Electronic Transactions on Numerical Analysis.
Volume 39, pp. 353-378, 2012.

ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

LOCALLY SUPPORTED EIGENVECTORS OF MATRICES ASSOCIATED WITH CONNECTED AND UNWEIGHTED POWER-LAW GRAPHS[*]

VAN EMDEN HENSON[†] AND GEOFFREY SANDERS[†]

Abstract. We identify a class of graph substructures that yields locally supported eigenvectors of matrices associated with unweighted and undirected graphs, such as the various types of graph Laplacians and adjacency matrices. We discuss how the detection of these substructures gives rise to an efficient calculation of the locally supported eigenvectors and how to exploit the sparsity of such eigenvectors to coarsen the graph into a (possibly) much smaller graph for calculations involving multiple eigenvectors. This preprocessing step introduces no spectral error and, for some graphs, may amount to considerable computational savings when computing any desired eigenpair. As an example, we discuss how these vectors are useful for estimating the commute time between any two vertices and bounding the error associated with approximations for some pairs of vertices.

Key words. graph Laplacian, adjacency matrix, eigenvectors, eigenvalues, sparse matrices

AMS subject classifications. 05C50, 05C82, 65F15, 65F50, 94C15

1. Introduction. Network scientists study complex systems from a diverse set of scientific fields [23]. Currently, there are great efforts in the study of communication networks, social networks, biological networks, chemical networks, etc., and the sizes of the networks being studied are growing exceptionally large. Such applications demand several computationally intense tasks such as clustering the network members in various ways [27], ranking network members by significance [6], measuring the distance between two network members [10], counting triangles between network members [25], or visualizing the topology of the network in some low-dimensional space [15]. One common approach to the study of a complex network is to model the network as a graph and use the mathematical properties of the graph to estimate a quantity of interest. This approach often leads to *spectral graph calculations*, where one or more eigenvalues or eigenvectors of a matrix associated with the graph are approximated by an eigensolver. The approximate eigenpairs are then used to estimate the quantities of interest.

An attractive advantage of the spectral approach is that one common linear algebra tool, the eigensolver, may be employed to address a wide class of complex network calculations. Another advantage is that the spectrum is a tool for quantifying errors incurred when simplifying graphs. However, there are many challenges behind designing eigensolvers that are efficient for all these calculations due to the variety of graph topologies present in large, real-world networks and the many types of eigenproblems that various calculations require. Our primary goal is to enhance existing eigensolvers so that they are more efficient and thus more useful for network science applications. A secondary goal is to use spectral properties to simplify the graph with only little change in the eigenvectors to potentially increase the efficiency of non-spectral methods as well.

Many real-world networks of interest have a large *periphery*, where many vertices of very low *degree* (number of connections) are present. Moreover, the graph topology often has a *power-law degree distribution*, meaning that the number of vertices with a given degree is approximately proportional to some negative power of the degree (cf. [23] and Figure 1.2). This type of topology is highly challenging for large-scale iterative eigensolvers, such as those

[*]Received January 18, 2012. Accepted July 16, 2012. Published online on November 5, 2012. Recommended by M. Benzi. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

[†]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Box 808, L-561, Livermore, CA 94551-0808 ({henson5, sanders29}@llnl.gov).

available in many of the common high-performance computing packages [12, 13, 17]. Many of the preconditioning techniques that are highly successful for *mesh-like graphs* (e.g., graphs coming from local discretizations of PDEs) are less successful for power-law graphs due to the limitations posed by the communication properties of distributive computing with matrices having some high degree vertices. Any graph simplification, such as graph *coarsening*, that introduces little spectral error will improve this situation. Our long term research interests are to further develop the techniques in [8, 14, 16] to better address power-law graphs. These techniques automatically generate multilevel hierarchies from which accurate approximations to eigenpairs are calculated.

This paper describes a technique that aids spectral calculations by taking advantage of specific types of local substructures that are often present throughout the periphery of unweighted and undirected graphs in real-world networks. We give a few examples in Figure 1.1. The existence of such substructures implies that the matrix has eigenvectors that are nonzero only within the local substructure. Typically, the associated eigenvalues have extraordinarily high multiplicity (for some classes of graphs, the multiplicity can be $\mathcal{O}(n)$). We adopt the terminology of functional analysis (where the *support* of a function means the subset of the function's domain where the function is nonzero) and call the eigenvectors that are nonzero only on a local substructure *locally supported eigenvectors* (LSEVs). Detecting substructures that supports LSEVs allows one to efficiently calculate some parts of the spectrum (and the associated LSEVs) and to reduce the size of the graph without any loss of accuracy for the rest of the spectrum (and the associated eigenvectors). When such substructures occur many times in a graph, this technique provides significant computational savings in the calculation of the other eigenvectors that have not been identified as LSEVs.

This paper is organized in the following manner. In the remainder of Section 1, we put this work into historical context, relating it to previous work on eigenpairs arising from special structures in the graph. We also provide a connection to and motivation from the field of multigrid, and then describe the matrices of interest and the eigenproblems that arise from them. In Section 2, we define LSEVs and establish algebraic conditions that yield LSEVs. We develop theoretical results regarding the existence of LSEVs of matrices associated with graphs and show the relation of the spectra of the original matrix with the coarsened version that results after removing the structures generating the LSEVs. We give several examples of classes of LSEVs that are common in real-world graphs. We present results that demonstrate how the knowledge of LSEVs is applied to the calculation of commute time in Section 3. In Section 4, we demonstrate the application of our theory to several graphs, both synthetic and real. We show that in some graphs, the LSEVs make up a substantial portion of the spectrum and further that they can be used to create a coarsened graph having substantially fewer vertices and edges giving rise to a more tractable problem of computing the remaining (non-LSEV) spectral components. We give concluding remarks and a statement regarding further work in Section 5. Finally, we present a pseudocode for the algorithms used to detect LSEVs in Appendix A.

1.1. Background. We note that some observations related to this work have been discussed previously in the spectral graph theory literature. At least as far back as the mid 1980's, Faria [9] was aware that graphs with many vertices that are exclusively connected to the same vertex give rise to eigenvalues of high multiplicity for adjacency matrices and graph Laplacians. Eigenvectors that are positive at one vertex, negative at another vertex, and zero everywhere else are referred to as *Faria vectors*, and their properties are described in [2, 20, 22]. These papers concentrate on the multiplicity of the eigenvalues associated with these eigenvectors. Also, Faria vectors are used as counterexamples to lower bounds on the number of nodal domains that the sign structure of eigenvectors induces [5]. Recently,

ETNA
Kent State University
http://etna.math.kent.edu

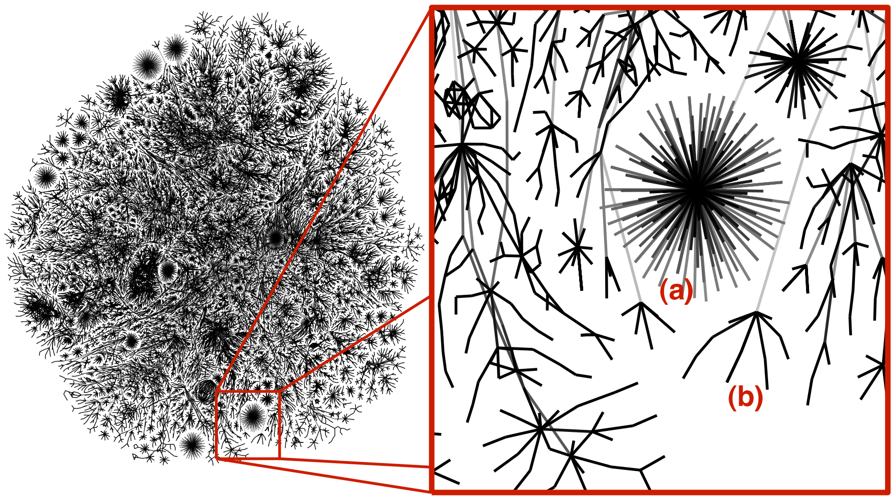


FIG. 1.1. *A visualization of the* `opte1` *graph from [19] (see Section 2 for a description). The detail on the right displays two common types of substructures in the periphery of this graph that can be exploited for more efficient eigenvalue computations: (a) many vertices that are exclusively connected to the same vertex and (b) chains of the same length connected to the same vertex.*

eigenspaces of extraordinarily high multiplicity were observed in many power-law graphs of large, real-world networks [11]. The connection of considerable portions of these eigenspaces with LSEVs is verified.

In this work, we emphasize the computational importance of the sparsity of the LSEVs and demonstrate that these vectors can be used to efficiently reduce the original graph into a smaller graph that perfectly represents all other eigenvectors that have not been identified as LSEVs. We generalize the concept of a Faria vector by giving algebraic conditions that fully classify all types of substructures that admit LSEVs. We describe a few substructure families that support local eigenvectors which are common in real-world graphs and provide algorithms to detect various types of them.

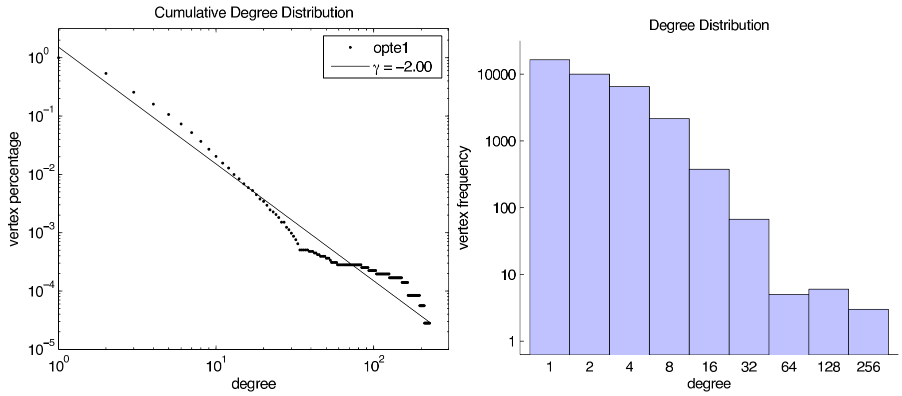


FIG. 1.2. *Two common types of degree distribution plots [23] for the* `opte1` *graph [19].*

The process of detecting locally supported eigenvectors can be thought of as a specialized version of *aggregation* or combining localized groups of vertices (*aggregates*) from within a graph. Aggregation multigrid [26] is a class of coarsening approaches where aggregates are

ETNA
Kent State University
http://etna.math.kent.edu

formed, and the number of degrees of freedom in each aggregate is reduced so that a certain portion of the operator's spectrum is well represented by a smaller matrix. These methods are typically used to build multilevel solvers for linear systems or eigensystems and can be quite effective for problems posed on mesh-like graphs (for example, for a problem that is a discretization of an elliptic PDE in a low-dimensional space) [1].

1.2. Graph associated matrices and common eigenproblems. A *graph* $\mathcal{G}(\mathcal{V},\mathcal{E})$ is a collection of n *vertices*, $\mathcal{V} = \{1,\dots,n\}$, and relationships between pairs of vertices, or *edges* $\mathcal{E}$. If there exists an edge between two vertices i and j, then $(i,j) \in \mathcal{E}$. Here, we look at a specific class of graphs. Assume that (i) $\mathcal{G}$ is *simple*: it contains no *self loops*, $(i,i) \notin \mathcal{E}$, and no *multiple edges*, (ii) $\mathcal{G}$ is *undirected*: $(i,j) \in \mathcal{E}$ if and only if $(j,i) \in \mathcal{E}$, (iii) $\mathcal{G}$ is *unweighted*: the distance or cost of each edge in $\mathcal{E}$ is the same, and (iv) that the graph is *connected*, meaning that there exists at least one path between any two vertices in the graph. The *degree* of a vertex i, written d_i, is the number of edges that share i.

We recall several commonly used matrices associated with a graph.

DEFINITION 1.1. *The structure of graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ is used to define several useful matrices.*

(i) Adjacency matrix: let $A \in \mathbb{R}^{n\times n}$ with entries given by

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}.$$

(ii) Degree matrix: let D be a diagonal matrix in $\mathbb{R}^{n\times n}$ such that $D_{ii} = d_i$.

(iii) Combinatorial graph Laplacian: let $L = D - A$, or

$$L_{ij} := \begin{cases} d_i & \text{if } i = j \\ -1 & \text{if } (i,j) \in \mathcal{E} \end{cases}.$$

(iv) Normalized graph Laplacian: let $\hat{L} = I - D^{-1/2}AD^{-1/2}$, or

$$(\hat{L})_{ij} := \begin{cases} 1 & \text{if } i = j \\ -\frac{1}{\sqrt{d_i d_j}} & \text{if } (i,j) \in \mathcal{E} \end{cases}.$$

(v) Signless graph Laplacian: $|L| = D + A$.

