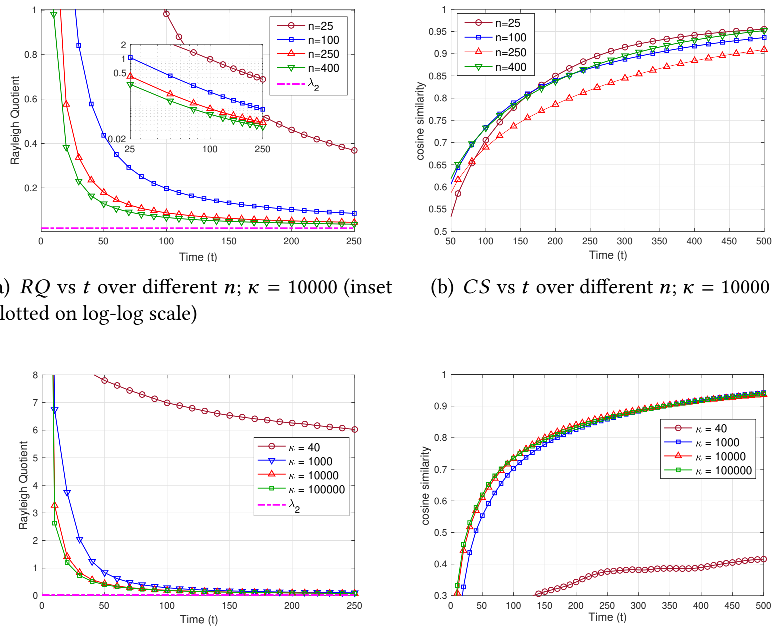
Fig. 3. Simulations for the *Dolphins* graph.

our datasets and use them to compare with RQ and to obtain values of CS respectively. It should be noted that this is purely for comparison purposes, and our framework never requires such computations to be carried out.

6.2 Simulation results over a range of parameters

We first present simulation results for the *Dolphins* graph. Figures 3(a) and (b) show the results for values of $n = 15, 20$ and 25 for each type of walker, with κ set at 1000. For each n, simulation shows the Rayleigh quotient RQ approaching λ_2 of the Laplacian of the graph. This hints convergence to $\mathbf{v}_2$, which is also confirmed by the cosine similarity CS approaching *one*. The inset in Figure 3(a) plots the data on the log-log scale, and shows that RQ decays roughly polynomially fast to λ_2 for each n. As would be expected, $\mathbf{z}^n(t)$ for $n = 25$ outperforms the other two choices of n early on in the simulation, but its early advantage in performance is less pronounced as the simulation progresses.

Figures 3(c) and (d) show convergence results under different values of $\kappa = 10, 100, 1000$ and 10000, while keeping $n = 15$ for each run. Observe that the simulation results for $\kappa = 10$ do not show any convergence to the Fiedler vector. This can be attributed to the failure of $\kappa = 10$ in satisfying Proposition 4.2 by not being sufficiently large. For larger values of κ, we observe robust convergence over a wide range of values with no visible trend in κ. We now present simulation results for the *ego-Facebook* graph, where we perform similar simulations as before. it is interesting to see how well our interacting random walk based method scales to a larger graph. Figures 4(a) and (b) show simulation results for $n = 25, 100, 250$ and 400, for a fixed $\kappa = 10000$. As can be seen in Figure 4(a), the Rayleigh quotient for all four simulation sets converges towards λ_2. The inset in Figure 4(a) shows that similar to results for the *Dolphins* graph, RQ decays roughly polynomially to λ_2. The simulation for $n = 25$ seems to perform worst, at least early on, and $n = 400$ performs best early on.



(a) *RQ* vs t over different n; $\kappa = 10000$ (inset plotted on log-log scale)

(b) *CS* vs t over different n; $\kappa = 10000$

(c) *RQ* vs t over different κ; $n = 100$

(d) *CS* vs t over different κ; $n = 100$

Fig. 4. Simulations for the *ego-Facebook* graph.

An increase in n clearly shows faster convergence of *RQ*, but the big improvement from $n = 25$ to $n = 100$ is nowhere to be seen between $n = 250$ and $n = 400$. This appears to be because the probability of the stochastic process of deviating from the deterministic ODE decays exponentially fast in n, as theorized in Theorem 3.4. This is also the reason why even though *ego-Facebook* is a much larger graph than *Dolphins*, even values of n that are less that 3% of the graph size can provide good approximations, compared to about 50% needed for the much smaller *Dolphins* graph. In the longer run, $\hat{\mathbf{z}}^n(t)$ for all the simulations seems to align well enough with $\mathbf{v}_2$, as can be seen in Figure 4(b). Even for small number of walkers ($n = 25$), the *CS* eventually catches up with the rest and shows the expected close alignment with $\mathbf{v}_2$. This suggests that if high precision approximation is not needed, lower values of n can provide quick, yet reasonably accurate approximations of the Fiedler vector. This is useful for applications such as graph partitioning, where only the signs of the entries of the Fiedler vector matter, not the numerical entries themselves.

We also present simulations over *ego-Facebook* to test the effect of κ. Figures 4 (c) and (d) show the results for $\kappa = 40, 1000, 10000$ and 100000, keeping $n = 100$ constant over all the run[7]. We can observe that $\kappa = 40$ fails to be sufficiently large to satisfy Proposition 4.2 and does not show convergence to $\mathbf{v}_2$. As observed before for the *Dolphins* graph, there is no visible trend when it comes to κ influencing the performance of the simulation, with similar performance for a wide range of higher values of κ. Overall from simulation results of both the *Dolphins* and *ego-Facebook* graphs, it appears that as long as κ is large enough to satisfy Proposition 4.2 (which it seems to satisfy as long as κ is atleast of order $o(N)$) its value does not affect the performance of our framework, with convergence being robust for any choice of large κ.

It should be noted that in no simulated case did large N or large κ affect the ability of our stochastic process $\left(\mathbf{x}^n(t), \mathbf{y}^n(t)\right)_{t\geq 0}$ to closely follow the solutions $\left(\mathbf{x}(t), \mathbf{y}(t)\right)_{t\geq 0}$ of our ODE system

[7] We plot these on a different scale compared to Figures 4(a) and (b) to accommodate all the results in a single frame.

(11), even though they possibly could have caused the bound in (15) from Theorem 3.4 to grow large[8]. This is because (15) is just an upper bound and in reality, the $\left(\mathbf{x}^n(t), \mathbf{y}^n(t)\right)$ could closely follow $\left(\mathbf{x}(t), \mathbf{y}(t)\right)$ over a much larger range of parameters. Moreover, it is an upper bound for the largest deviation over a finite time horizon $[0, T]$ and not for each time $t > 0$, meaning that it provides a worst case bound and efforts taken to make it smaller can be overkill.

