On a time-frequency approach to translation on finite graphs

Matthew Begué
Norbert Wiener Center for Harmonic Analysis and Applications
University of Maryland, College Park
College Park, MD 20742
Email: begue@math.umd.edu

Abstract—The authors of [1] have used spectral graph theory to define a Fourier transform on finite graphs. With this definition, one can use elementary properties of classical time-frequency analysis to define time-frequency operations on graphs including convolution, modulation, and translation. Many of these graph operators have properties that match our intuition in Euclidean space. The exception lies with the translation operator. In particular, translation does not form a group, i.e., \( T_i T_j \neq T_{i+j} \). We prove that graphs whose translation operators exhibit semigroup behavior are those whose eigenvectors of the Laplacian form a Hadamard matrix.

I. INTRODUCTION

Graph theory has developed into a useful tool in applied mathematics as many modern data sets can be represented as a graph. In these data sets, vertices correspond to different sensor observations, or data points and edges represent connections, similarities, or correlations among those points. Some immediate examples include social network data, electricity networks, and images. In fact, in many dimension reduction applications, similarities, or correlations among those points. Some relevant definitions from spectral graph theory as well as basic properties of classical time-frequency analysis to define a time-frequency transform, convolution, modulation, and translation. Many of these graph operators have properties that match our intuition in Euclidean space. The exception lies with the translation operator. In particular, translation does not form a group, i.e., \( T_i T_j \neq T_{i+j} \). We prove that graphs whose translation operators exhibit semigroup behavior are those whose eigenvectors of the Laplacian form a Hadamard matrix.

II. BACKGROUND

A. Graph definitions

We denote a finite, undirected graph by \( G = (V,E) \) where \( V = \{x_1, \ldots, x_N\} \) denotes the vertex set and \( E \) denotes the edge set. The edge set, \( E \), consists of ordered pairs that correspond to edges on a graph. If there is an edge between points \( x \in V \) and \( y \in V \) then we write \( x \sim y \). Hence, \( E = \{(x,y) : x,y \in V \text{ and } x \sim y\} \). For any point \( x \in V \), we define the degree of \( x \), denoted \( d_x \), to be the number of edges connected to point \( x \). In an undirected graph, the edge set \( E \) is symmetric, that is, \( x \sim y \) implies \( y \sim x \). In other words, we do not distinguish an incoming vs. outgoing orientation to any edge.

We consider functions defined on the vertex set \( f : V \to \mathbb{R} \), \( x_n \mapsto f(x_n) \). We shall occasionally use the shorthand notation \( f(n) = f(x_n) \) which shall be clear in context. Since the cardinality of the vertex set is finite, \( |V| = N \), we shall sometimes write functions on graphs as vectors in \( \mathbb{R}^N \).

B. The Laplacian Operator

The main differential operator that we shall study is the Laplacian, denoted by \( \Delta \). The pointwise formulation of the unweighted Laplacian applied to a function \( f : V \to \mathbb{R} \) is defined as

\[
\Delta f(x) = \sum_{y \sim x} f(x) - f(y) \quad (\text{II.1})
\]

This is precisely the same formulation given in [8], [9] (up to a sign).

For a finite graph, Laplace’s operator can be represented as a matrix. For every point \( x \in V \), (II.1) gives a linear equation depending on function values at other points in \( V \). Each of these linear equations can be represented in the rows of the Laplacian matrix yielding

\[
L(i,j) = \begin{cases} 
    d_x, & \text{if } i = j \\
    -1, & \text{if } x_i \sim x_j \\
    0, & \text{otherwise}
\end{cases} \quad (\text{II.2})
\]

Let the matrix \( D \) denote the degree matrix which is the diagonal \( N \times N \) matrix given by \( D = \text{diag}(d_x) \). Let \( A \) denote the adjacency matrix, also \( N \times N \), where

\[
A(i,j) = \begin{cases} 
    1, & \text{if } x_i \sim x_j \\
    0, & \text{otherwise}
\end{cases}
\]

Then the unweighted Laplacian (II.2) can be written as

\[
L = D - A. \quad (\text{II.3})
\]