The spectrum and the associated eigenvectors of the adjacency matrix and various graph Laplacian matrices are all of interest. However, to be concise, we only explicitly describe LSEVs for eigenproblems associated with the combinatorial graph Laplacian and an associated application. We include a few remarks regarding the other types of graph-associated matrices to emphasize that LSEVs apply to a broader class of problems than we describe.

A (normalized) eigenpair $(\mathbf{v}_k, \lambda_k)$ of L is a nontrivial *eigenvector* $\mathbf{v}_k \in \mathbb{R}^n$ and a scalar *eigenvalue* λ_k that satisfy

$$L\mathbf{v}_k = \lambda_k \mathbf{v}_k. \tag{1.1}$$

The properties of L offer several simplifications to (1.1). Because $\mathcal{G}$ is undirected, L is symmetric, $L = L^t$, which implies that the eigenvalues λ_k are all real and there exists a complete set of n orthogonal eigenvectors, $\mathbf{v}_k^t\mathbf{v}_l = \delta_{kl}$, where $\delta_{kl} = 1$ for $k = l$, and 0 for $k \neq l$. Gershgorin's theorem implies that the spectrum is non-negative, $\lambda_k \in [0, 2\max_i d_i]$. Let $\mathbf{1}$ and $\mathbf{0}$ be the vector of all ones and all zeros, respectively, and note that the definition of the graph Laplacian implies $L\mathbf{1} = \mathbf{0}$ such that $(\mathbf{v}_1, \lambda_1) = (\mathbf{1}, 0)$ is a known eigenpair. The assumption that the graph is connected implies that the multiplicity of $\lambda_1 = 0$ is 1. We order the eigenvalues in increasing order,

$$0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \ldots \leq \lambda_{n-1} \leq \lambda_n \leq 2 \max_i d_i. \tag{1.2}$$

For large n, it is computationally overwhelming to calculate all eigenpairs. For the eigenproblem (1.1), the eigenpairs associated with the K lowest eigenvectors are typically sought. Our computational task is to approximate solutions to

$$\begin{array}{rcl} L\mathbf{v}_k & = & \lambda_k \mathbf{v}_k \\ \mathbf{v}_k^t \mathbf{v}_l & = & \delta_{kl} \end{array} \qquad \text{for } k, l = 2, 3, \ldots, K.$$

We note that a vector-scalar pair is an eigenpair if and only if the pair has a zero *eigenresidual*, $(L - \lambda_k I)\mathbf{v}_k = \mathbf{0}$. For an approximate eigenpair $(\mathbf{x}, \mu)$, the size of the eigenresidual is used to gauge the accuracy of the approximation. See [24] for standard results regarding the connection between the size of $\|(L-\mu I)\mathbf{x}\|$ and the quality of the approximations $\mu \approx \lambda_k$ and $\mathbf{x} \approx \mathbf{v}_k$. In this work, we employ eigenresiduals as a theoretical tool to demonstrate the accuracy of estimating eigenvectors on graphs coarsened using the knowledge of LSEVs.

2. Locally suported eigenvectors (LSEVs).

Here we give the algebraic conditions for a portion of a graph having locally supported eigenvectors. First, we define a few concepts and then proceed with our main observations.

DEFINITION 2.1. *A subset of vertices $\mathcal{S} \subset \mathcal{V}$ is* connected *if for every pair of vertices i and $j \in \mathcal{U}$ there exists a path of vertices in $\mathcal{S}$ from i to j. Let the* dilation *of $\mathcal{S}$ be defined as*

$$\textit{dilate}(\mathcal{S}) := \mathcal{S} \cup \{i \in \mathcal{V} \,:\, (i,j) \in \mathcal{E} \textit{ for some } j \in \mathcal{S}\}.$$

We say $\mathcal{S}$ is nearly-connected *if dilate$(\mathcal{S})$ is connected but $\mathcal{S}$ is not connected.*

DEFINITION 2.2. *Let $\mathcal{S}(\mathbf{x})$ denote the* support *of a vector $\mathbf{x} \in \mathbb{R}^n$,*

$$\mathcal{S}(\mathbf{x}) := \{i \in \mathcal{V} \,:\, x_i \neq 0\}.$$

We say the support of $\mathbf{x}$ is local *if $\mathcal{S}(\mathbf{x})$ is connected or nearly-connected and contains a small number of vertices (much less than n).*

DEFINITION 2.3. *Assume $\mathbf{v}$ is an eigenvector of L. If $\mathcal{S}(\mathbf{v})$ is local, then we say $\mathbf{v}$ is a* locally supported eigenvector (LSEV) *of L.*

Let $\mathcal{S}$ be a small local subset of the vertices in $\mathcal{V}$. We decompose L in the following way. Organize all the vertices into an ordering with $\mathcal{S}$ first, $\{\mathcal{S}, \mathcal{V} \setminus \mathcal{S}\}$, and then write

$$L = \left[\begin{array}{c|c} L_{11} & L_{12} \\ \hline L_{21} & L_{22} \end{array}\right], \quad \text{where} \quad \left\{\begin{array}{l} L_{11} \text{ represents edges within } \mathcal{S}, \\ L_{12} \text{ represents edges from } \mathcal{S} \text{ to } \mathcal{V} \setminus \mathcal{S}, \\ L_{22} \text{ represents edges within } \mathcal{V} \setminus \mathcal{S}. \end{array}\right. \tag{2.1}$$

Note that the undirected edges in $\mathcal{G}$ imply that $L^t = L$, so we have $L_{11}^t = L_{11}$, $L_{22}^t = L_{22}$, and $L_{21} = L_{12}^t$. We state necessary and sufficient conditions for a LSEV to exist for the set $\mathcal{S}$.

THEOREM 2.4 (LSEV existence). *Let a matrix $L \in \mathbb{R}^{n\times n}$ be decomposed as in (2.1) with respect to a subset $\mathcal{S}$. Let $\mathbf{u}$ be a nonzero vector in $\mathbb{R}^{|\mathcal{S}|}$. A local subset $\mathcal{S}$ contains a locally supported eigenvector $\left[\mathbf{u}^t, \mathbf{0}^t_{|\mathcal{V}\setminus\mathcal{S}|}\right]^t$ of L corresponding to an eigenvalue λ if and only if*

$$L_{11}\mathbf{u} = \lambda \mathbf{u} \qquad \textit{and} \qquad L_{21}\mathbf{u} = \mathbf{0}_{|\mathcal{V}\setminus\mathcal{S}|}. \tag{2.2}$$

Proof. This result follows by applying L (via (2.1)) to $\mathbf{v} = \left[\mathbf{u}^t, \mathbf{0}^t_{|\mathcal{V}\setminus\mathcal{S}|}\right]^t$ and showing that (2.2) is equivalent to $L\mathbf{v} = \lambda\mathbf{v}$. ☐

We point out that (2.2) is a smaller eigenproblem of size $|\mathcal{S}|$ with a set of additional linear constraints imposed. There is an extra constraint for each linearly independent row of L_{21}. The fact that the graph is connected gives $L_{21} \neq O$ and at least one extra constraint. For general $\mathcal{S} \subset \mathcal{V}$, there is no guarantee that a solution $\mathbf{u}$ exists. However, there are several types of graph substructures common to many real-world networks such that the kernel of L_{21} does contain eigenvectors of L_{11}; see [9, 20, 22] and we discuss a few more general types of these in the next few sections. A general approach to test a set $\mathcal{S}$ for LSEVs is to fully solve $L_{11}\mathbf{u} = \lambda\mathbf{u}$ and check each eigenspace to see if it contains vectors that satisfy $L_{21}\mathbf{u} = \mathbf{0}$. Clearly, inspecting all vertex sets of a certain size or below is computationally intractable. We provide a few algorithms in Appendix A that detect specific types of substructures and give a framework for algorithms that detect more general types.

For now, we assume that we are able to identify a collection of R *disjoint* (non-overlapping) subsets of vertices, $\{\mathcal{S}_r\}_{r=1}^R$, that each have $M_r \geq 1$ orthogonal LSEVs. For each r, define some permutation $\Pi^{(r)}$ of $\mathcal{V}$ such that $(\Pi^{(r)})^t$ orders the vertices in $\mathcal{S}_r$ first. Then, let $L_{11}^{(r)}$ and $L_{21}^{(r)}$ correspond to the decomposition (2.1) of $(\Pi^{(r)})^t L \Pi^{(r)}$ (where the columns and rows corresponding to $\mathcal{S}_r$ appear first). Also, define an injection of $\mathbb{R}^{|\mathcal{S}_r|}$ into $\mathbb{R}^n$ as $P^{(r)} := [I_{|\mathcal{S}_r|}, O_{|\mathcal{S}_r|\times|\mathcal{V}\setminus\mathcal{S}_r|}]^t$. Then, denote the local portion of the M_r eigenvectors that are supported in $\mathcal{S}_r$ by $\{\mathbf{u}_m^{(r)}\}_{m=1}^{M_r}$. Each $\mathbf{u}_m^{(r)}$ is an independent eigenvector of $L_{11}^{(r)}$ such that $L_{21}^{(r)}\mathbf{u}_m^{(r)} = \mathbf{0}$. We can collect each locally supported eigenvector into a sparse matrix

$$Z = \left[\Pi^{(1)}P^{(1)}\mathbf{u}_1^{(1)}, \ldots, \Pi^{(1)}P^{(1)}\mathbf{u}_{M_1}^{(1)}, \Pi^{(2)}P^{(2)}\mathbf{u}_1^{(2)}, \ldots, \Pi^{(R)}P^{(R)}\mathbf{u}_{M_R}^{(R)}\right]. \tag{2.3}$$

2.1. Partitioning of spectra. Now we consider how to use the enumeration of LSEVs in Z to calculate the other eigenvectors of L more efficiently. The matrix L is symmetric and therefore has a complete orthogonal basis of eigenvectors. The eigenvectors in the columns of Z are all orthogonal to the injections of eigenvectors of $L_{11}^{(r)}$ that are not in the kernel of $L_{21}^{(r)}$. Additionally, the support of each eigenvector in Z is entirely contained in one of the sets $\mathcal{S}_r$. Therefore, it is inexpensive to compute a sparse *interpolation* matrix Q that completely spans the orthogonal complement to Range(Z) containing all eigenvectors not in the columns of Z. This is accomplished by collecting the eigenvectors of $L_{11}^{(r)}$ that are not in the kernel of $L_{21}^{(r)}$ for $r = 1, \ldots, R$ and adding the columns of the $n \times n$ identity matrix associated with vertices that are not contained in $\bigcup_{r=1}^R \mathcal{S}_r$.

$$Q = \begin{bmatrix} Z^\perp & O \\ O & I \end{bmatrix} \quad \begin{array}{ll} \longleftarrow & \text{sparse basis orthogonal to LSEVs, restricted to } \bigcup_{r=1}^R \mathcal{S}_r \\ \longleftarrow & \text{identity operator on } \mathcal{V} \setminus \{\bigcup_{r=1}^R \mathcal{S}_r\}. \end{array}$$

Let $n_c = n - \sum_{r=1}^R M_r$ and define an aggregation to contain all the groups we have identified with locally supported eigenvectors, $\{\mathcal{S}_r\}_{r=1}^R$, and singletons of vertices, $\{s\}$, that are not present in any of these groups

$$\mathcal{A}_r = \begin{cases} \mathcal{S}_r & \text{for } r = 1, \ldots, R \\ \{s\} & \text{for } r = R+1, \ldots, R + (n - \sum_{r=1}^R |\mathcal{S}_r|) \end{cases}.$$

An $n \times n_c$ binary aggregation matrix is given by

$$W_{ij} = \begin{cases} 1 & \text{if vertex } i \in \mathcal{A}_j \\ 0 & \text{if vertex } i \notin \mathcal{A}_j \end{cases}.$$

The matrix W serves as a template for Q, which has a similar block structure. The local vectors that are orthogonal to locally supported eigenvectors (eigenvectors of L_{11} that are in the range of L_{21}) are injected into the structure of W. For example, say we have identified two sets that contain locally supported eigenvectors (one with three vertices and two locally supported eigenvectors, the other with four vertices and two locally supported eigenvectors). Then Q has the same block structure as W with one column for the three vertex set and two columns for the four vertex set. Letting $\times$ denote a (possibly) nonzero entry, then

$$W = \begin{bmatrix} \ddots & & & \\ & 1 & & \\ & 1 & & \\ & 1 & & \\ & & 1 & \\ & & 1 & \\ & & 1 & \\ & & 1 & \\ & & & \ddots \end{bmatrix} \qquad \longrightarrow \qquad Q = \begin{bmatrix} \ddots & & & & \\ & \times & & & \\ & \times & & & \\ & \times & & & \\ & & \times & \times & \\ & & \times & \times & \\ & & \times & \times & \\ & & \times & \times & \\ & & & & \ddots \end{bmatrix}.$$

The columns of the matrices $Z \in \mathbb{R}^{n\times(n-n_c)}$ and $Q \in \mathbb{R}^{n\times n_c}$ form a complete orthogonal decomposition of $\mathbb{R}^n$. Additionally, the columns of Z are all eigenvectors of L. In order to compute eigenvectors that have not been collected into Z, we make use of a *coarsened* $n_c \times n_c$ matrix

$$L_c := Q^t L Q.$$

In the remainder of this section, we present a result that shows that all eigenvectors of L not collected into Z are obtained by solving for eigenvectors of L_c and mapping them back into $\mathbb{R}^n$ using Q. Additionally, we show that accurate approximations to the eigenvectors of L_c interpolate to approximations to the eigenvectors of L without a loss of accuracy. First we state several properties of the matrices involved.