6.3 Simulations on dynamic graphs

As briefly mentioned in Section 1, our random walk based framework is expected to perform well in the setting of dynamic graphs. In this section we provide simulation results to support that statement. We modify the *Dolphins* and *ego-Facebook* graphs by removing sets of nodes at certain time points during the course of each simulation run. When a node is removed, we also delete all the edges associated with it. While the instances at which these changes occur are selected for convenience of presentation, the sets of nodes to be removed are randomly generated before the simulation begins, with the condition that their removal does not affect the connectivity of the original graph. To differentiate the dynamic graphs from the original one, we rename our dynamic version of the graphs with node removal as *Dolphins-dyn* and *ego-Facebook-dyn*, respectively. We change the graphs three times in total by deleting a different set of nodes and all related edges at each of those instances. Overall, we delete around 21.0% of nodes from the original *Dolphins* graph and 12.4% of nodes from the original *ego-Facebook* graph, maintaining their connectivity as we do so. Table 1 gives details about the times at which nodes are removed and the number of nodes removed.

	(t_1, # nodes removed)	(t_2, # nodes removed)	(t_3, # nodes removed)
Dolphins-dyn	(25, 4 nodes)	(75, 3 nodes)	(100, 6 nodes)
ego-Facebook-dyn	(50, 196 nodes)	(100, 150 nodes)	(125, 154 nodes)

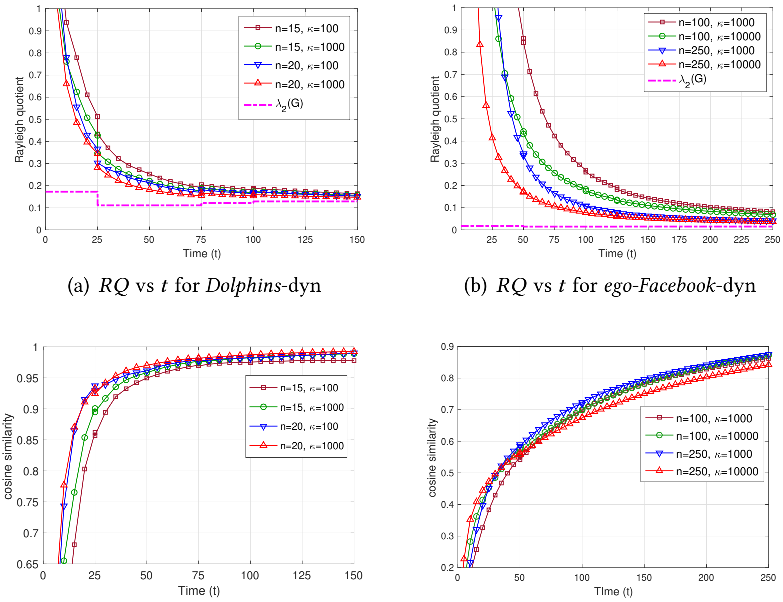
Table 1. Statistics of the dynamic changes made

During the three pre-selected times $t_1, t_2, t_3 > 0$ when all the nodes from the corresponding sets are removed, any random walker currently located at the removed node is redistributed to the position of another randomly selected walker (of the same group) present at an unremoved node of the graph[9]. While plotting the Rayleigh quotient for the dynamic graphs, we represent by the black dotted lines the values of $\lambda_2(\mathcal{G})$, the algebraic connectivity of $\mathcal{G}$, updated to reflect the changes in graph topology.

Figures 5(a) to (d) show our simulation results under the dynamic graph setting over a range of parameters. Figures 5(a) and (b) show results concerning Rayleigh quotient *RQ* for the *Dolphins-dyn* and *ego-Facebook-dyn* graphs. We represent by the black dotted lines, the values of λ_2 updated to reflect the modification in graph topology. As expected, we observe small jumps in *RQ* at the times when nodes are deleted. Better visible in Figure 5(a) than in Figure 5(b), these jumps are similar in size to the jumps in $\lambda_2(\mathcal{G})$, and apart from these, the Rayleigh quotient values monotonically decrease with time without any unexpected fluctuations. From this, we can safely conclude that even after nodes are deleted, trajectories of the simulation do not diverge from their intended paths and keep converging to the Fiedler vector (of the updated/modified graphs). While they do not diverge, the trajectories do face minor changes in terms of a small loss in progress, as can be observed across all the values of n and κ in Figure 5(c) at time $t = 25$, where the cosine similarity

[8] Failure to converge for smaller values of κ in Figures 3(c) and (d), and Figures 4(c) and (d) is attributed to failure to satisfy Proposition 4.2, and is an intrinsic trait of the ODE system (11) itself and not the stochastic process.

[9] We did this only for convenience of the simulation, and the redistribution need not follow a prescribed rule since our theory allows for convergence starting from any arbitrary initial condition.



(a) RQ vs t for *Dolphins*-dyn

(b) RQ vs t for *ego-Facebook*-dyn

(c) CS vs t for *Dolphins*-dyn

(d) CS vs t for *ego-Facebook*-dyn

Fig. 5. Simulations for *Dolphins* and *ego-Facebook* graphs subject to node removals.

CS for *Dolphins-dyn* drops due to deletion of nodes. On the other hand, these changes can also be minor improvements/gains in progress as can be seen more clearly for $n = 250, \kappa = 1000$ in Figure 5(d) at time $t = 50$, where CS for *ego-Facebook-dyn* actually increases due to deletion of nodes. Therefore changes in topology can randomly be either beneficial, or disadvantageous depending on the random state the simulation is in. However, these random effects seem too minor to be a cause of concern, and there is no observable improvement or reduction in performance.

Thus, we can conclude that the framework behaves as would be expected and is robust to dynamical changes in topology, as long as the graph remains connected. It should be noted that even if the graph gets disconnected from time to time, the framework will still, by its design, converge to the Fiedler vector as long as connectivity is eventually restored again[10].

7 EXTENSIONS TO REVERSIBLE MARKOV CHAINS

In Section 3 the CTMC used as the basis for constructing our stochastic process was given by $\mathbf{Q} = -\mathbf{L}$, the negative of the combinatorial Laplacian. We then went on the show convergence of a suitably scaled version of this process to the Fiedler vector $\mathbf{v}_2$ of $\mathbf{L}$ in Section 4 and discussed the long run behavior of our stochastic process in Section 5. In this section, we generalize our results by extending them to not just the combinatorial Laplacian $\mathbf{L}$, but also any time reversible CTMC kernel which, as mentioned in Section 1, is an important part of our paper's contribution.

Consider the kernel $\mathbf{Q} \in \mathbb{R}^{N \times N}$ of an ergodic time reversible CTMC on a finite state space $\mathcal{N}$ (where $|\mathcal{N}| = N$), not necessarily symmetric. We denote by $\pi \in \mathbb{R}^N$, its stationary distribution ($\pi^T \mathbf{Q} = 0$). The vector π is also the first left eigenvector of $\mathbf{Q}$. If we now define our stochastic process in Section 3.1 and all the consequent systems with respect to time reversible $\mathbf{Q}$, we still have $\mathbf{z}^n(t)^T \mathbf{1} = 0$ (and $\mathbf{z}(t)^T \mathbf{1} = 0$) by construction. Earlier in Section 3 when $\mathbf{Q}$ was a symmetric

[10] For a disconnected graph, the Fiedler vector is not well defined, since the second eigenvalue is also *zero*, and no longer strictly positive. Therefore talking about the Fiedler vector of the whole graph makes sense only when it is connected.

matrix, its eigenvectors formed an orthogonal basis of the N-dimensional Euclidean space. This meant that being orthogonal to $\mathbf{1}$ automatically guaranteed being in a set where $\mathbf{v}_2$ minimized the Lyapunov function V. This, however, is not the case for any general non-symmetric, time reversible CTMC kernel $\mathbf{Q}$. We will extend our results to time reversible Markov chain kernels by introducing a specially constructed inner product which ensures orthogonality to the first left eigenvector of $\mathbf{Q}$, π.