Remark 1. The matrix, \( L \), in (II.3) is the Laplacian matrix for unweighted graphs (each edge is assigned a weight of 1). Weighted graphs are more general and have Laplacian \( L = D - W \) where \( W \) is the 1’s in the adjacency are replaced with nonnegative weights \( \omega(i,j) \), and the degrees \( d_x \) become \( d_x = \sum_{j=1}^{N} \omega(i,j) \). While we consider unweighted graphs for this document, the results presented still hold for weighted graphs and their Laplacians.
Remark 2. Matrix \( L \) is called the unnormalized Laplacian to distinguish it from the normalized Laplacian, \( L = D^{-1/2}LD^{-1/2} \), used in some of the literature on graphs, e.g., [9]. We shall work exclusively with the unweighted Laplacian and shall henceforth just refer to it as the Laplacian.

Since \( L \) is a real symmetric matrix, it has nonnegative eigenvalues \( \{\lambda_k\}_{k=0}^{N-1} \) with associated orthonormal eigenvectors \( \{\varphi_k\}_{k=0}^{N-1} \). If the graph is connected then \( \lambda_0 = 0 \) and \( \lambda_k > 0 \) for all \( 1 \leq k \leq N - 1 \). We denote the conjugate transpose of a vector \( \varphi \) by \( \varphi^* \). Furthermore, let us define \( \Phi \) to be the \( N \times N \) orthogonal matrix whose \((k+1)\)th column is the vector \( \varphi_k \). The spectrum of the Laplacian, \( \sigma(L) \), is fixed but one’s choice of eigenvectors \( \{\varphi_k\}_{k=0}^{N-1} \) can vary. Throughout the document, we assume that the choice of orthonormal eigenvectors is fixed. Furthermore, since \( L \) is Hermitian, the eigenvectors can be selected to be entirely real-valued. Unless specifically noted, all of the following theory holds for real or complex eigenvectors.

Figure II.1 shows the first four nontrivial eigenfunctions of the Minnesota Road Network graph [10]. They correspond to the three lowest nonzero eigenvalues. As the eigenvalues increase, the eigenfunctions oscillate more.

![Fig. II.1. The first four nontrivial eigenfunctions (left to right) on the Minnesota Road Network graph. Blue corresponds to lower values while red corresponds to larger values.](image)

### III. GRAPH TIME-FREQUENCY OPERATIONS

#### A. The Graph Fourier Transform

The classical Fourier transform on the real line is the expansion of a function \( f \) in terms of the eigenfunctions of the Laplace operator, i.e. \( \hat{f}(\xi) = \langle f, e^{2\pi i \xi t} \rangle \). Analogously, we define the graph Fourier transform, \( \hat{f} : \sigma(L) \to \mathbb{R} \), of a function \( f : V \to \mathbb{R} \) as the expansion of \( f \) in terms of the eigenfunctions of the graph Laplacian. It is defined by

\[
\hat{f}(\lambda_l) = \langle f, \varphi_l \rangle = \sum_{n=1}^{N} f(n) \varphi_l^*(n). \tag{III.1}
\]

This definition of the graph Fourier transform was introduced by Vandergheynst et al. in [1], [11]. Notice that the graph Fourier transform is only defined on values of \( \sigma(L) \) which is a discrete finite set in \( \mathbb{R} \) and that the graph Fourier transform is completely dependent on the choice of Laplacian eigenvectors.

The inverse Fourier transform is then given by

\[
f(n) = \sum_{l=0}^{N-1} \hat{f}(\lambda_l) \varphi_l(n). \tag{III.2}
\]

If we consider the function \( f \) and \( \hat{f} \) as \( N \times 1 \) vectors, then (III.1) and (III.2) become

\[
\hat{f} = \Phi^* f \quad \text{and} \quad f = \Phi \hat{f}.
\]

Since, \( \Phi \) is a unitary matrix, the derivation of (III.2) is a direct result of (III.1). Furthermore, (III.2) implies that every function equals its “Fourier series”. In addition, \( \Phi \) being unitary immediately gives Parseval’s relation, i.e. \( \langle f, g \rangle = \langle \hat{f}, \hat{g} \rangle \), hence Plancherel’s theorem holds as well, i.e. \( \|f\|_2 = \|\hat{f}\|_2 \).