LEMMA 2.5. *Given Z and Q as described above, then we have*

(i) $Q^tQ = I_{n_c}$ *(ii)* $Z^tZ = I_{n-n_c}$ *(iii)* $I_n = QQ^t + ZZ^t$
(iv) $Z^tQ = O_{(n-n_c)\times n_c}$ *(v)* $Z^tLQ = O_{(n-n_c)\times n_c}$.

Proof. Because the columns of $[Z, Q]$ are orthonormal in $\mathbb{R}^n$, (i), (ii), and (iv) hold. Also, because the columns of Z are eigenvectors, we have Range$(LZ) \subset$ Range(Z), implying that $Q^tLZ = O_{(n-n_c)\times n_c}$. Thus, (v) holds by the symmetry of L. Finally, we have the orthogonal decomposition (iii) due to the completeness of the basis. ∎

Now we show that all eigenvectors not collected into Z are perfectly represented by the eigenvectors of the coarser matrix L_c.

THEOREM 2.6 (Coarse graph spectral representation). *If $\{(\mathbf{v}_k, \lambda_k)\}_{k=1}^{n_c}$ are all eigenpairs not enumerated in Z then $\{(Q^t\mathbf{v}_k, \lambda_k)\}_{k=1}^{n_c}$ is the complete set of eigenpairs of L_c.*

Proof. Let $(\mathbf{v}, \lambda)$ be an eigenpair of L that is not in span(Z). Employing Lemma 2.5 (i),

ETNA
Kent State University
http://etna.math.kent.edu

(iii), and (v) gives

$$\begin{aligned}
(L-\lambda I_n)\mathbf{v} &= \mathbf{0}_n \\
Q^t(L-\lambda I_n)(QQ^t+ZZ^t)\mathbf{v} &= Q^t\mathbf{0}_n && \text{by Lemma 2.5 (iii)} \\
Q^t(L-\lambda I_n)(QQ^t)\mathbf{v} &= \mathbf{0}_{n_c} && \text{by Lemma 2.5 (v)} \\
Q^t(LQ-\lambda Q)Q^t\mathbf{v} &= \mathbf{0}_{n_c} \\
(Q^tLQ-\lambda Q^tQ)Q^t\mathbf{v} &= \mathbf{0}_{n_c} \\
(L_c-\lambda I_{n_c})Q^t\mathbf{v} &= \mathbf{0}_{n_c} && \text{by Lemma 2.5 (i).}
\end{aligned}$$

Therefore, $(Q^t\mathbf{v},\lambda)$ is an eigenpair for L_c. Additionally, let $\mathbf{v}$ and $\mathbf{w}$ be any two orthogonal eigenvectors of L that are not in span(Z). Due to the orthogonality of the eigenspaces of $L=L^t$, we have $Z^t\mathbf{w}=Z^t\mathbf{v}=\mathbf{0}_{n_c}$. Using this fact and Lemma 2.5 (iii) gives

$$\left\langle Q^t\mathbf{v},Q^t\mathbf{w}\right\rangle = \left\langle QQ^t\mathbf{v},\mathbf{w}\right\rangle = \left\langle (ZZ^t+QQ^t)\mathbf{v},\mathbf{w}\right\rangle = \langle \mathbf{v},\mathbf{w}\rangle = 0.$$

The completeness of the eigenbasis of L gives n_c orthogonal eigenvectors not enumerated in Z and therefore $\{(Q^t\mathbf{v}_k,\lambda_k)\}_{k=1}^{n_c}$ is a complete set of eigenpairs of L_c. ∎

COROLLARY 2.7. *If $\{(\mathbf{x}_k,\lambda_k)\}_{k=1}^{n_c}$ is a complete set of eigenpairs of the matrix L_c, then $\{(Q\mathbf{x}_k,\lambda_k)\}_{k=1}^{n_c}$ are all eigenpairs not enumerated in Z.*

Proof. Consider any two orthogonal eigenvectors of L_c paired with their eigenvalues, $(\mathbf{x}_k,\lambda_k)$ and $(\mathbf{x}_l,\lambda_l)$ for $k\neq l$. First we show that $Q\mathbf{x}_k$ is an eigenvectors of L corresponding to λ_k using Lemma 2.5 (iii) and (v),

$$LQ\mathbf{x}_k = (ZZ^t+QQ^t)LQ\mathbf{x}_k = Q(Q^tLQ\mathbf{x}_k) = \lambda_k Q\mathbf{x}_k.$$

Next we show that $Q\mathbf{x}_k$ and $Q\mathbf{x}_l$ are orthogonal,

$$\langle Q\mathbf{x}_k,Q\mathbf{x}_l\rangle = \left\langle Q^tQ\mathbf{x}_k,\mathbf{x}_l\right\rangle = \langle \mathbf{x}_k,\mathbf{x}_l\rangle = 0.$$

Thus, $\{(Q\mathbf{x}_k,\lambda_k)\}_{k=1}^{n_c}$is a set of n_c orthogonal eigenvectors of L, which must be all eigenpairs not enumerated in $Z\in\mathbb{R}^{n\times(n-n_c)}$ by a counting argument. ∎

This result shows that we can obtain any eigenvector not represented in Z by an eigensolve involving L_c. We have effectively reduced the problem to a coarser graph without loss of accuracy. The following result shows that a computation of the eigenmodes spanned by Q does not have to return to the original graph (until the modes themselves are needed) because coarse eigenresidual error measures are equal to the original eigenresidual error measures of the interpolated approximations.

THEOREM 2.8. *Let $\mathbf{w}_c$ be a vector in $\mathbb{R}^{n_c}$. For approximate eigenpairs of L of the form $(Q\mathbf{w}_c,\mu)$, we have*

$$\frac{\|(L-\mu I_n)Q\mathbf{w}_c\|}{\|Q\mathbf{w}_c\|} = \frac{\|(L_c-\mu I_{n_c})\mathbf{w}_c\|}{\|\mathbf{w}_c\|} \quad .$$

Proof. For the denominator, $Q^tQ=I$ implies $\|Q\mathbf{w}_c\|=\|\mathbf{w}_c\|$. For the numerator, we use similar techniques as in the previous theorem to show that

$$\begin{aligned}
\|(L-\mu I)Q\mathbf{w}_c\|^2 &= \|QQ^t(L-\mu I)Q\mathbf{w}_c\|^2 + \|ZZ^t(L-\mu I)Q\mathbf{w}_c\|^2 \\
&= \|Q(L_c-\mu I_{n_c})\mathbf{w}_c\|^2 + \|Z^tLQ\mathbf{w}_c - \mu Z^tQ\mathbf{w}_c\|^2 \\
&= \|(L_c-\mu I_{n_c})\mathbf{w}_c\|^2. \qquad ∎
\end{aligned}$$

REMARK 2.9 (Several observations regarding L_c). It is important to realize that, in general, the graph associated with the matrix L_c may not have some of the graph properties that the original graph $\mathcal{G}$ enjoys. The nonzero, off-diagonal entries are no longer all -1's. Depending on the choice of the basis in Q or the type of LSEVs in question, some of the off-diagonal entries may be positive. Because many of the low-degree vertices have been removed from the graph, a power-law is often not retained for the coarser graph. However, the important algebraic properties of the matrix L are kept for L_c, such as symmetry and the preservation of eigenvalues that are not associated with the LSEVs that have been detected.

Due to the lack of a perfect hierarchical structure in real-world graphs, L_c typically has few LSEVs and we have observed little computational advantage in applying LSEV-based coarsening recursively. However, if a graph of interest is expected to have a repeated hierarchical symmetry, then this coarsening process should be repeated as well.

2.2. The simplest example: Faria's shared leaves. This section discusses Faria's example [9] of locally supported eigenvectors in detail.

EXAMPLE 2.10 (Faria's shared leaves). Assume that $\mathcal{G}$ has the following substructure: there are some vertices that have only one connection (called *leaves*) and some of these leaves are connected to the same vertex (their *parent*). Note that the graph theory community often calls leaves *pendants* and parents *quasipendants*; see Figure 2.1 for two examples of this type of substructure. We demonstrate that this substructure admits locally supported eigenvectors.

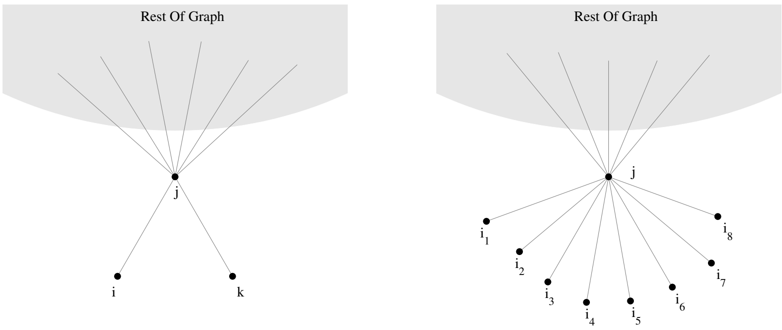


FIG. 2.1. *Leaves that share a parent. Left: two leaves i, k have parent j. Right: parent j has q leaves. Edges from j into the rest of the graph are depicted by lines into the shaded regions.*

For any eigenpair $(\mathbf{v}, \lambda)$ of L, we have the system of equations $(L-\lambda I)\mathbf{v} = \mathbf{0}$. Consider the simplest case first. Assume we have a parent j with two child leaves i and k (see the left-hand side of Figure 2.1). The i-th equation of $(L - \lambda I)\mathbf{v} = \mathbf{0}$ is

$$(1-\lambda)v_i - v_j = 0. \tag{2.4}$$

Similarly, the k-th equation is

$$(1-\lambda)v_k - v_j = 0. \tag{2.5}$$

Assume for the moment that $\lambda = 1$ is an eigenvalue. To satisfy (2.4), we see that $v_j = 0$ is necessary, in turn implying that (2.4) and (2.5) are both automatically satisfied for any values of v_i and v_k. If we choose these values so that the j-th equation is also satisfied, then we have an eigenvector associated with the eigenvalue $\lambda = 1$ that is nonzero only on i and k. The j-th equation is

$$(d_j - \lambda)v_j - \sum_{p\in\mathcal{N}_j} v_p = 0, \tag{2.6}$$

where d_j is the degree of j and $\mathcal{N}_j$ is the set of vertices connected to j (excluding j itself). Using $v_j = 0$ reduces this constraint to $\sum_{p\in\mathcal{N}_j} v_p = 0$. Because $i,k \in \mathcal{N}_j$ for the *Faria vector*

$$\mathbf{v} = \begin{array}{c} [\\ \end{array} \begin{array}{cccccc} -1 & 0 & 1 & 0 & \cdots & 0 \\ i & j & k & & \text{else} & \end{array} \begin{array}{c}]^t, \\ \end{array}$$

we have $L\mathbf{v} - \mathbf{v} = \mathbf{0}$, i.e., $(\mathbf{v}, 1)$ is an eigenpair for L. Note that all other equations are satisfied because $\mathbf{v}$ is zero over all variables that are involved in these equations. It is easily verified that the conditions of Theorem 2.4 are satisfied by $\mathcal{S} = \{i,k\}$ and $\mathbf{u} = [-1, 1]^t$.