For these purposes, we first formally define time reversible Markov chains.

Definition 7.1. *A Markov chain with $\mathbf{Q}$ is called 'time reversible' if and only if there exists a unique π such that for all $i, j \in \mathcal{N}$, $i \neq j$, the pair $(\pi, \mathbf{Q})$ satisfies the 'detailed balance equation' $\pi_i Q_{ij} = \pi_j Q_{ji}$.*

For a given pair $(\pi, \mathbf{Q})$, define an N-dimensional vector space $\mathcal{H}_{\frac{1}{\pi}}$ endowed with an inner product $\langle \cdot, \cdot \rangle_{\frac{1}{\pi}}$, defined as

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\frac{1}{\pi}} = \mathbf{x}^T \Pi^{-1} \mathbf{y}, \qquad \text{for any } \mathbf{x}, \mathbf{y} \in \mathbb{R}^N$$

where $\Pi = \mathbf{D}_\pi$.

Lemma 7.2. *[9] The pair $(\pi, \mathbf{Q})$ is reversible if and only if $\mathbf{Q}^T$, acting as a linear operator from $\mathcal{H}_{\frac{1}{\pi}}$ onto itself, satisfies $\langle \mathbf{Q}^T \mathbf{u}, \mathbf{v} \rangle_{\frac{1}{\pi}} = \langle \mathbf{y}, \mathbf{Q}^T \mathbf{v} \rangle_{\frac{1}{\pi}}$.*

Such operators are called self-adjoint or Hermitian and always have real eigenvalues. Thus, we can define the ordering of eigenvalues of $-\mathbf{Q}$ as $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_N$, just like we did for $\mathbf{L}$. The eigenvectors of self-adjoint operators are orthogonal with respect to the related inner product. Hence, for any two left eigenvectors $\mathbf{v}_i$ and $\mathbf{v}_k$ of $\mathbf{Q}$, $i \neq k$, we have $\langle \mathbf{v}_i, \mathbf{v}_k \rangle_{\frac{1}{\pi}} = 0$. Picking $i = 1$, and substituting $\mathbf{v}_1 = \pi$, we obtain

$$\langle \pi, \mathbf{v}_k \rangle_{\frac{1}{\pi}} = \pi^T \Pi^{-1} \mathbf{v}_k = \mathbf{1}^T \mathbf{v}_k = 0.$$

Therefore, the Euclidean subspace that is orthogonal to $\mathbf{1}$, is also a subspace of $\mathcal{H}_{\frac{1}{\pi}}$ orthogonal to π in terms of our new inner product, $\langle \cdot, \cdot \rangle_{\frac{1}{\pi}}$. Also, $\mathbf{z}^n(t)^T \mathbf{1} = 0 \iff \langle \mathbf{z}^n(t), \pi \rangle_{\frac{1}{\pi}} = 0$, giving us the desired orthogonality with π. This allows us to recover results from Section 4 by a redefinition of the Lyapunov function V in terms of $\langle \cdot, \cdot \rangle_{\frac{1}{\pi}}$, i.e.

$$V(\mathbf{u}) = \frac{1}{2} \langle \mathbf{u}, -\mathbf{Q}^T \mathbf{u} \rangle_{\frac{1}{\pi}} \text{ for all } \mathbf{u} \in \mathcal{H},$$

and appropriately using the new inner product (i.e. $\langle \mathbf{u}, \mathbf{w} \rangle_{\frac{1}{\pi}}$ instead of $\mathbf{u}^T \mathbf{w}$, for any two vectors $\mathbf{u}$ and $\mathbf{w}$) wherever necessary. The results from Section 5 never required a specific form of $\mathbf{Q}$ to work as long the results from Sections 3 and 4 held, and now therefore hold true for any choice of time reversible $\mathbf{Q}$.

From the above, we have extended our main results to be applicable to any time reversible CTMC kernel $\mathbf{Q}$. It is important to note that our extension is purely based on an update in the inner product used for the analysis purposes in Section 4, and requires no modification to the way our stochastic process is constructed in Section 3, making $\mathbf{Q}$ simply a *plug-and-play* term in our framework.

We now show the applicability of this extension to any ergodic, time reversible DTMC kernel $\mathbf{P}$ as well. The Fiedler vector for such matrices is now the eigenvector corresponding to the second largest eigenvalue of $\mathbf{P}$, also known as the spectral gap [1, 9, 32]. We start by defining a CTMC kernel $\mathbf{Q}_p$ based on the time reversible $\mathbf{P}$ as $\mathbf{Q}_p \triangleq \mathbf{P} - \mathbf{I}$[11]. Suppose π is the unique stationary distribution

[11] See that $\mathbf{Q}_p \mathbf{1} = (\mathbf{P} - \mathbf{I})\mathbf{1} = 0$, and $\mathbf{Q}_p$ has non-negative off-diagonal entries and negative diagonal entries. It is therefore a well defined CTMC kernel

of $\mathbf{P}$, such that the pair $(\pi, \mathbf{P})$ satisfies the *detailed balance equation* (DBE) from Definition 7.1. Then, the pair $(\pi, \mathbf{Q}_p)$ also satisfies the DBE and $\mathbf{Q}_p$ is a time reversible CTMC. This means that all our theoretical results stand for $\mathbf{Q}_p$ if we use it as the basis for generating our stochastic process in Section 3.1. Therefore, $\mathbf{z}^n(t)$ for a system of random walkers, which walk according to the CTMC $\mathbf{Q}_p = \mathbf{P} - \mathbf{I}$ and interact in the same manner as in the previous sections, will approximate $\mathbf{v}_2$ of the DTMC $\mathbf{P}$[12] in the long run.

We can also apply this extension to the random walk Laplacian $L^{rw} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$, and the normalized Laplacian $\mathcal{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, introduced earlier in Section 2.2. We can do this by first observing that $\mathbf{D}^{-1}\mathbf{A}$ is a DTMC kernel with the degree distribution vector $\frac{1}{\mathbf{d}^T\mathbf{1}}\mathbf{d}$ serving as its stationary distribution. The pair $\left(\frac{1}{\mathbf{d}^T\mathbf{1}}\mathbf{d}, \mathbf{D}^{-1}\mathbf{A}\right)$ is also time reversible [1, 9] and satisfies the DBE. Thus, setting $\mathbf{Q}_{rw} \triangleq \mathbf{D}^{-1}\mathbf{A} - \mathbf{I} = -\mathbf{L}^{rw}$ as our CTMC kernel extends all the results of the paper to the random walk Laplacian and allows us to obtain approximations for $\mathbf{v}_2^{rw}$, the Fiedler vector of the random walk Laplacian. Given a component $[\mathbf{v}_2^{rw}]_i$, the corresponding component of the Fiedler vector $\bar{\mathbf{v}}_2$ of the normalized Laplacian $\mathcal{L}$ can be obtained by setting $[\bar{\mathbf{v}}_2]_i = [\mathbf{v}_2^{rw}]_i / \sqrt{d(i)}$, due to the similarity relationship between the two matrices as shown in Section 2.2. Therefore, once $[\mathbf{v}_2^{rw}]_i$ is approximated, a simple localized computation involving only the degree of the concerned node allows us to obtain $[\bar{\mathbf{v}}_2]_i$ as well. With this, we can now use our framework to approximate (on-the-fly) the Fiedler vectors of all the three important graph Laplacians as special cases of our extended theoretical results.