With the Fourier transform now defined on a graph, we exploit elementary time-frequency properties of real signals to derive graph convolution, modulation, and translation, as was done in [1].

#### B. Graph Convolution

For signals \( f, g \in L^2(\mathbb{R}) \) we define the convolution as \( f \ast g(t) = \int_{\mathbb{R}} f(u)g(t-u) \, du \). However, there is no clear analogue of translation in the graph setting. So we exploit the property of the convolution \( (f \ast g)(\xi) = \hat{f}(\xi)\hat{g}(\xi) \). Then by the inverse graph Fourier transform, (III.2), we can define convolution in the graph domain. For \( f, g : V \to \mathbb{R} \), we define the graph convolution of \( f \) and \( g \) as

\[
f \ast g(n) = \sum_{l=0}^{N-1} \hat{f}(\lambda_l)\hat{g}(\lambda_l)\varphi_l(n). \tag{III.3}
\]

**Proposition 3** ([1]). For \( \alpha \in \mathbb{R} \), and \( f, g, h : V \to \mathbb{R} \) then the graph convolution defined in (III.3) satisfies the following properties:

1. \( \hat{f} \ast \hat{g} = \hat{f} \hat{g} \).
2. \( \alpha(f \ast g) = (\alpha f) \ast g = f \ast (\alpha g) \).
3. **Commutativity**: \( f \ast g = g \ast f \).
4. **Distributivity**: \( f \ast (g + h) = f \ast g + f \ast h \).
5. **Associativity**: \( (f \ast g) \ast h = f \ast (g \ast h) \).

The proofs of these properties are relatively straightforward and some follow immediately from the definition, (III.3).

If we want to express the graph convolution as a vector then (III.3) is equivalent to \( \Phi^*(\hat{f} \odot \hat{g}) \) where \( \odot \) is the pointwise multiplication operator (written as \( \ast \) in MATLAB). In MATLAB, (recalling that \( \hat{f} = \Phi^* f \)) one would express the vector \( f \ast g \) as

\[
\Phi^*\Phi^* f \ast g \text{ or } \Phi^*(\Phi^* f \ast g).
\]
Example 4. Let us introduce the function $g_0 : V \to \mathbb{R}$ by first defining it in the Fourier (frequency) domain. We set $g_0(\lambda_l) = 1$ for all $l = 0, \ldots, N-1$. Then the values of $g_0$ can be obtained by taking the inverse Fourier transform (III.2) giving $g_0(n) = \sum_{l=0}^{N-1} \varphi_l(n)$ (in vector notation, $g_0 = \Phi \ast \mathbb{I}_{N \times 1}$, where $\mathbb{I}_{N \times 1}$ is the $N \times 1$ vector of all ones). The function $g_0$ is shown in Figure III.1. Then as a result of Proposition 3, we have
\[
\begin{align*}
  f(n) &= \sum_{l=0}^{N-1} \hat{f}(\lambda_l) \varphi_l(n) \\
&= \sum_{l=0}^{N-1} \hat{f}(\lambda_l) g_0(\lambda_l) \varphi_l(n) = f \ast g_0(n).
\end{align*}
\]
In other words, $f \ast g_0 = f$, so convolution with the function $g_0$ is the identity.

![Fig. III.1. An overhead (left) and side (right) view of the function $g_0$ which is the unique function that acts as the identity with convolution on the Minnesota graph [10].](image)

C. Graph Modulation

Motivated by the fact that in Euclidean space, modulation of a function is multiplication of a Laplacian eigenfunction, we define for any $k = 0, 1, \ldots, N-1$ the graph modulation operator $M_k : \mathbb{R}^N \to \mathbb{R}^N$ as
\[
(M_k f)(n) = \sqrt{\mathcal{N}} f(n) \varphi_k(n).
\]
We normalize by $\sqrt{\mathcal{N}}$ in (III.5) so that $M_0$, that is, convolution with $\varphi_0 \equiv 1/\sqrt{\mathcal{N}}$, is the identity operator.