Now assume j is connected to q leaves $\mathcal{S} = \{i_1, i_2, \ldots, i_q\} \subset \mathcal{N}_j$ as on the right-hand side of Figure 2.1. For each leaf in $\mathcal{S}$, we have an equation similar to (2.4) that is automatically satisfied for $\lambda = 1$ and $v_j = 0$ independent of the value of $\mathbf{v}$ on the leaf. Following the same argument as above, any vector that satisfies Equation (2.6) and is nonzero only at the leaves connected to j is an eigenvector associated with $\lambda = 1$. In terms of the decomposition in (2.1), $L_{11} = I$, and $L_{21} = [-\mathbf{1}, O]^t$. Restricting $\mathcal{S}$ to $\mathbb{R}^q$, we have an orthonormal basis $\{\mathbf{u}_p\}$, for $p = 1, 2, \ldots, (q-1)$. The i-th entry of the p-th vector is given by

$$(\mathbf{u}_p)_i = \begin{cases} c_{q,p}\cos\left(\frac{(p+1)\pi i}{q}\right) & \text{if } p \text{ is odd} \\ c_{q,p}\sin\left(\frac{p\pi i}{q}\right) & \text{if } p \text{ is even} \end{cases}, \qquad i = 1, 2, \ldots, q, \tag{2.7}$$

where the normalization constants are $c_{q,p} = \sqrt{2/q}$ (except for the special case when q is even and $p = q-1$, then $c_{q,p} = 1/\sqrt{q}$). Invoke Theorem 2.4 to show that these $(q-1)$ vectors are locally supported eigenvectors,

$$L = \left[\begin{array}{c|cc} I & -\mathbf{1} & O \\ \hline -\mathbf{1}^t & d_j & \cdots \\ O & \vdots & \ddots \end{array}\right], \quad I\mathbf{u}_p = 1\cdot\mathbf{u}_p, \quad \text{and} \quad \begin{bmatrix} -\mathbf{1}^t \\ O \end{bmatrix}\mathbf{u}_p = \mathbf{0}.$$

Thus, there are $(q-1)$ orthogonal eigenvectors that are locally supported by $\mathcal{S}$ corresponding to the eigenvalue $\lambda = 1$. The only eigenvector of $L_{11} = I$ that is not in the kernel of L_{21} is the constant vector. Furthermore, these LSEVs give a lower bound on the *multiplicity* of the eigenvalue $\lambda = 1$, denoted $\text{mult}(\lambda = 1, \sigma(L))$.

PROPOSITION 2.11 (Faria's star degree [9]). *Let $\mathcal{P}$ be the set of nodes connected to 2 or more leaves. For any $r \in \mathcal{P}$, let q_r be the number of leaves connected to r, and collect these leaves into a set $\mathcal{S}_r$. Repeating the above argument yields*

$$\textit{mult}(\lambda = 1, \sigma(L)) \geq \sum_{j\in\mathcal{P}} (q_j - 1). \tag{2.8}$$

Let Z_j be the $n \times (q_j - 1)$ matrix whose p-th column represents the values of $\mathbf{u}_p^{(r)}$ injected into $\mathbb{R}^n$ over the q_r leaves. Then the matrix

$$Z = [Z_1, Z_2, \ldots, Z_r, \ldots]$$

ETNA
Kent State University
http://etna.math.kent.edu

gives a (possibly partial) orthogonal and sparse decomposition of the eigenspace of L associated with $\lambda = 1$.

REMARK 2.12. Note that it is possible to have a graph with $\lambda = 1$ having larger multiplicity than the bound given in Equation (2.8). Consider the following 8×8 example of this:

$$L = \begin{bmatrix} 1 & & -1 & & & & & \\ & 1 & -1 & & & & & \\ -1 & -1 & 3 & -1 & & & & \\ & & -1 & 2 & -1 & & & \\ & & & -1 & 2 & -1 & & \\ & & & & -1 & 3 & -1 & -1 \\ & & & & & -1 & 1 & \\ & & & & & -1 & & 1 \end{bmatrix}, \left\{ \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \\ -2 \\ -2 \\ 0 \\ 1 \\ 1 \end{bmatrix} \right\}. \tag{2.9}$$

The vectors on the right form a complete set of independent eigenvectors corresponding to $\lambda = 1$. Our counting of the locally supported basis vectors gives us the first 2 vectors. The third vector is an additional, nonlocal vector. Related examples are given in [9, 20].

FIG. 2.2. *Graph corresponding to Equation (2.9).*

We continue to describe the shared-leaves example by describing the process of obtaining a coarser graph from identifying shared leaves. The matrix Q is determined by collecting the local eigenvectors of $L_{11}^{(r)}$ that are not in the kernel of $L_{21}^{(r)}$ and adding columns of the identity matrix corresponding to the vertices that are not in any of the sets $\mathcal{S}_r$. For $\lambda \neq 1$ it is immediately evident that vectors that are constant over each $\mathcal{S}_r$ should be included into Q. Reconsidering Equations (2.4) and (2.5), we see that

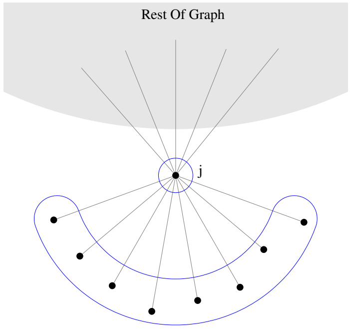
$$v_i = \frac{v_j}{1 - \lambda} = v_k$$

for any two leaves i, k with the same parent j. The eigenvectors that are not enumerated in Z and have $\lambda \neq 1$ will be in the range of Q. For eigenvectors corresponding to $\lambda = 1$ that are not enumerated in Z (if they exist), an orthogonality and counting argument must be employed to see that they must be in the range of Q. We coarsen the graph with a full representation of these vectors by forming a group for each set of leaves belonging to a single parent and letting $L_c = Q^t L Q$.

2.3. Further examples. There are many types of substructures that have locally supported eigenvectors. Here we describe three different types that are commonly observed in real-world graphs. However, this short list is not exhaustive. See Appendix A for algorithms that detect the following types of substructures and for a framework that could be used to detect more general ones. The first example is a generalization of the shared leaves.

EXAMPLE 2.13 (Hanging duplicate structures). Let $\mathcal{S}$ comprise of q identical subgraphs, each having c vertices with each of the subgraphs connected by a single edge to a common vertex $j \notin \mathcal{S}$ (which, in turn, has connection(s) into the rest of the graph, $\mathcal{V} \setminus \{j \cup \mathcal{S}\}$).

ETNA
Kent State University
http://etna.math.kent.edu



$$Q = \left[\begin{array}{c|c} \begin{matrix} \vdots & & \\ & 1/\sqrt{q_r} & \\ & \vdots & \\ & 1/\sqrt{q_r} & \\ & & \vdots \end{matrix} & \\ \hline & \begin{matrix} \ddots & & \\ & 1 & \\ & & \ddots \end{matrix} \end{array}\right]$$

FIG. 2.3. *Example of aggregating leaves that share a parent and all non-leaves in singlets.*

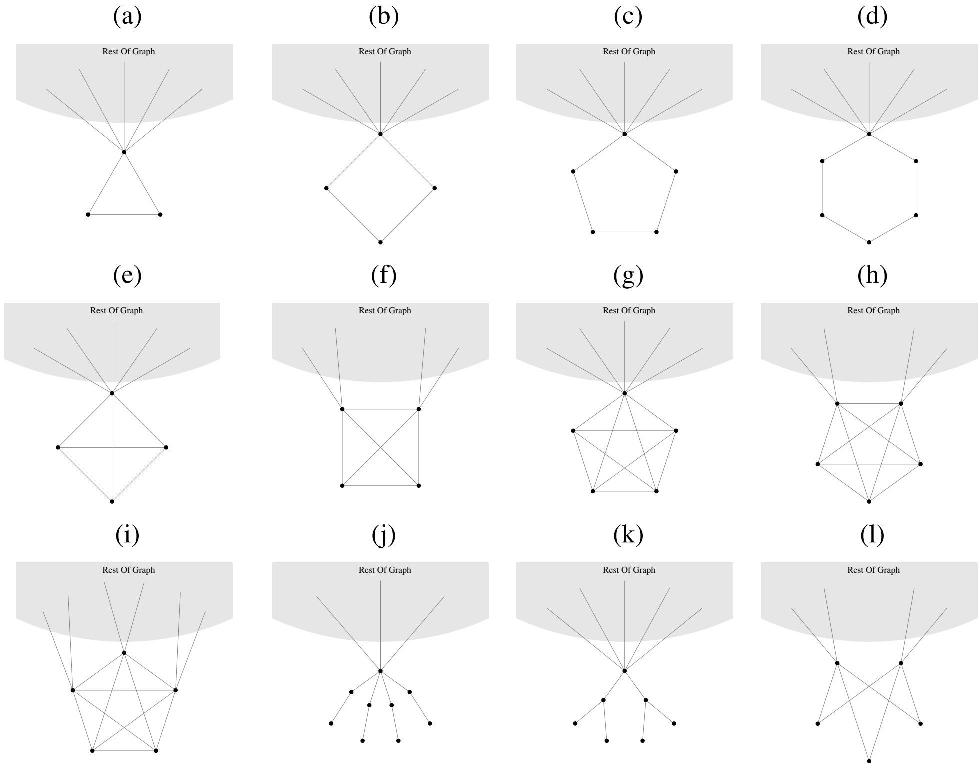


FIG. 2.4. *Further examples of common substructures associated with LSEVs: (a–d) hanging cycles (see Example 2.16), (a), (e–i) hanging cliques (see Example 2.15), (j–k) hanging duplicate structures (see Example 2.10 and Remark 2.12), and (l) vertices of degree 2 that share the same neighborhood.*

Then L_{11} is block-diagonal

$$L_{11} = \begin{bmatrix} B_c & & \\ & \ddots & \\ & & B_c \end{bmatrix} \qquad \text{and} \qquad L_{21} = \begin{bmatrix} \mathbf{c}_c^t & \cdots & \mathbf{c}_c^t \\ O & O & O \end{bmatrix},$$

ETNA
Kent State University
http://etna.math.kent.edu

with $\mathbf{c}_c = [-1, 0, \cdots, 0]^t$. Let $\{(\mathbf{w}_r, \mu_r)\}_{r=1}^c$ be the c eigenpairs of B_c and define $\mathbf{u}_p$ as in (2.7) for $p = 1, \ldots, q$. Also, let $P_\mathcal{S}$ be the appropriate injection from $\mathbb{R}^{|\mathcal{S}|}$ into $\mathbb{R}^n$. There are $(q-1)c$ locally supported eigenpairs of L on $\mathcal{S}$,

$$(P_\mathcal{S}(\mathbf{u}_p \otimes \mathbf{w}_r), \mu_r), \qquad \text{for} \qquad r = 1, \ldots, c \qquad \text{and} \qquad p = 1, \ldots, (q-1),$$

where $\otimes$ denotes the Kronecker tensor product. The eigenvectors of L that are not enumerated here are in the span of

$$\left\{P_\mathcal{S}(\mathbf{1}^{(q)} \otimes \mathbf{w}_r), r = 1, \ldots, c\right\} \cup \{\mathbf{e}_j \ :, \forall j \in \mathcal{V} \setminus \mathcal{S}\},$$

where $\mathbf{1}^{(q)}$ is the vector of all ones and length q and $\mathbf{e}_j$ denotes the i-th column of I_n.

EXAMPLE 2.14 (Duplicate chains). A version of Example 2.13 that commonly occurs in real-world graphs having tree-like structure is given by a set of chains of length c all connected to a common vertex. The case $c = 1$ corresponds to the shared leaves from Example 2.10. We have

$$B_1 = [1], \qquad B_2 = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \qquad \text{and} \qquad B_c = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}.$$

We make the following amusing observation: several chains of length 2 hanging from the same vertex yield eigenspaces of large multiplicities associated with the eigenvalues $\{\frac{3\pm\sqrt{5}}{2}\} =: \{1 + \phi^{\pm}\}$, where $\phi^{\pm} = \frac{1\pm\sqrt{5}}{2}$ and ϕ^+ is the famous *golden ratio*.