8 CONCLUDING REMARKS

In this paper, given any time reversible Markov Chain kernel, we have detailed the construction of a stochastic process based on interacting random walkers. Random walk algorithms usually relate to the leading/principal eigenvectors of their respective kernels. In fact, the field of Markov chain Monte Carlo (MCMC) is dedicated to the problem of using a random walk to sample according to a given probability distribution. Famous examples include the Metropolis Hastings Random walk and the Gibbs sampler [23, 30, 33, 40], which construct Markov chains and leverage the ergodic theorem to sample according to its first eigenvector. While usually these are restricted to sampling on undirected graphs, [29] samples from a directed graph using a novel method which involves *mapping* a target distribution to the quasi-stationary distribution (QSD) of a sub-stochastic Markov chain. However, this technique also essentially leverages properties of a leading eigenvector, which is the QSD in this case. No such random walk type technique has been applied towards approximating the second eigenvector. By relating our stochastic process to a deterministic ODE system we show convergence to the second eigenvector, making our random walk based method a first in literature.

9 ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions that greatly improved the quality of this paper. This work was supported in part by National Science Foundation under Grant Nos. IIS-1910749 and CNS-1824518.

REFERENCES

[1] David Aldous and James Allen Fill. 2002. Reversible Markov Chains and Random Walks on Graphs. Unfinished monograph, recompiled 2014, available at http://www.stat.berkeley.edu/~aldous/RWG/book.html.

[2] R. Andersen, F. Chung, and K. Lang. 2006. Local Graph Partitioning using PageRank Vectors. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. 475–486.

[12] Since eigenvectors of $\mathbf{Q}_p$ and $\mathbf{P}$ are the same.

[3] B. Aspvall and J. Gilbert. 1984. Graph Coloring Using Eigenvalue Decomposition. *SIAM Journal on Algebraic Discrete Methods* 5, 4 (1984), 526–538.

[4] John Ball. 2016. Semiflows, Lyapunov functions and approach to equilibrium. University lecture. https://people.maths.ox.ac.uk/ball/Teaching/cdtsemiflows16.pdf

[5] S. T. Barnard, A. Pothen, and H. D. Simon. 1993. A spectral algorithm for envelope reduction of sparse matrices. In *Proceedings of the 1993 ACM/IEEE Conference on Supercomputing (SC'93)*. 493–502.

[6] Stephen T. Barnard and Horst D. Simon. 1993. A Fast Multilevel Implementation of Recursive Spectral Bisection for Partitioning Unstructured Problems. *Concurrency - Practice and Experience* 6 (1993), 101–117.

[7] Michel Benaïm and Jörgen W. Weibull. 2003. Deterministic Approximation of Stochastic Evolution in Games. *Econometrica* 71, 3 (2003), 873–903.

[8] Alexander Bertrand and Marc Moonen. 2013. Distributed computation of the Fiedler vector with application to topology inference in ad hoc networks. *Signal Processing* 93, 5 (2013), 1106–1117.

[9] P. Brémaud. 1999. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues.* Springer-Verlag.

[10] John C. Urschel, Xiaozhe Hu, Jinchao Xu, and Ludmil T. Zikatanov. 2015. A Cascadic Multigrid Algorithm for Computing the Fiedler Vector of Graph Laplacians. *Journal of Computational Mathematics* 33 (2015), 209–226.

[11] F. R. K. Chung. 1997. *Spectral Graph Theory.* American Mathematical Society.

[12] Richard Combes and Mikael Touati. 2019. Computationally Efficient Estimation of the Spectral Gap of a Markov Chain. *Proc. ACM Meas. Anal. Comput. Syst.* 3, 1, Article 7 (2019), 21 pages.

[13] K.Ch. Das. 2004. The Laplacian spectrum of a graph. *Computers & Mathematics with Applications* 48, 5 (2004), 715–724.

[14] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. 2004. Kernel K-means: Spectral Clustering and Normalized Cuts. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*. 551–556.

[15] Chris Ding, Xiaofeng He, and Horst D. Simon. 2005. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SIAM International Conference on Data Mining (ICDM'05)*.

[16] C. H. Q. Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and H. D. Simon. 2001. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings 2001 IEEE International Conference on Data Mining (ICDM'01)*. 107–114.

[17] Moez Draief and Laurent Massouli. 2010. *Epidemics and Rumours in Complex Networks* (1st ed.). Cambridge University Press, New York, NY, USA.

[18] Rafael Van Driessche and Dirk Roose. 1995. A Grapg Contraction Algorithm for the Fast Calculation of the Fiedler Vector of a Graph. In *Proceedings of the seventh SIAM conference on Parallel Processing for Scientific Computing (PPSC'95)*.

[19] Miroslav Fiedler. 1973. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal* 23(98) (1973), 298–305.

[20] Miroslav Fiedler. 1975. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal* 25(100), 4 (1975), 619–633.

[21] L. Hagen and A. B. Kahng. 1992. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 11, 9 (1992), 1074–1085.

[22] Insu. Han, Dmitry. Malioutov, Haim. Avron, and Jinwoo. Shin. 2017. Approximating Spectral Sums of Large-Scale Matrices using Stochastic Chebyshev Approximations. *SIAM Journal on Scientific Computing* 39, 4 (2017), A1558–A1585.

[23] W. K. Hastings. 1970. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika* 57, 1 (1970), 97–109.

[24] Morris W. Hirsch and Stephen Smale. 1974. *Differential equations, dynamical systems, and linear algebra.* Acad. Press, San Diego, CA, USA.

[25] Roger A. Horn and Charles R. Johnson. 2012. *Matrix Analysis* (2nd ed.). Cambridge University Press, New York, NY, USA.

[26] Daniel Hsu, Aryeh Kontorovich, and Csaba Szepesvári. 2015. Mixing Time Estimation in Reversible Markov Chains from a Single Sample Path. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'15)*. 1459–1467.

[27] Yehuda. Koren, Liran. Carmel, and David. Harel. 2003. Drawing Huge Graphs by Algebraic Multigrid Optimization. *Multiscale Modeling & Simulation* 1, 4 (2003), 645–673.

[28] Sibsankar Kundu, Dan C. Sorensen, and George N. Phillips, Jr. 2004. Automatic domain decomposition of proteins by a Gaussian Network Model. *Proteins: Structure, Function, and Bioinformatics* 57, 4 (2004), 725–733.

[29] Chul-Ho Lee, Min Kang, and Do Young Eun. 2019. Non-Markovian Monte Carlo on Directed Graphs. In *Proceedings of the ACM SIGMETRICS/IFIP Performance 2019 (SIGMETRICS'19)*.

[30] Chul-Ho Lee, Xin Xu, and Do Young Eun. 2012. Beyond Random Walk and Metropolis-Hastings Samplers: Why You Should Not Backtrack for Unbiased Graph Sampling. In *Proceedings of the ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'12)*. 319–330.