An important remark is that in the classical case, modulation in the time domain represents translation in the frequency domain, i.e. $\hat{M}_\xi f(\omega) = \hat{f}(\omega - \xi)$. The graph modulation in general does not exhibit this property due to the fact that the discrete spectral domain does not have a clear structure. However, it is worthy to notice the special case if $\hat{g}(\lambda_l) = \delta_0(\lambda_l)$, where
\[
\delta_0(\lambda) = \begin{cases} 
1, & \text{if } \lambda = 0 \\
0, & \text{if } \lambda \neq 0,
\end{cases}
\]
then
\[
\begin{align*}
  \hat{M}_k g(\lambda_l) &= \sum_{n=1}^{N} \varphi_l^*(n)(M_k g)(n) \\
&= \sum_{n=1}^{N} \varphi_l^*(n) \sqrt{\mathcal{N}} \varphi_k(n) \frac{1}{\sqrt{\mathcal{N}}} = \delta(l, k)
\end{align*}
\]
by the orthonormality of $\{\varphi_k\}_{k=0}^{N-1}$.

We can express the operator $M_k$ as a diagonal matrix
\[
M_k = \begin{pmatrix} 
\varphi_k(1) & 0 \\
0 & \ddots \\
0 & \varphi_k(N)
\end{pmatrix}
\]
and in the language of MATLAB we have
\[
\text{Mk = diag}(\Phi(:,k)).
\]

D. Graph Translation

In the classical setting, the translation operator, $T_u$, translates the function $f : \mathbb{R} \to \mathbb{R}$ by vector $u$. The translation action can be expressed as a convolution with $\delta_u$ where
\[
\delta_u(x) = \begin{cases} 
1, & \text{if } x = u \\
0, & \text{if } x \neq u.
\end{cases}
\]

Then in $\mathbb{R}$, by exploiting (III.3) we have
\[
(T_u f)(t) = f(t - u) = (f \ast \delta_u)(t)
\]
\[
= \int_{\mathbb{R}} \hat{f}(k) \delta_u(k) \varphi_k(t) dk = \int_{\mathbb{R}} \hat{f}(k) \varphi_u^*(u) \varphi_k(t) dk
\]
since $\delta_u(k) = \int_{\mathbb{R}} \delta_u(x) \varphi_x^*(x) dx = \varphi_k(u)$.

Motivated by this example, for any $f : V \to \mathbb{R}$ we can define the graph translation operator, $T_u$, via the graph convolution of the Dirac delta centered at the $i$th vertex:
\[
(T_i f)(n) = \sqrt{\mathcal{N}} (f \ast \delta_i)(n) = \sqrt{\mathcal{N}} \sum_{l=0}^{N-1} \hat{f}(\lambda_l) \varphi_l^*(i) \varphi_l(n). 
\]

We can express $T_i f$ in matrix notation as follows:
\[
\begin{pmatrix} 
\varphi_0^*(i) \varphi_0(1) & \cdots & \varphi_{N-1}^*(i) \varphi_{N-1}(1) \\
\vdots & \ddots & \vdots \\
\varphi_0^*(i) \varphi_0(N) & \cdots & \varphi_{N-1}^*(i) \varphi_{N-1}(N)
\end{pmatrix} 
\begin{pmatrix} 
\hat{f}(\lambda_0) \\
\vdots \\
\hat{f}(\lambda_{N-1})
\end{pmatrix}
= \Phi \boxtimes (\varphi_0^*(i) \cdots \varphi_{N-1}^*(i)) \Phi^* f = \frac{1}{\sqrt{\mathcal{N}}} T_i f
\]
where we introduce the binary operation $\boxtimes$ to signify element-wise multiplication. For example, if
\[
A = \begin{pmatrix} a_{11} & a_{12} \\
a_{21} & a_{22} \end{pmatrix}, \quad b = (b_1, b_2)
\]
then $A \boxtimes b$ multiplies the row vector $b$ component-wise with each row of $A$; that is
\[
A \boxtimes b = \begin{pmatrix} a_{11} b_1 & a_{12} b_2 \\
a_{21} b_1 & a_{22} b_2 \end{pmatrix}.
\]
Similarly if $b'$ was instead a column vector, i.e.
\[
A = \begin{pmatrix} a_{11} & a_{12} \\
a_{21} & a_{22} \end{pmatrix}, \quad b' = \begin{pmatrix} b_1 \\
b_2 \end{pmatrix},
\]
then $A \boxtimes b'$ multiplies the column vector $b'$ component-wise with each column of $A$; that is
\[
A \boxtimes b' = \begin{pmatrix} a_{11} b_1 & a_{12} b_1 \\
a_{21} b_2 & a_{22} b_2 \end{pmatrix}.
\]
The MATLAB command for $A \boxplus b$ is \texttt{bsxfun(@times,A,b)}.