EXAMPLE 2.15 (Hanging cliques). Let c and l be integers with $c \geq 3$ and $1 \leq l \leq c-2$. A clique of size c that has exactly l vertices with edges connecting to the rest of the graph has $(c-l)-1$ locally supported eigenvectors of L associated with the eigenvalue $\lambda = c$. Their support $\mathcal{S}$ is the set of the $(c-l)$ vertices without external edges.

In the context of Theorem 2.4, we have a $(c-l) \times (c-l)$ matrix $L_{11} = cI - \mathbf{1}\mathbf{1}^t$ and an $l \times (c-l)$ matrix of all negative ones, which makes up the nonzero rows in L_{21}. Letting $q = (c-l)$, it is straightforward to verify that

$$L_{11}\mathbf{u}_p = c\mathbf{u}_p \qquad \text{and} \qquad L_{21}\mathbf{u}_p = \mathbf{0} \qquad \text{for} \qquad p = 1, \ldots, (q-1),$$

where $\{\mathbf{u}_p\}_{p=1}^{(q-1)}$ is the basis given in (2.7).

EXAMPLE 2.16 (Hanging cycles). Let c be an integer such that $c \geq 3$. A cycle of size c that has exactly one vertex with edges connecting to the rest of the graph has $\lfloor\frac{c-1}{2}\rfloor$ locally supported eigenvectors of L whose support $\mathcal{S}$ comprises the $(c-1)$ vertices without external edges.

Let j be the one vertex in the cycle that is connected to the rest of the graph. The subset $\mathcal{S} \subset \mathcal{G}$ contains the $(c-1)$ vertices in the cycle excluding j. Define the basis $\{\mathbf{u}_p\}_{p=1}^{\lfloor\frac{c-1}{2}\rfloor}$ such that

$$(\mathbf{u}_p)_i = \sin\left(\tfrac{2\pi p}{c} i\right) \qquad \text{for} \qquad i = 1, \ldots, c-1.$$

Again, decompose L as in (2.1). Verify that $L_{11}\mathbf{u}_p = \left[2 - 2\cos\left(\frac{2\pi p}{c}\right)\right]\mathbf{u}_p$, where

$$L_{11} = \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 2 \end{bmatrix},$$

ETNA
Kent State University
http://etna.math.kent.edu

which is the portion of L corresponding to the hanging cycle without vertex j. Also, verify that $L_{21}\mathbf{u}_p = \mathbf{0}$ for $p = 1, \ldots, \lfloor \frac{c-1}{2} \rfloor$ because the nonzero row of L_{21} is $[-1, 0, \cdots, 0, -1]$ and $\sin(\frac{2\pi p}{c}) = -\sin(\frac{2\pi p(c-1)}{c})$ for these values of p.

We conclude this subsection by providing Table 2.1, which contains various types of substructures common to real-world graphs and their spectral contributions for four of the graph matrices listed in Definition 1.1.

TABLE 2.1
The eigenvalues corresponding to common LSEVs for various graph-associated matrices, where $\xi_{p,c} := \cos\left(\frac{2\pi p}{c}\right)$. *(Note: when mentioning c-cycles, we have* $p = 1, \ldots, \lfloor \frac{c-1}{2} \rfloor$.*)*

	A	L	$\hat{L}$	$\lvert L\rvert$
Leaves	$\{0\}$	$\{1\}$	$\{1\}$	$\{1\}$
2-Chains	$\{\pm 1\}$	$\left\{\frac{3}{2} \pm \frac{\sqrt{5}}{2}\right\}$	$\left\{1 \pm \frac{1}{\sqrt{2}}\right\}$	$\left\{\frac{3}{2} \pm \frac{\sqrt{5}}{2}\right\}$
c-Cliques	$\{-1\}$	$\{c\}$	$\left\{1 + \frac{1}{c-1}\right\}$	$\{c-2\}$
c-Cycles	$\{2\xi_{p,c}\}_p$	$\{2 - 2\xi_{p,c}\}_p$	$\{1 - \xi_{p,c}\}_p$	$\{2 + 2\xi_{p,c}\}_p$

2.4. The slightly weighted case. Consider a graph with weights that differ slightly from one. Use the weight $1 + \epsilon_{ij}$ for each edge $(i, j) \in \mathcal{E}$, where $|\epsilon_{ij}| < \epsilon/(2\max\{d_i, d_j\})$ with a small positive constant ϵ. Let the matrix L be the graph Laplacian for the unweighted case, and introduce $E \in \mathbb{R}^{n\times n}$ to represent the perturbations from one in the edge weights,

$$E_{ij} = \begin{cases} \sum_{j\in\mathcal{N}_i} \epsilon_{ij} & \text{if } i = j \\ -\epsilon_{ij} & \text{if } i \neq j \end{cases},$$

so that $(L + E)$ is the graph Laplacian of the weighted graph. Using Gerschgorin's theorem, we have

$$\|E\| \leq \epsilon.$$

Consider seeking the eigenpairs of $(L + E)$ associated with the $(K - 1)$ smallest nonzero eigenvalues.

$$\begin{array}{rcl} (L+E)\mathbf{v}_k &=& \lambda_k \mathbf{v}_k \\ \mathbf{v}_k^t \mathbf{v}_l &=& \delta_{kl} \end{array} \quad \text{for} \quad k, l = 2, 3, \ldots, K. \tag{2.10}$$

Let Z be a collection of LSEVs of L that have been identified and let Q be a sparse matrix that spans the orthogonal complement of Z. The following theorem and corollary show that Q can be used to approximate the eigenpairs we seek in (2.10) within an eigenresidual tolerance of ϵ.

THEOREM 2.17. *Let* $(\mathbf{w}_c, \mu_c)$ *be an eigenpair for* $Q^t(L + E)Q$. *Then*

$$\frac{\|(L + E - \mu_c I)Q\mathbf{w}_c\|}{\|Q\mathbf{w}_c\|} \leq 2\epsilon.$$

Proof. We have $\|Q\mathbf{w}_c\| = \|\mathbf{w}_c\|$ and $\|Q^tEQ\| \leq \|E\|$ due to $Q^tQ = I_{n_c}$. Rearranging the equation $Q^t(L+E)Q\mathbf{w}_c = \mu_c\mathbf{w}_c$ and taking norms, we see the quality of $(\mathbf{w}_c, \mu_c)$ as an eigenpair for Q^tLQ,

$$\begin{aligned} \|(Q^tLQ - \mu_c I_{n_c})\mathbf{w}_c\| &= \| - Q^tEQ\mathbf{w}_c\| \\ &\leq \|Q^tEQ\|\|\mathbf{w}_c\| = \|Q^tEQ\|\|Q\mathbf{w}_c\| \\ &\leq \|E\|\|Q\mathbf{w}_c\| = \epsilon\|Q\mathbf{w}_c\|. \end{aligned}$$

Using the triangle inequality, the result in Theorem 2.8, and the previous estimate, we prove the inequality,

$$\begin{aligned}\|((L+E)-\mu_c I)Q\mathbf{w}_c\| &= \|(L-\mu_c I)Q\mathbf{w}_c + EQ\mathbf{w}_c\| \\ &\le \|(L-\mu_c I)Q\mathbf{w}_c\| + \|EQ\mathbf{w}_c\| \\ &= \|(L_c-\mu_c I_c)\mathbf{w}_c\| + \|EQ\mathbf{w}_c\| \le 2\epsilon\|Q\mathbf{w}_c\|. \qquad \square\end{aligned}$$

COROLLARY 2.18. *Let $(\mathbf{w}_c, \mu_c)$ be an eigenpair for L_c. Then*

$$\frac{\|((L+E)-\mu_c I)Q\mathbf{w}_c\|}{\|Q\mathbf{w}_c\|} \le \epsilon.$$

The implications of these theorems are: (i) for a slightly-weighted graph $\mathcal{G}$, the knowledge of LSEVs of the unweighted version of $\mathcal{G}$ is useful for obtaining accurate initial approximations to the eigenvectors of the graph Laplacian and (ii) for a graph $\mathcal{G}$ with time-dependent edge weights that vary slightly, $w_{ij}(t) = 1 + \epsilon_{ij}(t)$, the same eigenvectors serve as accurate initial approximations independent of t. These results apply only to applications where the topology of a network remains fixed, but the weights of their edges fluctuate slightly over time.

2.5. The edge principle for L. We conclude this section with an interesting property of the LSEVs associated with the combinatorial graph Laplacian that is not shared by the other common graph-associated matrices. Certain types of LSEVs (e.g., those associated with shared leaves or hanging cliques) span the local orthogonal complement of the constant vector restricted to the local subset $\mathcal{S}$. Therefore, all other eigenvectors are constant across $\mathcal{S}$. We can use the following result to demonstrate that we can add, delete, or reweight edges within $\mathcal{S}$ without changing the global eigenvectors L.

THEOREM 2.19 (Edge principle, [21]). *Let L be the combinatorial graph Laplacian of a given graph $\mathcal{G}$. For any eigenpair $(\lambda, \mathbf{v})$ of L, consider a vertex pair (s,t) for which $v_s = v_t$. Let L' be the combinatorial Laplacian associated with the graph $\mathcal{G}'$ which is obtained by adding (or deleting) the edge between the two vertices (s,t). Then, $(\lambda, \mathbf{v})$ is an eigenpair of L' as well.*

Proof. The tuple $(\lambda, \mathbf{v})$ is an eigenpair, so all n equations $\lambda\mathbf{v} = L\mathbf{v}$ hold. The s-th row of this system of equations is

$$\lambda v_s = \sum_{j\in\mathcal{N}_s} (v_s - v_j).$$

We rewrite this equation as

$$\lambda v_s = w_{st}(v_s - v_t) + \sum_{j\in\mathcal{N}_s\setminus\{t\}} (v_s - v_j),$$

which demonstrates that if $v_s = v_t$, then this equation holds independent of the value of w_{st} (and the absence or presence of the edge (s,t)). Similarly, the validity of the t-th equation is unaffected by changes regarding the edge (s,t). All the other $(n-2)$ equations are trivially unaffected so that $L'\mathbf{v} = \lambda\mathbf{v}$ holds as well. $\square$

COROLLARY 2.20. *Let Z be the matrix (2.3) containing the LSEVs of L corresponding to shared leaves or hanging cliques over a collection of local subsets, $\{\mathcal{S}_r\}_{r=1}^R$. Let Q be the orthogonal complement to Z. Let L' be any combinatorial Laplacian obtained by adding, deleting, or reweighting edges (s,t) in $\mathcal{G}$ such that $s,t \in \mathcal{S}_r$ for some r. Then any eigenpair $(\lambda, \mathbf{v})$ of L such that $\mathbf{v} \in span(Q)$ is also an eigenpair of L'.*

Proof. Examples 2.10 and 2.15 demonstrate that every vector in $\text{span}(Q)$ is constant over each $\mathcal{S}_r$. Theorem 2.19 demonstrates that we can change edges (s,t) such that $s,t \in \mathcal{S}_r$ without changing the eigenvectors $\mathbf{v} \in \text{span}(Q)$ or the associated λ. ☐

EXAMPLE 2.21. Consider three shared leaves in an unweighted graph. Add an edge between leaf one and leaf two. The LSEVs of L have changed, yet the global eigenvectors have not. Note that there are two LSEVs on this new structure for L', yet there is only one for the respective adjacency matrix (associated with the hanging triangle).

The property given in Corollary 2.20 may allow for more aggressive coarsening in many real-world graphs than the techniques used in Section 4. However, the success of such an approach requires a clever method for efficiently detecting this wide class of graph substructures.

3. Application of LSEVs to commute time. Any network science computation that has connection with the eigenpairs of a graph-associated matrix may potentially benefit from detecting the existence of LSEVs. There are quite a few common data mining related computations that have spectral formulas: query rankings can be inferred from eigenpairs [6], partitioning and clustering can be performed using eigenpairs [27], triangles can be counted with eigenvalues [25], etc.

We focus on employing LSEVs to aid in the calculation of *commute time*, a distance measure for pairs of vertices, due to recent interest in a wide range of application areas. As a distance measure, commute time can be used to perform several data mining related tasks, such as query ranking and clustering.