[31] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[32] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. 2006. *Markov chains and mixing times.* American Mathematical Society.

[33] Jun S. Liu. 2004. *Monte Carlo Strategies in Scientific Computing.* Springer-Verlag.

[34] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson. 2003. The Bottlenose Dolphin Community of Doubtful Sound Features a Large Proportion of Long-Lasting Associations. *Behavioral Ecology and Sociobiology* 54 (2003), 396–405.

[35] Carl D. Meyer. 2000. *Matrix Analysis and Applied Linear Algebra.* SIAM.

[36] R. K. Miller and A. N. Michel. 1982. *Ordinary Differential Equations.* Academic Press, New York, NY, USA.

[37] A.Y. Ng, Michael Jordan, and Y Weiss. 2001. On Spectral Clustering: Analysis and an Algorithm. *Adv. Neural Inf. Process. Syst.* 2 (11 2001).

[38] Pekka Orponen and Satu Elisa Schaeffer. 2005. Local Clustering of Large Graphs by Approximate Fiedler Vectors. In *Proceedings of the 4th International Conference on Experimental and Efficient Algorithms (WEA'05).* 524–533.

[39] Lawrence Perko. 2001. *Differential Equations and Dynamical Systems (Third ed.).* New York: Springer.

[40] P. H. Peskun. 1973. Optimum Monte-Carlo Sampling Using Markov Chains. *Biometrika* 60, 3 (1973), 607–612.

[41] Alex Pothen, Horst D. Simon, and Kan-Pu Liou. 1990. Partitioning Sparse Matrices with Eigenvectors of Graphs. *SIAM J. Matrix Anal. Appl.* 11, 3 (1990), 430–452.

[42] Alex. Pothen, Horst D. Simon, and Kang-Pu. Liou. 1990. Partitioning Sparse Matrices with Eigenvectors of Graphs. *SIAM J. Matrix Anal. Appl.* 11, 3 (1990), 430–452.

[43] S. J. Shepherd, C. B. Beggs, and S. Jones. 2007. Amino acid partitioning using a Fiedler vector model. *European Biophysics Journal* 37, 1 (2007), 105–109.

[44] Jianbo Shi and J. Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 8 (2000), 888–905.

[45] H.D. Simon. 1991. Partitioning of unstructured problems for parallel processing. *Computing Systems in Engineering* 2, 2 (1991), 135–148.

[46] Brian Slininger. 2013. Fiedlers Theory of Spectral Graph Partitioning.

[47] Jean-Jacques E. Slotine and Weiping Li. 1991. *Applied Nonlinear Control.* Prentice-Hall.

[48] Daniel A. Spielman and Nikhil. Srivastava. 2011. Graph Sparsification by Effective Resistances. *SIAM J. Comput.* 40, 6 (2011), 1913–1926.

[49] Daniel A. Spielman and Shang-Hua. Teng. 2013. A Local Clustering Algorithm for Massive Graphs and Its Application to Nearly Linear Time Graph Partitioning. *SIAM J. Comput.* 42, 1 (2013), 1–26.

[50] Daniel A. Spielman and Shang-Hua. Teng. 2014. Nearly Linear Time Algorithms for Preconditioning and Solving Symmetric, Diagonally Dominant Linear Systems. *SIAM J. Matrix Anal. Appl.* 35, 3 (2014), 835–885.

[51] Ashok Srinivasan and Michael Mascagni. 2002. Monte Carlo Techniques for Estimating the Fiedler Vector in Graph Applications. In *Proceedings of the International Conference on Computational Science-Part II (ICCS'02).* 635–645.

[52] M. Vidyasagar. 1993. *Nonlinear Systems Analysis.* Prentice-Hall, Englewood Cliffs, NJ, USA.

[53] Ulrike von Luxburg. 2007. A Tutorial on Spectral Clustering. *Statistics and computing* 17, 4 (2007), 395–416.

[54] Yu and Shi. 2003. Multiclass spectral clustering. In *Proceedings Ninth IEEE International Conference on Computer Vision (ICCV'03).* 313–319.

A APPENDIX FOR SECTION 3

A.1 Proof of Proposition 3.1:

We only show that (7) implies (9), since the steps relating (8) and (10) are exactly the same in terms of algebra. By substituting (7) into (3), we obtain

$$F_x(\mathbf{x},\mathbf{y}) = \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N},j\neq i}(\mathbf{e}_i-\mathbf{e}_j)Q_{ji}x_j + \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N},j\neq i}(\mathbf{e}_i-\mathbf{e}_j)(\kappa x_j y_j)x_i = \mathbf{u}+\mathbf{w}, \tag{31}$$

where we use $\mathbf{u}\in\mathbb{R}^N$ and $\mathbf{w}\in\mathbb{R}^N$ to denote the two summation terms. Observe that the k^{th} entry of $\mathbf{u}$ can be written as

$$u_k = \sum_{j\neq k}(1-0)Q_{jk}x_j + \Big[\sum_{i\neq k}(0-1)Q_{ki}\Big]x_k = \sum_{j\neq k}Q_{jk}x_j + Q_{kk}x_k = \sum_{j\in\mathcal{N}}Q_{jk}x_j = [\mathbf{Q}^T\mathbf{x}]_k,$$

suggesting that $\mathbf{u}=[u_k]$ can be written as $\mathbf{u}=\mathbf{Q}^T\mathbf{x}$. Similarly, the k^{th} entry of $\mathbf{w}$ can be written as

$$\begin{aligned} w_k &= \Big[\sum_{j\neq k}(1-0)\kappa x_j y_j\Big]x_k + \sum_{i\neq k}(0-1)(\kappa x_k y_k)x_i \\ &= \Big[\sum_{j\neq k}(1-0)\kappa x_j y_j\Big]x_k + (\kappa x_k y_k)x_k + \sum_{i\neq k}(0-1)(\kappa x_k y_k)x_i - (\kappa x_k y_k)x_k \\ &= \Big[\sum_{j\in\mathcal{N}}\kappa x_j y_j\Big]x_k - \kappa x_k y_k\sum_{i\in\mathcal{N}}x_i = [\kappa\mathbf{x}^T\mathbf{y}]x_k - \kappa x_k y_k, \end{aligned}$$

suggesting that $\mathbf{w}=[w_k]$ can be written as $\mathbf{w}=[\kappa\mathbf{x}^T\mathbf{y}]\mathbf{x}-\kappa\mathbf{D}_{\mathbf{y}}\mathbf{x}$. Substituting the expressions for $\mathbf{u}$ and $\mathbf{w}$ in (31), we obtain (9), which completes the proof.