Therefore for any $i = 1, \ldots, N$ the MATLAB command for the translation operator as

\[ T_i = \text{sqrt}(N) \ast \text{bsxfun}(\times, \Phi, \Phi(i,:)) \ast \Phi'/2. \]

There are some properties of graph translation that agree with our intuition with classical Euclidean translation. For example, the graph translation operator is distributive with the convolution and they commute among themselves.

**Proposition 5** (\cite{1}). For any $f, g : V \to \mathbb{R}$ and for any $i, j \in \{1, 2, \ldots, N\}$ then

1. $T_i(f \ast g) = (T_i f) \ast g = f \ast (T_i g)$.
2. $T_i T_j f = T_j T_i f$.

**Corollary 6.** As long as the eigenvectors, $\{\varphi_k\}_{k=1}^{N-1}$ are real-valued then for any $i, n \in \{1, \ldots, N\}$ and for any function $g : V \to \mathbb{R}$ we have

\[ T_i g(n) = T_n g(i). \]

**Proof.** By definition $(T_i f)(n) = \sqrt{N} \sum_{l=0}^{N-1} \hat{f}(\lambda_l) \varphi_l^*(i) \varphi_l(n)$.

But by assumption, the eigenbasis is entirely real-valued. Therefore, for any $i, n \in \{1, \ldots, N\}$ and any $l \in \{0, \ldots, N-1\}$, then $\varphi_l^*(i) \varphi_l(n) = \varphi_l(i) \varphi_l(n)$, which proves the theorem. \qed

**Corollary 7.** Let $\alpha(0) \in \{1, \ldots, N\}$ and $\alpha = (\alpha(1), \alpha(2), \ldots, \alpha(K))$ where $\alpha(j) \in \{1, \ldots, N\}$ for $1 \leq j \leq K$. We let $T_{\alpha}$ denote the composition $T_{\alpha_1} \circ T_{\alpha_2} \circ \cdots \circ T_{\alpha_K}$. Then for any $f : V \to \mathbb{R}$ and any permutation, $\sigma$, of the set $\{0, 1, \ldots, K\}$, we have $T_{\alpha} f(a_0) = T_{\beta} f(\sigma(0))$ where $\beta = (\alpha(\sigma(1)), \ldots, \alpha(\sigma(K)))$.

**Proof.** Define an equivalence relation among the space $\{1, \ldots, N\}^{K+1}$ by $(a_0, a_1, a_2, \ldots, a_K)$ if and only if $T_{a_1} \circ \cdots \circ T_{a_K} f(a_0) = T_{b_0} \circ \cdots \circ T_{b_K} f(b_0)$.

By Corollary 6, $(a_0, a_1, a_2, \ldots, a_K) \cong (a_1, a_0, a_3, \ldots, a_K)$, i.e. $\sigma_1$ is the permutation $(1, 2)$. In general, we write $\sigma_i$ to denote the permutation $(i, i+1)$.

By Proposition 5, $b, (a_0, a_1, a_2, \ldots, a_K) \cong (a_i, a_0, a_3, \ldots, a_K)$ for any $i = 2, 3, \ldots, K - 1$.