DEFINITION 3.1. *The* commute time *between vertices i and j, denoted $C(i,j)$, is defined to be the expected length of random walks that start from vertex i, visit vertex j, and return to vertex i.*

Recall that the full eigendecomposition of a graph Laplacian is $L = V\Lambda V^t$, where V is an orthogonal matrix with the eigenvectors of L in its columns, $V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$, Λ is a non-negative diagonal matrix with $\Lambda = \text{diag}[\lambda_1 = 0, \lambda_2, \ldots, \lambda_n]$, and the eigenvalue ordering given in (1.2). We introduce the notation $v_i^{(k)} = V_{ik}$, i.e., $v_i^{(k)}$ is the i-th entry of the k-th eigenvector of L. It is well-known that the commute time is given by the spectral formula [10]

$$C(i,j) = \text{vol}(\mathcal{G}) \sum_{k=2}^{n} \frac{1}{\lambda_k} \left(v_i^{(k)} - v_j^{(k)} \right)^2, \tag{3.1}$$

where $\text{vol}(\mathcal{G}) = \sum_{k=1}^{n} d_k$ is the *graph volume*. This can also be written as the quadratic form,

$$C(i,j) = \text{vol}(\mathcal{G})(\mathbf{e}_i - \mathbf{e}_j)^t L^+ (\mathbf{e}_i - \mathbf{e}_j),$$

where $\mathbf{e}_i$ is the i-th column of the identity and L^+ is the Moore-Penrose pseudo-inverse of the combinatorial graph Laplacian $L^+ = V\Lambda^+ V^t$. There is a similar formula based on the pseudo-inverse of the normalized graph Laplacian [10].

An interesting property of the spectral commute time formula is that it is a sum of positive terms, meaning that any partial sum gives a lower bound on the actual value. We consider approximating the commute time by truncating the sum in (3.1), which would amount to calculating the $K-1$ eigenpairs corresponding to the lowest eigenvalues excluding $\lambda_1 = 0$. For the arguments presented here, assume that we calculate these eigenpairs exactly (including

ETNA
Kent State University
http://etna.math.kent.edu

numerical errors is beyond the scope of this work)

$$C(i,j) \approx C_K(i,j) = \mathrm{vol}(\mathcal{G}) \sum_{k=2}^{K} \frac{1}{\lambda_k} \left(v_i^{(k)} - v_j^{(k)} \right)^2 .$$

The *truncation error*, $\tau_K(i,j) := C(i,j) - C_K(i,j)$, is bounded in a simple way by noting that $\|\mathbf{v}_k\| = 1$ implies $\left(v_i^{(k)} - v_j^{(k)} \right)^2 \leq 2$ for any i and j

$$\tau_K(i,j) \leq \mathrm{vol}(\mathcal{G}) \frac{2(n-K)}{\lambda_{K+1}} \quad .$$

This bound is not sharp: there is no $K \ll n$ with vertices i and j such that equality is reached. In fact, the use of Hölder's inequality gives us another $\mathcal{O}(1/\lambda_{K+1})$-bound where the constant is quite a bit better.

THEOREM 3.2 (Uniform truncation error for commute time). *For all pairs of vertices i and j, we have*

$$\tau_K(i,j) \leq \frac{2\mathrm{vol}(\mathcal{G})}{\lambda_{K+1}}.$$

Proof. Using Hölder's inequality, $\|\mathbf{fg}\|_1 \leq \|\mathbf{f}\|_\infty \|\mathbf{g}\|_1$, the fact that $V^t \mathbf{e}_i$ (the rows of V) are orthonormal ($VV^t = I$), and the ordering of the eigenvalues, we have

$$\begin{aligned} C(i,j) - C_K(i,j) &= \mathrm{vol}(\mathcal{G}) \sum_{k=K+1}^{n} \frac{1}{\lambda_k} \left(v_i^{(k)} - v_j^{(k)} \right)^2 \\ &\leq \mathrm{vol}(\mathcal{G}) \left[\max_{k=K+1,\ldots n} \frac{1}{\lambda_k} \right] \left[\sum_{k=K+1}^{n} \left(v_i^{(k)} - v_j^{(k)} \right)^2 \right] \\ &\leq \frac{\mathrm{vol}(\mathcal{G})}{\lambda_{K+1}} \left[\sum_{k=1}^{n} \left(v_i^{(k)} - v_j^{(k)} \right)^2 \right] \\ &= \frac{\mathrm{vol}(\mathcal{G})}{\lambda_{K+1}} \|V^t \mathbf{e}_i - V^t \mathbf{e}_j\|^2 \\ &= \frac{2\mathrm{vol}(\mathcal{G})}{\lambda_{K+1}} \quad . \qquad \square \end{aligned}$$

If the decay of $1/\lambda_k$ is fast enough as k increases, then these types of bounds are immediately useful. However, if a graph has a large number of LSEVs, then the decay of $1/\lambda_k$ may be quite slow and this bound is not useful unless the LSEVs are detected and all known eigenvectors are used to improve the truncation error bound. We give the following corollary to the previous theorem.

COROLLARY 3.3 (Pair-wise truncation error for commute time). *Let $\mathcal{K}$ be the set of indices of all known eigenpairs (the LSEVs detected and eigenpairs that have been computed). Let $C_{\mathcal{K}}(i,j)$ be the estimation of commute time using all the known eigenpairs. Then, for any pair of vertices i and j, we have*

$$\tau_{\mathcal{K}}(i,j) := C(i,j) - C_{\mathcal{K}}(i,j) \leq \frac{\mathrm{vol}(\mathcal{G})}{\lambda_{K+1}} \left[2 - \sum_{k \in \mathcal{K}} \left(v_i^{(k)} - v_j^{(k)} \right)^2 \right].$$

Proof. Using $\|V^t\mathbf{e}_i - V^t\mathbf{e}_j\|^2 = 2$, we have

$$\sum_{k\in\mathcal{K}}\left(v_i^{(k)} - v_j^{(k)}\right)^2 + \sum_{k\notin\mathcal{K}}\left(v_i^{(k)} - v_j^{(k)}\right)^2 = 2.$$

Inserting this equation into the proof of Theorem 3.2, we obtain the result. $\square$

The assumption that there is no numerical error associated with the eigenpairs is realistic for a wide class of LSEVs. For the other eigenpairs, this assumption is not met in practice, and the estimates involved should depend on the residuals of the eigenpair approximations. Below we give an example of a common practical situation where the truncation error is known to be zero for certain pairs of vertices by using only LSEVs as known eigenvectors.

Additionally, bounds on $C(i,j)$ are easily obtained by the detection of LSEVs. We use the shared-leaf example to illustrate this.

THEOREM 3.4. *For a graph with shared leaves, we have the following results.*

(i) If neither vertex i or j are shared leaves, then $C(i,j)$ can be obtained from L_c.

(ii) If the vertex i is a shared leaf of a parent with $q > 1$ shared leaves, then

$$\mathrm{vol}(\mathcal{G})\frac{q-1}{q} \le C(i,j) \le \mathrm{vol}(\mathcal{G})\left(\frac{q-1}{q} + 1 + \frac{1}{\sqrt{q}}\right).$$

(iii) If the vertices i and j are both shared leaves of different parents, p_i and p_j, each with $q_i, q_j > 1$ shared leaves, then

$$\mathrm{vol}(\mathcal{G})\left(\frac{q_i-1}{q_i} + \frac{q_j-1}{q_j}\right) \le C(i,j) \le \mathrm{vol}(\mathcal{G})\left(\frac{q_i-1}{q_i} + \frac{q_j-1}{q_j} + \frac{1}{\sqrt{q_i}} + \frac{1}{\sqrt{q_j}}\right).$$

(iv) If the vertices i and j are both shared leaves of the same parent with $q > 1$ shared leaves, then

$$C(i,j) = 2\mathrm{vol}(\mathcal{G}).$$

Proof. For (i), note that all the LSEVs in Z are zero-valued at i and j. The only nonzero terms in (3.1) are associated with the eigenvectors in the orthogonal complement Q. These eigenvectors are perfectly represented by eigenvectors of L_c. Using the basis in (2.7), we prove the lower bounds in (ii) and (iii). Because the entire eigenspace associated with Z has eigenvalue $\lambda = 1$, we have $C(i,j) \ge \mathrm{vol}(\mathcal{G})\|Z^t(\mathbf{e}_i - \mathbf{e}_j)\|^2$. If i is in a shared leaf but j is not, then $Z^t\mathbf{e}_j = \mathbf{0}$ and

$$\|Z^t(\mathbf{e}_i - \mathbf{e}_j)\|^2 = \|Z^t\mathbf{e}_i\|^2 = \sum_{p=1}^{(q-1)}[(\mathbf{u}_p)_i]^2 = \frac{q-1}{q},$$

from which the lower bound in (ii) follows. For part (iii), assume i and j are shared leaves from different parents. Let $Z^{(1)}$ be the columns of Z associated with the LSEVs that are nonzero over the leaves of p_i. Define $Z^{(2)}$ similarly with respect to p_j. Then $(Z^{(1)})^t\mathbf{e}_j = \mathbf{0}$ and $(Z^{(2)})^t\mathbf{e}_i = \mathbf{0}$, implying

$$\|Z^t(\mathbf{e}_i - \mathbf{e}_j)\|^2 \ge \|(Z^{(1)})^t\mathbf{e}_i\|^2 + \|(Z^{(2)})^t\mathbf{e}_j\|^2 = \frac{q_i-1}{q_i} + \frac{q_j-1}{q_j}\ .$$

The upper bounds for (ii) and (iii) are realized by applying Corollary 3.3. We prove (iv) by noting that $(\mathbf{e}_i - \mathbf{e}_j)$ is orthogonal to the constant vector and therefore it is in the range of Z. This implies $\|Z^t(\mathbf{e}_i - \mathbf{e}_j)\|^2 = \langle(QQ^t + ZZ^t)(\mathbf{e}_i - \mathbf{e}_j), \mathbf{e}_i - \mathbf{e}_j\rangle = \|\mathbf{e}_i - \mathbf{e}_j\|^2 = 2$. The other terms involved in $C(i,j)$ are all zero because the eigenvectors in the range of Q satisfy $v_i^{(k)} = v_j^{(k)}$. $\square$

ETNA
Kent State University
http://etna.math.kent.edu

3.1. LSEVs as counterexamples to conjectures regarding scale-free graphs. In [28], it is stated that ”The raw commute distance is not a useful distance function on large graphs.” It is important to note that, out of context, the scope of this statement seems very wide. The authors do not consider all large graphs in their theory. Instead, they assume a class of graphs common to machine learning applications (so called *nearest neighbor graphs* and ϵ*-graphs*) with the following properties: (i) the minimal degree slowly increases with the number of vertices **and** (ii) random walks are quickly mixing. Given such graphs, the commute time between two vertices is well-approximated by a function of the degree density of both vertices,

$$C(i,j) \approx \mathrm{vol}(\mathcal{G}) \left(\frac{1}{d_i} + \frac{1}{d_j} \right), \tag{3.2}$$

and the quality of the approximation is better for larger graphs. We review the conjecture that some members in the research community seem to have made: the approximation (3.2) is highly accurate for **any** large scale-free graph.

Many scale-free graphs of interest do not have property (i), and the results in [28] do not apply for such graphs. For example, there are scale-free graphs with billions of vertices that have many vertices of degree one and two. If certain types of LSEVs are present in a graph, then this demonstrates that the error in the approximation (3.2) can be bounded from zero independent of the size of the graph. We give the simplest example.

EXAMPLE 3.5 (Hanging triangles). Consider the class of graphs with one or more triangles that have only a single vertex with any connection to vertices not in the triangle (Example 2.16 with $c = 3$ and $l = 1$). Let the vertices i, j, and k comprise a connected triangle that hangs off the rest of $\mathcal{G}$ ($d_i = d_j = 2$ and k contains at least one connection with vertices in $\mathcal{V} \setminus \{i, j, k\}$). Equation (3.2) suggests that if $|\mathcal{V}|$ is sufficiently large, the commute time $C(i, j)$ should be well-approximated by

$$\mathrm{vol}(\mathcal{G}) \left(\frac{1}{d_i} + \frac{1}{d_j} \right) = \mathrm{vol}(\mathcal{G}).$$

However, there is a LSEV supported on $\mathcal{S} = \{i, j\}$. Define the vector $\mathbf{v}$ with the coefficients being $v_i = 1/\sqrt{2}$, $v_j = -1/\sqrt{2}$, and zero-valued for all other vertices. The decomposition in (2.1) yields

$$L_{11} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, \qquad \text{and} \qquad L_{21} = \begin{bmatrix} -1 & -1 \\ 0 & 0 \\ \vdots & \vdots \end{bmatrix}.$$

It is easy to verify that $\mathbf{v}$ is a LSEV associated with the eigenvalue $\lambda = 3$ (see Example 2.16). Due to orthogonality, all other eigenvectors are equal at the vertices i and j. Therefore, the commute time between i and j only involves the LSEV, and

$$C(i,j) = \mathrm{vol}(\mathcal{G}) \left(\frac{1}{3} \right) \left(\frac{1}{\sqrt{2}} - \frac{-1}{\sqrt{2}} \right)^2 = \frac{2}{3} \mathrm{vol}(\mathcal{G}).$$

The error in the approximation offered by (3.2) is not arbitrarily small for large $|\mathcal{V}|$ within this class of graphs.