A.2 Proof of Proposition 3.2:

For this proof, we make a change of notation. Vectors $(\mathbf{x},\mathbf{y})\in\mathbb{R}^{2N}$ will now be written as $\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}$. Similarly, we have

$$F(\mathbf{x},\mathbf{y}) = \begin{bmatrix}F_x(\mathbf{x},\mathbf{y})\\ F_y(\mathbf{x},\mathbf{y})\end{bmatrix} = \begin{bmatrix}\mathbf{Q}^T-\kappa\mathbf{D}_{\mathbf{y}} & 0\\ 0 & \mathbf{Q}^T-\kappa\mathbf{D}_{\mathbf{x}}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} + [\kappa\mathbf{x}^T\mathbf{y}]\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}.$$

Denote by $||\cdot||$ the 2−norm for any vector in $\mathbb{R}^{2N}$, and the induced matrix norm for any $2N\times 2N$ dimensional matrix. Using this notation for any $(\mathbf{x},\mathbf{y})\in S$ and $(\mathbf{u},\mathbf{w})\in S$, by the mean value theorem, there exists a point $(\mathbf{a},\mathbf{b})\in S$ on the line segment joining $(\mathbf{x},\mathbf{y})$ and $(\mathbf{u},\mathbf{w})$ such that

$$||F(\mathbf{x},\mathbf{y})-F(\mathbf{u},\mathbf{w})|| = ||\mathbf{J}_F(\mathbf{a},\mathbf{b})||\Big|\Big|\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}-\begin{bmatrix}\mathbf{u}\\ \mathbf{w}\end{bmatrix}\Big|\Big|, \tag{32}$$

where $\mathbf{J}_F(\mathbf{a},\mathbf{b}))$ is the Jacobian matrix of F evaluated at $(\mathbf{a},\mathbf{b})$, given by

$$\mathbf{J}_F(\mathbf{a},\mathbf{b}) = \begin{bmatrix}\mathbf{Q} & 0\\ 0 & \mathbf{Q}\end{bmatrix} + \kappa\begin{bmatrix}\mathbf{a}\mathbf{b}^T+\mathbf{D}_{\mathbf{a}}\mathbf{D}_{\mathbf{b}}-\mathbf{D}_{\mathbf{b}} & \mathbf{a}\mathbf{a}^T-\mathbf{D}_{\mathbf{a}}\\ \mathbf{b}\mathbf{b}^T-\mathbf{D}_{\mathbf{b}} & \mathbf{b}\mathbf{a}^T+\mathbf{D}_{\mathbf{b}}\mathbf{D}_{\mathbf{a}}-\mathbf{D}_{\mathbf{a}}\end{bmatrix}.$$

All elements of the Jacobian matrix are bounded uniformly over the line segment joining $(\mathbf{x},\mathbf{y})$ and $(\mathbf{u},\mathbf{w})$, which in turn means that $||\mathbf{J}_F(\mathbf{a},\mathbf{b})||$ is bounded too. Therefore, $F:S\longrightarrow\mathbb{R}^{2N}$ is Lipschitz continuous which is necessary and sufficient for the existence and uniqueness of solutions to (11) [24, 36].

A.3 Proof of Theorem 3.4

Proof. (Theorem 3.4) Before the main part of the proof, we borrow some results from literature which will be used later.

Proposition A.1. *(Proposition 5.2 in [17]) Let Y be a Poisson process with unit rate (i.e. $Y(t) \sim$ Poisson(t) for all $t \geq 0$). Then for any $\epsilon > 0$ and $T > 0$,*

$$\mathbf{P}\Big(\sup_{0\leq t\leq T} |Y(t) - t| \geq \epsilon\Big) \leq 2\epsilon \exp\big(-T \cdot h(\epsilon/T)\big)$$

where $h(x) = (1 + x)\log(1 + x) - x$.

We also state Gronwall's inequality.

Lemma A.2. *(Gronwall's inequality) Let f be a bounded, real valued function on $[0, T]$ satisfying $f(t) \leq a + b\int_0^t u(s)ds$ for all $t \in [0, T]$, where a and b are non-zero, real valued constants. Then, $f(t) \leq a\exp(bt)$ for all $t \in [0, T]$.*

Observe that since $Q_{ji} = -L_{ji} = A_{ji}$ is no bigger than 1 for all $j \neq i$ and $x_i, y_i \in (0, 1)$ for all $i \in \mathcal{N}$,we can bound the terms in (3) and (4), and obtain[13]

$$\bar{Q}^n_{x:j\to i}(\mathbf{x}, \mathbf{y}) \leq n(1 + \kappa) \quad \text{and} \quad \bar{Q}^n_{y:j\to i}(\mathbf{x}, \mathbf{y}) \leq n(1 + \kappa). \tag{33}$$

We now proceed with the main body of our proof.

For a unit rate Poisson process $Y(t)$, its centered version is given by $\hat{Y}(t) \triangleq Y(t) - t$ for all $t \geq 0$. Let Y^x_{ji} and Y^y_{ji} for all $i, j \in \mathcal{N},\ j \neq i$ be independent Poisson processes of unit rate. Let $\hat{Y}^x_{ji}$ and $\hat{Y}^y_{ji}$ be their centered versions, i.e. $\hat{Y}^x_{ji}(t) = Y^x_{ji}(t) - t$ and $\hat{Y}^y_{ji}(t) = Y^y_{ji}(t) - t$ for any $t \geq 0$.

The continuous time Markov chain $\{\mathbf{x}^n(t), \mathbf{y}^n(t)\}_{t\geq 0}$ can be constructed for any $t \geq 0$ as

$$\begin{aligned}\big(\mathbf{x}^n(t), \mathbf{y}^n(t)\big) = \big(\mathbf{x}^n(0), \mathbf{y}^n(0)\big) &+ \sum_{i\in\mathcal{N}}\sum_{j\neq i}\Big(\frac{\mathbf{e}_i - \mathbf{e}_j}{n}, 0\Big)Y^x_{ji}\Big(\int_0^t \bar{Q}^n_{x:j\to i}\big(\mathbf{x}^n(s), \mathbf{y}^n(s)\big)\Big)\\ &+ \sum_{i\in\mathcal{N}}\sum_{j\neq i}\Big(0, \frac{\mathbf{e}_i - \mathbf{e}_j}{n}\Big)Y^y_{ji}\Big(\int_0^t \bar{Q}^n_{y:j\to i}\big(\mathbf{x}^n(s), \mathbf{y}^n(s)\big)\Big).\end{aligned} \tag{34}$$

Indeed, for any $i, j \in \mathcal{N},\ j \neq i$, $\Big(\frac{\mathbf{e}_i-\mathbf{e}_j}{n}, 0\Big)$ and $\Big(0, \frac{\mathbf{e}_i-\mathbf{e}_j}{n}\Big)$ are the admissible jumps. From (3) and (4), at any instant $s \in [0, t]$, these jumps take place with rate $\bar{Q}^n_{x:j\to i}\big(\mathbf{x}^n(s), \mathbf{y}^n(s)\big)$ and $\bar{Q}^n_{y:j\to i}\big(\mathbf{x}^n(s), \mathbf{y}^n(s)\big)$. The Poisson processes that counts the number of such jumps up till time $t \in [0, \infty)$ are therefore non-homogeneous Poisson processes, which are given by $Y^x_{ji}\Big(\int_0^t \bar{Q}^n_{x:j\to i}\big(\mathbf{x}^n(s), \mathbf{y}^n(s)\big)\Big)$ and $Y^y_{ji}\Big(\int_0^t \bar{Q}^n_{y:j\to i}\big(\mathbf{x}^n(s), \mathbf{y}^n(s)\big)\Big)$.