We now have that any permutation $\sigma_i$ for $i = 1, \ldots, K - 1$ preserves equivalence. This collection of $K - 1$ transpositions allow for any permutation, $\sigma$, which proves the corollary. \qed

However, the niceties end here; many of the properties of Euclidean translation do not hold with this definition of graph translation. Unlike in the classical setting, graph translation is not an isometric operation, that is, $\|T_i f\|_2 \neq \|f\|_2$. However, we do have the following estimate.

**Lemma 8** (\cite{1}). For any $f : V \to \mathbb{R}$,

\[ |\hat{f}(0)| \leq \|T_i f\|_2 \leq \sqrt{N} \max_{l \in \{0, \ldots, N-1\}} |\varphi_l(i)| \|f\|_2 \leq \sqrt{N} \max_{l \in \{0, \ldots, N-1\}} \|\varphi_l\|_\infty \|f\|_2. \tag{III.7} \]

**Proposition 9.** In general, the operator $T_i$ is not injective and therefore not invertible.

**Proof.** Suppose there exist some $k \in \{1, \ldots, N-1\}$ and some $i \in \{1, \ldots, N\}$ for which $\varphi_k(i) = 0$. Then if $f = \varphi_k$ we have $f(\lambda_l) = \delta_k(l)$ and

\[ T_i f(n) = \sqrt{N} \sum_{l=0}^{N-1} \hat{f}(\lambda_l) \varphi_l^*(i) \varphi_l(n) = \sqrt{N} \varphi_k^*(i) \varphi_k(n) = 0 \]

for all $n = 1, \ldots, N$. Therefore, there exist nonzero $f : V \to \mathbb{R}$ such that $T_i f = 0$, hence $T_i$ is not injective and therefore not invertible. \qed

**IV. GRAPH TRANSLATION AS A SEMIGROUP**

The discrepancies between classical and graph translation continue. For arbitrary graphs, we do not have the collection of translation operators forming a group, i.e., in general $T_i T_j \neq T_{i+j}$, unlike the case in the Euclidean setting. There are some exceptions to this in the very special case of shift-invariant graphs such as the cycle graph (The eigenvector matrix for the cycle graph can be chosen to equal the Discrete Fourier Transform matrix). In fact, we cannot even assert that the translation operators form a semigroup, i.e. $T_i T_j \neq T_k$ for some semigroup operator $\bullet : \{1, \ldots, N\} \times \{1, \ldots, N\} \to \{1, \ldots, N\}$.

**Theorem 10.** Consider the graph, $G(V, E)$, with real eigenvector matrix $\Phi = [\varphi_1 \cdots \varphi_{N-1}]$. Graph translation on $G$ is a semigroup, i.e. $T_i T_j = T_{i+j}$, for some semigroup operator $\bullet : \{1, \ldots, N\} \times \{1, \ldots, N\} \to \{1, \ldots, N\}$, only if $\Phi = (1/\sqrt{N}) H$, where $H$ is a Hadamard matrix.

**Proof.** i. We first show that graph translation on $G$ is a semigroup, i.e. $T_i T_j = T_{i+j}$ for some semigroup operator $\bullet : \{1, \ldots, N\} \times \{1, \ldots, N\} \to \{1, \ldots, N\}$, if and only if $\sqrt{N} \varphi_l(i) \varphi_l(j) = \varphi_l(i \bullet j)$ for all $l = 0, \ldots, N - 1$. By repeating the calculations in the proof of Proposition 5, we have

\[ T_i T_j f(n) = N \sum_{l=0}^{N-1} \hat{f}(\lambda_l) \varphi_l^*(j) \varphi_l^*(i) \varphi_l(n) \]

and by definition

\[ T_k f(n) = \sqrt{N} \sum_{l=0}^{N-1} \hat{f}(\lambda_l) \varphi_l^*(k) \varphi_l(n). \]