We remark that this example does not prove that commute time is a good distance measure for all scale-free graphs, it only serves to show that the approximation (3.2) is not accurate for all vertex pairs in this class of graphs. In [7], the authors demonstrate that the k

ETNA
Kent State University
http://etna.math.kent.edu

closest vertices to a source vertex, using commute time or (3.2), tend to be quite similar for a few prototypical scale-free graphs provided that k is a big enough number. The example rankings are often quite different for lists of length $k = 5$ but tend to be highly similar for larger k. A revised conjecture is that the quality of the approximation (3.2) is high for pairs of large-degree vertices and that this property can explain the correlation in the rankings as an exceptionally large number of paths involve the high degree vertices.

4. Numerical experiments. We present a series of experiments designed to demonstrate the use of LSEVs to facilitate the computation of eigenpairs of graph Laplacian matrices. These tests are intended to be illustrative rather than exhaustive. Principally, we show that LSEVs can be identified by detecting their associated graph substructure and that they can subsequently be used to generate coarse graphs, L_c, with significantly reduced complexity, which can be used to compute eigenpairs in the remaining portion (non-LSEVs) of the spectrum of the original graphs. We do this for a selection of graphs including both a synthetic graph generator and some real-world graphs to indicate robustness of the method. We also demonstrate that in addition to reducing the complexity of the coarse graph, the method in some cases also noticeably reduces the computational effort (number of iterations) and computational time required to compute the Laplacian spectrum.

4.1. Graphs. Our tests are conducted on graph Laplacian matrices formed for a class of synthetic graphs as well as two well-known real-world graphs. The graphs we employ are the following.

1. The *Preferential Attachment Model* employs a synthetic graph generated using a common random graph model, a version of the preferential attachment (PA) model proposed in [3]. Here, random graphs are generated by starting with a small core graph and successively adding new vertices, each with one or two new edges. These edges are randomly attached to old vertices with a probability that is proportional to the degrees of those existing vertices. This graph generation method is often described as *the rich getting richer.* It results in a graph with a power-law degree distribution but without well-developed internal communities. Our examples all have essentially the same number of edges as vertices. We employ three such graphs, with, respectively, 33,000, 66,000, and 131,000 vertices and edges.
2. The *Opte Internet Graph* (denoted Opte1), shown in Figure 1.1, is the result of scanning connections between class C networks on the internet. The graph and the visualization were downloaded from [19]. The Opte1 graph contains just under 36,000 vertices and 43,000 edges. We note that this graph has quite a bit of a tree-like structure in its periphery and there are many LSEVs associated with short, shared chains.
3. The *Enron Email Correspondence Graph*, downloaded from [18], was created using email traffic from employees of the Enron corporation. The data were originally released by the investigators of the Enron scandal that unfolded in 2001. Vertices in the graph are either Enron email accounts or non-Enron email accounts that sent (or received) one or more messages to (or from) an Enron account. An undirected edge (i, j) is assigned if there was any email communication between i and j during the span of time the data represents. This is an example of a dilation of an *induced subgraph* of a larger graph, namely the graph of all email accounts and the presence of communication between two email accounts. (The subgraph induced by the set of all Enron email accounts would give all Enron accounts and presence of communication between Enron-only accounts.) Graphs of this type are prone to a highly simplistic periphery in cases where many of the vertices outside the inducing set are only connected to few vertices within. The Enron graph has 34,000 vertices and 181,000 edges.

4.2. LSEVs for graph coarsening. For the graphs described in Section 4.1, we demonstrate the reduction in graph complexity offered by the detection of LSEVs. The results of detecting LSEVs and forming coarsened matrices L_c are presented in Table 4.1. For all of the graphs, we see that a large portion ($23\% - 46\%$) of the Laplacian spectrum is made up of the detected LSEVs. Moreover, with the exception of the Enron graph, the complexity of L_c (i.e., the number of edges in the coarse matrix) can be considerably reduced as well ($18\% - 46\%$). While in the Enron graph we do see a significant portion of the spectrum made up of LSEVs, the complexity reduction is only $4\% - 5\%$.

The graph substructures that induce LSEVs are detected using the algorithms described in Appendix A. Specifically, shared leaves (LF) and shared chains of length 2 (2C) are detected using Algorithm 2. Hanging triangles (T) are detected using Algorithm 3.

It is natural to ask for the computational cost of these detection algorithms. Combinatorial mathematicians are well aware that rigorous detection algorithms can be extremely difficult and expensive. We note, however, that we are never doing an expensive combinatorial search. For example, consider the case where we seek LSEVs supported on hanging cliques of size c. We are not looking for *all* cliques of size c in the graph, which would amount to a $\mathcal{O}(n^c)$ cost without employing heuristics (or additional graph properties). Instead, we are looking for cliques of size c that contain at least two vertices of degree equal to $(c-1)$. This constraint allows us to narrow the search greatly, and, for small c, this search is fairly inexpensive.

We give an example that demonstrates some potential computational savings of detecting LSEVs and using them to coarsen the graph before applying an eigensolver. First, we apply several instances of MATLAB's iterative eigensolver, `eigs()`, to the original graph Laplacian L (associated with the Opte1 graph) and monitor the number of iterations and wall-clock time. We ask for several different numbers of smallest eigenvalues, `nev`, and associated eigenvectors for a few different error tolerances, `tol`. We seed `eigs()` with the same random non-negative initial vector each time and allow the algorithm to determine how many storage vectors `nsv` are appropriate. Secondly, we coarsen the graph by detecting shared-leaves, which takes about 0.46 seconds, and we apply the same eigensolver to L_c using similar parameters as we use for the corresponding eigensolver involving L, again monitoring the number of iterations and wall-clock time. Table 4.2 displays the performance for each of these solves, where detecting the LSEVs amounts to significant savings in time and storage and additionally computes a large number of interior eigenvalues and associated eigenpairs (over $0.22n$+`nev` eigenpairs are calculated using the LSEV approach).

From the vertex and edge reductions of L_c displayed in Table 4.1, we know that the storage cost involved in retaining `nsv` storage vectors is reduced by over 22% and the computational cost of applying a matvec is reduced by around 18%. Additionally, the number of iterations that `eigs()` uses is also greatly reduced for L_c, which can be attributed to removing a very large number of eigenvectors associated with $\lambda = 1$, allowing the Krylov process to select polynomials that concentrate on the low eigenvalues.

Note that we use the `'SA'` (smallest algebraic) option in `eigs()`, which does not employ a preconditioner, whereas the `'SM'` (smallest magnitude) would use a Cholesky preconditioner. Our reason for not using the preconditioner is two-fold: (i) for a large enough real-world graph this type of preconditioner runs out of memory, and (ii) we aim to demonstrate the potential computational savings in graph coarsening, which are magnified by a less efficient method.

5. Conclusion and further work. Our primary contributions are to characterize a class of graph substructures that admits locally supported eigenvectors, to demonstrate how to detect such structures, and to calculate the associated eigenpairs. We develop a fairly extensive

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 4.1
Five different examples of using detected LSEVs to reduce the complexity of graphs. The original graphs are from a preferential attachment model (PA), internet router connections (Opte1), and electronic communications (Enron). The original number of vertices and edges are denoted $|\mathcal{V}|$ and $|\mathcal{E}|$, respectively. The types of LSEVs detected are shared leaves (LF), shared chains of length 2 (C2), and hanging triangles (T). The numbers of edges and vertices in the coarsened graph, $|\mathcal{V}_c|$ and $|\mathcal{E}_c|$, are reported as well as the percentages of eigenpairs identified and edges reduced.

Graph	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	Detection	$\|\mathcal{V}_c\|$	$\|\mathcal{E}_c\|$	Epairs Identified	Edge Red.
PA	32,768	32,767	LF	20,515	20,514	37.4%	37.4%
			LF, 2C	17,651	17,650	46.1%	46.1%
PA	65,536	65,536	LF	41,190	41,189	37.2%	37.2%
			LF, 2C	35,476	35,475	45.9%	45.9%
PA	131,072	131,071	LF	82,346	82,345	37.2%	37.2%
			LF, 2C	70,656	70,655	46.1%	46.1%
Opte1	35,635	42,822	LF	27,548	34,735	22.7%	18.9%
			LF, 2C	25,686	32,873	27.9%	23.2%
Enron	33,696	180,811	LF	24,981	172,096	25.9%	4.8%
			LF, T	24,564	171,262	27.1%	5.3%

TABLE 4.2
Iteration counts and timings (in parentheses) of MATLAB's eigensolver `eigs()` *applied to L and L_c (coarsened using shared leaves) for the Opte1 graph for several different numbers of lowest eigenvalues,* `nev`*, and eigenresidual tolerance levels,* `tol`*. The value* `nsv` *is the number of vectors the algorithm chooses to store by default, with the exception for* `nev` $= 10$*, where* `nsv` $= 30$ *was chosen. DNC means the method did not converge in 25,000 iterations.*

		`tol` = 1e-3		`tol` = 1e-5		`tol` = 1e-8	
`nev`	`nsv`	L	L_c	L	L_c	L	L_c
1	20	336 (20.2s)	132 (5.5s)	1130 (66.0s)	354 (14.3s)	3262 (189.2s)	826 (33.0s)
5	20	5297 (388.7s)	1258 (67.3s)	8459 (618.9s)	2135 (110.1s)	17257 (1246.0s)	3006 (156.7s)
10	30	2180 (262.9s)	1010 (81.3s)	4159 (522.1s)	1046 (92.3s)	5229 (670.9s)	1601 (134.4s)
25	50	14366 (2512.1s)	2653 (334.6s)	15768 (2672.3s)	3767 (474.3s)	15514 (2704.8s)	4813 (642.85s)
50	100	DNC (14300.6s)	840 (381.4s)	DNC (15150.6s)	1206 (517.1s)	DNC (14968.2s)	926 (437.2s)

theory of these structures showing the spectral partitioning they give rise to, and we demonstrate how the sparsity of the local eigenvectors may be exploited to reduce the complexity in calculations for the other (non-LSEV) eigenpairs. We elucidate the theory governing the relationship between the original and coarsened matrices. We give an example where the knowledge of locally supported eigenvectors helps to predict the accuracy of spectral calculations. We also present numerical experiments illustrating the potential efficacy of employing LSEVs in practice, demonstrating both the efficient computation of sizeable fractions of the Laplacian spectrum and the significant reduction in size and complexity of the coarsened graph used to find the remainder of the spectrum. These numerical results quantify the reductions of graph size and complexity for small graphs from generators and a few real-world networks and demonstrate the computational savings involved in coarsening a graph before employing an eigensolver.

Our future work will be geared towards the use of LSEVs in the context of many diverse types of spectral calculations for undirected, scale-free graphs and exploiting the knowledge of LSEVs to improve bounds on the numerical error incurred by such computations. In addition, we are working on generalizing this theory to eigenvectors of *essentially local support*, that is, to eigenvectors that are not strictly local in their support but whose nonzero entries decay rapidly away from a local subset of the vertices. Our examinations of more general graph spectra have shown that such eigenvectors likely exist, but a theory of the graph properties that give rise to them and methods for detecting them have yet to be discovered. Developing a better understanding of the graph and matrix characteristics that admit essentially locally supported eigenvectors will allow us to further study graph coarsening and quantify the spectral error associated with coarsening processes.

Acknowledgments. We thank Panayot Vassilevski for many useful discussions on this subject and for his invaluable help editing this manuscript.

Appendix A. Algorithms to find LSEVs. We propose a general algorithm, Algorithm 1, that could be used to find out if individual sets within a family support LSEVs. For each

---

Algorithm 1: Framework for finding locally supported eigenvectors.