(34) can be rewritten in terms of F, $\hat{Y}^x_{ji}$ and $\hat{Y}^y_{ji}$ (the centered verions of Y^x_{ji} and Y^y_{ji}) as

$$\begin{aligned}\big(\mathbf{x}^n(t), \mathbf{y}^n(t)\big) = \big(\mathbf{x}^n(0), \mathbf{y}^n(0)\big) &+ \sum_{i\in\mathcal{N}}\sum_{j\neq i}\Big(\frac{\mathbf{e}_i - \mathbf{e}_j}{n}, 0\Big)\hat{Y}^x_{ji}\Big(\int_0^t \bar{Q}^n_{x:j\to i}\big(\mathbf{x}^n(s), \mathbf{y}^n(s)\big)\Big)\\ &+ \sum_{i\in\mathcal{N}}\sum_{j\neq i}\Big(0, \frac{\mathbf{e}_i - \mathbf{e}_j}{n}\Big)\hat{Y}^y_{ji}\Big(\int_0^t \bar{Q}^n_{y:j\to i}\big(\mathbf{x}^n(s), \mathbf{y}^n(s)\big)\Big) + \int_0^t F\big(\mathbf{x}(s), \mathbf{y}(s)\big)ds.\end{aligned} \tag{35}$$

[13] for $\mathbf{Q} \neq -\mathbf{L}$, the entries are still bounded by some positive constant given by $C = \max_{i,j\in\mathcal{N}} Q_{ji}$. In this case, we use C instead of 1 to bound Q_{ji}. The rest of the steps remain the same.

Subtracting (6) from (35) and taking norm yields

$$\begin{aligned}\left\|\big(\mathbf{x}^n(t),\mathbf{y}^n(t)\big)-\big(\mathbf{x}(t),\mathbf{y}(t)\big)\right\| \leq{}& \left\|\big(\mathbf{x}^n(0),\mathbf{y}^n(0)\big)-\big(\mathbf{x}(0),\mathbf{y}(0)\big)\right\| \\ &+\int_0^t \left\|F\big(\mathbf{x}^n(s),\mathbf{y}^n(s)\big)-F\big(\mathbf{x}(s),\mathbf{y}(s)\big)\right\| \\ &+\sum_{i\in\mathcal{N}}\sum_{j\neq i}\left\|\Big(\frac{\mathbf{e}_i-\mathbf{e}_j}{n},0\Big)\right\|\left|\hat{Y}^x_{ji}\Big(\int_0^t \bar{Q}^n_{x:j\to i}\big(\mathbf{x}^n(s),\mathbf{y}^n(s)\big)\Big)\right| \\ &+\sum_{i\in\mathcal{N}}\sum_{j\neq i}\left\|\Big(0,\frac{\mathbf{e}_i-\mathbf{e}_j}{n}\Big)\right\|\left|\hat{Y}^y_{ji}\Big(\int_0^t \bar{Q}^n_{y:j\to i}\big(\mathbf{x}^n(s),\mathbf{y}^n(s)\big)\Big)\right|\end{aligned} \tag{36}$$

where the inequality comes from repeated applications of triangle inequality. Note that $\left\|\Big(0,\frac{\mathbf{e}_i-\mathbf{e}_j}{n}\Big)\right\| = \left\|\Big(\frac{\mathbf{e}_i-\mathbf{e}_j}{n},0\Big)\right\| = \sqrt{2}/n$. Thus, we can rewrite the above inequality as

$$\begin{aligned}\left\|\big(\mathbf{x}^n(t),\mathbf{y}^n(t)\big)-\big(\mathbf{x}(t),\mathbf{y}(t)\big)\right\| \leq{}& \left\|\big(\mathbf{x}^n(0),\mathbf{y}^n(0)\big)-\big(\mathbf{x}(0),\mathbf{y}(0)\big)\right\| \\ &+\int_0^t \left\|F\big(\mathbf{x}^n(s),\mathbf{y}^n(s)\big)-F\big(\mathbf{x}(s),\mathbf{y}(s)\big)\right\| \\ &+\sum_{i\in\mathcal{N}}\sum_{j\neq i}\frac{\sqrt{2}}{n}\left|\hat{Y}^x_{ji}\Big(\int_0^t \bar{Q}^n_{x:j\to i}\big(\mathbf{x}^n(s),\mathbf{y}^n(s)\big)\Big)\right| \\ &+\sum_{i\in\mathcal{N}}\sum_{j\neq i}\frac{\sqrt{2}}{n}\left|\hat{Y}^y_{ji}\Big(\int_0^t \bar{Q}^n_{y:j\to i}\big(\mathbf{x}^n(s),\mathbf{y}^n(s)\big)\Big)\right|.\end{aligned}$$

(A1) says $\|\big(\mathbf{x}(0),\mathbf{y}(0)\big)-\big(\mathbf{x}^n(0),\mathbf{y}^n(0)\big)\| = 0$ for all $n\in\mathbb{N}$. From (A1), Proposition 3.2, and using the bounds from (33), we obtain

$$\begin{aligned}\left\|\big(\mathbf{x}^n(t),\mathbf{y}^n(t)\big)-\big(\mathbf{x}(t),\mathbf{y}(t)\big)\right\| \leq{}& \int_0^t M\left\|\big(\mathbf{x}^n(t),\mathbf{y}^n(t)\big)-\big(\mathbf{x}(t),\mathbf{y}(t)\big)\right\| \\ &+\sum_{i\in\mathcal{N}}\sum_{j\neq i}\frac{\sqrt{2}}{n}\Big[\left|\hat{Y}^x_{ji}\big(n(1+\kappa)t\big)\right|+\left|\hat{Y}^y_{ji}\big(n(1+\kappa)t\big)\right|\Big].\end{aligned} \tag{37}$$

Define

$$\varepsilon^n_{ji}(t) \triangleq \sqrt{2}\Big[\left|\hat{Y}^x_{ji}\big(n(1+\kappa)t\big)\right|+\left|\hat{Y}^y_{ji}\big(n(1+\kappa)t\big)\right|\Big] \tag{38}$$

for any $i,j\in\mathcal{N},\ j\neq i$. This quantity is what we would like to control. Observe that

$$\begin{aligned}\mathbf{P}\Big(\sup_{0\leq t\leq T}\sum_{i\in\mathcal{N}}\sum_{j\neq i}\frac{1}{n}\varepsilon^n_{ji}(t)\geq\epsilon\Big) &\leq \sum_{i\in\mathcal{N}}\sum_{j\neq i}\mathbf{P}\Big(\sup_{0\leq t\leq T}\varepsilon^n_{ji}(t)\geq\frac{n\epsilon}{N(N-1)}\Big) \\ &\leq \sum_{x,y}\sum_{i\in\mathcal{N}}\sum_{j\neq i}\mathbf{P}\Big(\sup_{0\leq t\leq T}|\hat{Y}(n(1+\kappa)t)|\geq\frac{n\epsilon}{\sqrt{2}N(N-1)}\Big) \\ &= 2N(N-1)\mathbf{P}\Big(\sup_{0\leq s\leq n(1+\kappa)T}|\hat{Y}(s)|\geq\frac{n\epsilon}{\sqrt{2}N(N-1)}\Big) \\ &\leq 4N(N-1)\exp\left(-n(1+\kappa)T\cdot h\Big(\frac{\epsilon}{\sqrt{2}N(N-1)(1+\kappa)T}\Big)\right).\end{aligned} \tag{39}$$

The first inequality comes from the fact that $\mathbf{P}\big(\sum_{i=1}^{k} X_i \geq \epsilon\big) \leq \sum_{i=1}^{k} \mathbf{P}\big(X_i \geq \epsilon/n\big)$.. Applying the same to $\varepsilon_{ji}^{n}(t)$ in (38), which contains two terms (one corresponding to x and the other to y), gives us the second inequality. The centered Poisson processes $\hat{Y}$ are written in an un-indexed manner to emphasize their independence, which gives us the (third) equality. Finally, last inequality is a result of applying Proposition A.1.