Therefore $T_i T_j f = T_{i+j} f$ for any function $f : V \to \mathbb{R}$ if and only if $\sqrt{N} \varphi_l(i) \varphi_l(j) = \varphi_l(i \bullet j)$ for all $l = 0, \ldots, N - 1$. ii. We show next that $\sqrt{N} \varphi_l(i) \varphi_l(j) = \varphi_l(i \bullet j)$ for all $l = 0, \ldots, N - 1$ only if the eigenvectors are constant amplitude. Assume $\sqrt{N} \varphi_l(i) \varphi_l(j) = \varphi_l(i \bullet j)$, which, in particular, implies $\sqrt{N} \varphi_l(i) \varphi_l(i) = \sqrt{N} \varphi_l(i \bullet i)$. Suppose that $|\varphi_l(a_1)| < 1/\sqrt{N}$ for some $a_1 \in \{1, \ldots, N\}$ and for some $l \in \{0, \ldots, N - 1\}$. Then $\sqrt{N} \varphi_l(a_1)^2 < \varphi_l(a_1)$ and so $a_1 \bullet a_1 = a_2 \neq a_1$. Then by the same argument, $\varphi_l(a_2) = \sqrt{N} \varphi_l(a_1)^2 < 1/\sqrt{N}$ and hence $a_2 \bullet a_2 = a_3 \neq a_2$. This procedure can be repeated producing an infinite number of unique indices $\{a_l\}$ on a graph, $G$, with only $N < \infty$ nodes, a contradiction. A similar argument gives a contradiction if $|\varphi_l(i)| > 1/\sqrt{N}$ for any $l, i$. Therefore, the graph translation operators form a semigroup only if $|\varphi_l(i)| = 1/\sqrt{N}$ for all $l = 0, 1, \ldots, N - 1$ and $i = 1, \ldots, N$. Since $\Phi$ is an orthogonal
matrix, i.e. $\Phi \Phi^* = \Phi^* \Phi = I$, then $\Phi = (1/\sqrt{N}) H$, where $H$ is a Hadamard matrix.

**Remark 11.** a) The order of a Hadamard matrix must be 1, 2, or a multiple of 4. It remains an open problem to show that there exists a Hadamard matrix of order equal to any multiple of 4, [12].

b) It is shown in [13, Theorem 5] that if $\Phi = (1/\sqrt{N}) H$ for Hadamard $H$, then the spectrum of the Laplacian, $\sigma(L)$, must be entirely even integers.

c) The converse to Theorem 10 does not necessarily hold. That is, if the eigenvector matrix $\Phi$ is a renormalized Hadamard matrix, then the translation operators on $G$ need not form a semigroup. For example, consider the real Hadamard matrix, $H$, of order 12 given by (IV.1). Then the second and third columns multiplied componentwise equals the vector $[1, -1, 1, -1, -1, 1, 1, -1, 1, 1, -1, -1]^T$, which does not equal any of the columns of $H$.

What kinds of graphs have a Hadamard eigenvector matrix? The authors of [13] prove that the complete graph on $n$ vertices, $K_n$, is one such graph.

**Theorem 12** ([13]). Suppose $H$ is a real Hadamard matrix of order $N$. Then $1/\sqrt{N} H$ is an eigenvector matrix for the complete graph of $N$ vertices, $K_N$.

**Proof.** It is a standard result (see [9]) that the complete graph has Laplacian matrix $L = NI - J$, where $J$ is the matrix with all entries 1, with eigenvalues $\lambda_0 = 0$ and $\lambda_i = N$ for $i = 1, \ldots, N - 1$. Let $D$ be the diagonal matrix $D(i,i) = \lambda_i$.

We can write the $N \times N$ matrix $H$ as

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 \end{bmatrix},$$

where the first equality holds from the property of $H$ being Hadamard. Additionally, one can compute $H D H^T = N \tilde{H} H^T$.

Thus we have

$$\left( \frac{1}{\sqrt{N}} \right) D \left( \frac{1}{\sqrt{N}} \right)^{-1} = \left( \frac{1}{N} \right) H D H^T = NI - J = L,$

which completes the proof.

Characterizing non-complete Hadamard graphs remains an open problem and potential area for future work.

**ACKNOWLEDGMENT**

The author would like to thank Professor Kasso A. Okoudjou for his guidance and mentorship, as well as for introducing him to [1]. The author also thanks the Norbert Wiener Center for Harmonic Analysis and Applications for its support.

**REFERENCES**