---

input : a connected, unweighted, and undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$
output : locally supported eigenvectors Z and orthogonal complement Q
Identify a family of small subsets $\{\mathcal{S}_r\}_{r=1}^R$ to check.
$Z \leftarrow [\,]$
$Q \leftarrow [\,]$
Initialized unvisited vertices, $\mathcal{U} \leftarrow \mathcal{V}$.
for $r = 1, \ldots, R$ **do**
  $\hat{Q} \leftarrow [\,]$
  $M \leftarrow 0$
  Use $\mathcal{S}_r$ to define L_{11} and L_{21} as in (2.1).
  Let Π and P be defined as in (2.3).
  Solve $L_{11}\mathbf{u} = \lambda\mathbf{u}$.
  for *each eigenspace* U *of* L_{11} **do**
    Solve $U^t L_{21}^t L_{21} U \mathbf{y} = \mu \mathbf{y}$.
    for *each* $\mathbf{y}_j$ *with* $\mu_j = 0$ **do**
      $M \leftarrow M + 1$
      $Z \leftarrow [Z, \Pi P U \mathbf{y}_j]$
    end
    for *each* $\mathbf{y}_j$ *with* $\mu_j \neq 0$ **do**
      $\hat{Q} \leftarrow [\hat{Q}, \Pi P U \mathbf{y}_j]$
    end
  end
  if $M \geq 1$ **then**
    $Q \leftarrow [Q, \hat{Q}]$
    $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{S}_r$
  end
end
for $i \in \mathcal{U}$ **do**
  $Q \leftarrow [Q, \mathbf{e}_i]$
end

Algorithm 2: Detecting shared chains of identical length.

input : a connected, unweighted, and undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, max chain length c_{max}
output : collection of subsets $\{\mathcal{S}_r\}_{r=1}^{R}$ containing locally supported eigenvectors.
Set $\mathcal{L}_j = \{i \in \mathcal{V} \,:\, d_i = j\}$ for $j = 1$ and 2.
Initialize vector $\mathbf{g} \in \mathbb{R}^n$ to $g_i = 0$ for $i \in \mathcal{L}_1$ and -1 otherwise.
Initialized unvisited vertices, $\mathcal{U} \leftarrow \mathcal{V} \setminus \mathcal{L}_1$.
Set $\mathcal{T} = \mathcal{L}_1$.
r = 0.
for $c = 1, \ldots, c_{max}$ **do**
  $\mathcal{T}_{new} = \emptyset$.
  for $i \in \mathcal{T}$ **do**
    $\mathcal{T} \leftarrow \mathcal{T} \setminus \{i\}$
    Set j to the unique element in $\mathcal{N}_i \cap \mathcal{U}$.
    if $d_j = 2$ **then**
      $g_j = g_i + 1$.
      $\mathcal{T}_{new} \leftarrow \mathcal{T}_{new} \cup \{j\}$.
      $\mathcal{U} \leftarrow \mathcal{U} \setminus \{j\}$
    end
    else
      Let $\mathcal{C} = \{p \in \mathcal{N}_j \,:\, g_p = g_i\}$
      if *card*$(\mathcal{L}_{d_i} \cup \mathcal{C}) > 1$ **then**
        $r \leftarrow r + 1$.
        Set $\mathcal{S}_r$ to include all of each chain.
      end
    end
  end
  $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{T}_{new}$
end
$R = r$.

set $\mathcal{S}_r$, decompose the matrix as in (2.1), let Π^t be a permutation that lists $\mathcal{S}_r$ first, and let P be an injection from $\mathbb{R}^{|\mathcal{S}_r|}$ to $\mathbb{R}^n$. Then fully solve $L_{11}\mathbf{u} = \lambda\mathbf{u}$. For each eigenspace U of distinct eigenvalues, see if there is a vector in the kernel of L_{21} within this subspace by solving

$$U^t L_{21}^t L_{21} U \mathbf{y} = \mu \mathbf{y}.$$

If there is any vector of coefficients $\mathbf{y}$ associated with $\mu = 0$, then $\Pi P U \mathbf{y}$ is a locally supported eigenvector and it should be included into Z. If there are any local eigenvectors supported by $\mathcal{S}_r$, then include all the other vectors into Q. Repeat this process for each set in the family. Lastly, augment Q to include columns of the identity for all vertices that are not part of the support of any local eigenvector detected in this process.

A.1. Algorithm 2: shared chain detection algorithm. Here we present an algorithm that detects sets of chains with the same length that hang off the same vertex. This algorithm detects shared chains of length $c = 1$ (shared leaves), $2, 3, \ldots, c_{max}$, where c_{max} is the maximal chain length. It is highly related to the first phase of a core detection algorithm given in [4]. The algorithm is guaranteed to be $\mathcal{O}(m)$ in cost.

Initially, all vertices i of degree 1 are assigned a value $g_i = 0$ that represents the number

ETNA
Kent State University
http://etna.math.kent.edu

Algorithm 3: Detecting hanging cliques.

```
input  : a connected, unweighted, and undirected graph G(V, E), max clique
         size c_max
output : collection of subsets {S_r}_{r=1}^R containing locally supported eigenvectors.
Set L_j = {i ∈ V : d_i = j} for j = 3, ..., c_max.
Initialized unvisited vertices, U ← V.
Set T = L_2.
r = 0.
for c = 3, ..., c_max do
    for i ∈ T do
        if i ∈ U then
            Let C^- ← {p ∈ (N_i ∩ U) : d_p < (c − 1)}
            Let C   ← {p ∈ (N_i ∩ U) : d_p = (c − 1)}
            Let C^+ ← {p ∈ (N_i ∩ U) : d_p > (c − 1)}
            if card(C ∪ C^+) = (c − 1) and card(C^+) < (c − 2) then
                if C ∪ C^+ is a clique then
                    r ← r + 1
                    S_r = C
                end
            end
            U ← U \ ({i} ∪ C^- ∪ C)
        end
    end
    T_new ← L_c.
    T ← T ∪ T_new.
end
R = r
```

of links below the i-th vertex. Start at each vertex i of degree 1 and go to the only vertex connected to i, which we label j. If j is degree 2, then set $g_j = g_i + 1$ and save it for the next iteration of the algorithm by placing it into a queue $\mathcal{T}$. If j has degree higher than 2, then look in its neighborhood for vertices p such that $d_p = d_i$ and $g_p = g_i$. If such a $p \neq i$ exists, then we have detected a set of shared chains. Repeat this process for the degree 2 nodes identified earlier. See Algorithm 2 for a complete description.

A.2. Algorithm 3: hanging clique detection algorithm. We describe an algorithm to detect cliques of size $c = 3, \ldots, c_{max}$ that hang off the edge of the graph with at most $c-2$ vertices connected to the rest of the graph. Initially, set $c = 3$ and put all vertices of degree $c-1$ into a queue. For each vertex i in the queue, check for a hanging c-clique by ensuring that all unvisited vertices in i's neighborhood have at least degree $c-1$ and at most $c-2$ of them have degree greater than $c-1$. If this test is passed, then make sure that all connections are present within i's neighborhood. If this test is passed, then the portion of this clique that hangs off the graph contains LSEVs. Next eliminate the vertex i and the members of its neighborhood with degree $c-1$ or less from further consideration. Repeat the process for $c \leftarrow c+1$. See Algorithm 3 for a complete description. The cost of this algorithm can be much worse than $\mathcal{O}(m)$ if c_{max} is set as a high value because checking for a clique of size c requires to look for $(c-1)(c-2)$ potential edges.

REFERENCES

[1] P. ARBENZ, U. HETMANIUK, R. LEHOUCQ, AND R. TUMINARO, *A comparison of eigensolvers for large-scale 3D modal analysis using AMG-preconditioned iterative methods*, Internat. J. Numer. Methods Engrg., 64 (2005), pp. 204–236.

[2] A. BANERJEE AND J. JOST, *On the spectrum of the normalized graph Laplacian*, Linear Algebra Appl., 428 (2008), pp. 3015–3022.

[3] A.-L. BARABÁSI AND R. ALBERT, *Emergence of scaling in random networks*, Science, 286 (1999), pp. 509–512.

[4] V. BATAGELJ AND M. ZAVERSNIK, *An $O(m)$ algorithm for cores decomposition of networks*, preprint, CoRR, cs.DS/0310049, 2003. http://arxiv.org/abs/cs.DS/0310049.

[5] T. BIYIKOGU, J. LEYDOLD, AND P. STADLER, *Laplacian Eigenvectors of Graphs*, Lecture Notes in Mathematics 1915, Springer, Heidelberg, 2007.

[6] S. BRIN AND L. PAGE, *The anatomy of a large-scale hypertextual web search engine*, Comput. Networks ISDN Systems, 30 (1998), pp. 107–117.

[7] F. BONCHI, P. ESFANDIAR, D. GLEICH, C. GRIEF, AND L. LAKSHAMANAN, *Fast matrix computations for pairwise and columnwise commute times and Katz scores*, Internet Math., 8 (2012), pp. 73–112.

[8] A. BORZÍ AND G. BORZÍ, *Algebraic multigrid methods for solving generalized eigenvalue problems*, Internat. J. Numer. Methods Engrg., 65 (2006), pp. 1186–1196.

[9] I. FARIA, *Permanental roots and the star degree of a graph*, Linear Algebra Appl., 64 (1985), pp. 255–265.

[10] F. FOUSS, A. PIROTTE, J. RENDERS, AND M. SAERENS, *Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation*, IEEE Trans. Knowl. Data Engrg., 19 (2007), pp. 355–369.

[11] D. Gleich, *The Spectre of the Spectrum*, Lecture given at the Indiana University, Bloomington, IN, and personal correspondence, 2011.

[12] V. HERNANDEZ, J. ROMAN, AND V. VIDAL, *SLEPc: a scalable and flexible toolkit for the solution of eigenvalue problems*, ACM Trans. Math. Software, 31 (2005), pp. 351–362.

[13] M. HEROUX, R. BARTLETT, V. HOWLE, R. HOEKSTRA, J. HU, T. KOLDA, R. LEHOUCQ, K. LONG, R. PAWLOWSKI, E. PHIPPS, A. SALINGER, H. THORNQUIST, R. TUMINARO, J. WILLENBRING, AND A. WILLIAMS, *An Overview of Trilinos*, Sand Report, SAND2003-2927, Sandia National Laboratories, Albuquerque, 2003.

[14] U. HETMANIUK, *A Rayleigh quotient minimization algorithm based on algebraic multigrid*, Numer. Linear Algebra Appl., 14 (2007), pp. 563–580.

[15] Y. KOREN, *Drawing graphs by eigenvectors: theory and practice*, Comput. Math. Appl., 49 (2005), pp. 1867–1888.

[16] D. KUSHNIR, M. GALUN, AND A. BRANDT, *Efficient multilevel eigensolvers with applications to data analysis tasks*, IEEE Trans. Pattern Anal. Mach. Intell., 32 (2010), pp. 1377–1391.

[17] I. LASHUK, M. E. ARGENTATI, E. OVCHINNIKOV, AND A. V. KNYAZEV, *Preconditioned eigensolver LOBPCG in hypre and PETSc*, in Proceedings of the 16th International Conference on Domain Decomposition Methods 2005, O. B. Widlund and D. E. Keyes, eds., Lect. Notes Comput. Sci. Eng., 55, Springer, Berlin, 2007, pp. 635–642.

[18] J. LESKOVEC, *Stanford network analysis platform*. http://snap.stanford.edu.

[19] B. LYON, *The Opte project*. http://www.opte.org.

[20] R. MERRIS, *Laplacian matrices of graphs: a survey*, Linear Algebra Appl., 197/198 (1994), pp. 143–176.

[21] ———, *Laplacian graph eigenvectors*, Linear Algebra Appl., 278 (1998), pp. 221–236.

[22] M. NEWMAN, *The Laplacian spectrum of graphs*, Ph.D. Thesis, Dept. of Math., University of Manitoba, Winnipeg, Canada, 2000.

[23] M. E. J. NEWMAN, *Networks: An Introduction*, Oxford University Press, Oxford, 2010.

[24] B. PARLETT, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, 1998.

[25] C. TSOURAKAKIS, *Fast counting of triangles in large real networks: algorithms and laws*, in ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining 2008, G. Fosca, ed., IEEE Conference Publication, IEEE, Piscataway, NJ, 2008, pp. 608–617.

[26] P. VANEK, J. MANDEL, AND M. BREZINA, *Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems*, Computing, 56 (1996), pp. 179–196.

[27] U. VON LUXBURG, *A tutorial on spectral clustering*, Stat. Comput., 17 (2007), pp. 395–416.

[28] U. VON LUXBURG, A. RADL, AND M. HEIN, *Getting lost in space: large sample analysis of the commute distance*, in Advances in Neural Information Processing Systems 23, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel and A. Culotta, eds., NIPS Foundation, La Jolla, CA, 2010, pp. 2622–2630.