Applying Lemma A.2 (Gronwall's inequality) to (37), we get

$$\begin{aligned}
&\mathbf{P}\Big(\sup_{0\leq t\leq T} \big\|\big(\mathbf{x}^n(t), \mathbf{y}^n(t)\big) - \big(\mathbf{x}(t), \mathbf{y}(t)\big)\big\| \geq \epsilon e^{MT}\Big) \\
&\qquad\qquad \leq \mathbf{P}\Big(\sup_{0\leq t\leq T} \big(\sum_{i\in\mathcal{N}}\sum_{j\neq i} \frac{1}{n}\varepsilon_{ji}^{n}(t)\big)e^{MT} \geq \epsilon e^{MT}\Big) \\
&\qquad\qquad = \mathbf{P}\Big(\sup_{0\leq t\leq T} \sum_{i\in\mathcal{N}}\sum_{j\neq i} \frac{1}{n}\varepsilon_{ji}^{n}(t) \geq \epsilon\Big) \\
&\qquad\qquad \leq 2N(N-1)\exp\left(-n(1+\kappa)T \cdot h\Big(\frac{\epsilon}{2\sqrt{2}N(N-1)(1+\kappa)T}\Big)\right).
\end{aligned}$$

This can also be written as

$$\begin{aligned}
&\mathbf{P}\Big(\sup_{0\leq t\leq T} \big\|\big(\mathbf{x}^n(t), \mathbf{y}^n(t)\big) - \big(\mathbf{x}(t), \mathbf{y}(t)\big)\big\| \geq \epsilon\Big) \\
&\qquad \leq 4N(N-1)\exp\left(-n(1+\kappa)T \cdot h\Big(\frac{\epsilon e^{-MT}}{\sqrt{2}N(N-1)(1+\kappa)T}\Big)\right)
\end{aligned}$$

which is (15). Now observe that the above bound is finite and decreasing exponentially as n increases. Therefore,

$$\sum_{n=1}^{\infty} \mathbf{P}\Big(\sup_{0\leq t\leq T} \big\|\big(\mathbf{x}^n(t), \mathbf{y}^n(t)\big) - \big(\mathbf{x}(t), \mathbf{y}(t)\big)\big\| \geq \epsilon\Big) < \infty$$

and the almost sure convergence in (14) follows from the *Borel Cantelli Lemma.* This completes the proof of Theorem 3.4. □

B LYAPUNOV THEORY AND LASALLE INVARIANCE PRINCIPLE

In this section, we collect some definitions and results from the Lyapunov theory for non-linear systems. By the term *Flow* (or *Semi-flow*) with respect to some deterministic system, denoted by a function $\Phi : \mathbb{R} \times X \longrightarrow X$, we mean that $\Phi(x_0, t)$ gives the solution to that deterministic system at time t and starting at x_0 at time 0 (note that $t \geq 0$ always in case of semi-flows, while $t \in (-\infty, +\infty)$ for flows).

Often in literature, the term 'Lyapunov function' $V : X \longrightarrow \mathbb{R}$ is defined as a non-negative function, with $V(0) = 0$ and $d(V)/dt < 0$ for any $x \in X, x \neq 0$. This is useful for analyzing systems with only one fixed point that can be easily translated to the origin $0 \in X$ without loss of generality. However, our ODE system (18) consists of multiple fixed points. The results we state below are accordingly adjusted to cover such general cases.

Definition B.1. *(Section 3 in [4]) Consider a flow $\Phi : \mathbb{R} \times X \longrightarrow X$ contained within some set X. A point in $z \in X$ is a fixed point of the flow Φ if $\Phi(t, z) = z$ for all time $t \geq 0$. A function $V : X \longrightarrow \mathbb{R}$ is called a 'Lyapunov function' if*

(i) *V is continuous over X,*
(ii) *$V(\phi(t, x)) \leq V(x)$ for all $x \in X$ and $t \geq 0$ (negative semi-definiteness),*

(iii) *If $V(\psi(t)) = c$, where c is some constant, for some periodic oribit $\psi(t)$ for all $t \in R$, then $\psi(t)$ is actually some fixed point x of the flow ϕ.*

Such Lyapunov functions can be used to prove convergence of a flow induced by a system of ODEs using the following famous theorem.

Theorem B.2 (LaSalle invariance principle). *(Theorem 3.1 in [4], Chapter 5 in [52], Chapter 3 in [47]). Let $V : X \to \mathbb{R}$ be a 'Lyapunov function' for some set X and flow Φ. Let $\gamma^+(x)$ denote the forward (in time) orbit of the flow $\Phi(\cdot, x)$, i.e. starting at x. If $\gamma^+(x)$ is relatively compact, and all fixed points of Φ are isolated, then for all $x \in X$, $\Phi(t, x) \to z$ for some fixed point z.*

To help characterize fixed points of the system, we give the definitons of Lyapunov stability, and then state a theorem that helps characterize the stable and unstable fixed points in terms of the Lyapunov function $V : X \to \mathbb{R}$.

Definition B.3. *(Stable, unstable and asymptotically stable fixed points).*

(i) *A fixed point $z \in X$ of a flow Φ is 'stable' if for all $\epsilon > 0$, there exists a $\delta > 0$ such that for any $x \in X$, $\|x - z\| < \delta \implies \|\Phi(t, x) - z\| < \epsilon$ for all $t \geq 0$.*

(ii) *A fixed point $z \in X$ of a flow Φ is 'unstable' if it is not stable.*

(iii) *A fixed point $z \in X$ of a flow Φ is 'asymptotically stable' if it is stable and there exists a $\delta > 0$ such that $x \in X$, $\|x - z\| < \delta \implies \|\Phi(t, x) - z\| \to 0$ as $t \to \infty$.*

Theorem B.4. *(Theorems 4.1 and 4.2 in [4], Chapter 5 in [52], Chapter 4 in [47]). Let $z \in X$ be an isolated fixed point of flow Φ, and let $V : X \to \mathbb{R}$ be a Lyapunov function. Let $\gamma^+(x)$ be relatively compact for any $x \in X$ with $\gamma^+(x)$ bounded. Then,*

(i) *z is 'asymptotically stable' if there exists a $\delta > 0$ such that $V(x) > V(z)$ for any $x \in X$ where $\|x - z\| < \delta$, implying that z is a local minimizer of V.*

(ii) *z is 'unstable' if it is not a local minimizer of V, i.e. for any $\epsilon > 0$, there exists an $x \in X$ such that $\|x - z\| < \epsilon$ and $V(x) < V(z)$.*