The Spectral Graph Wavelet Transform: Fundamental Theory and Fast Computation

David K. Hammond, Pierre Vandergheynst, Rémi Gribonval

The Spectral Graph Wavelet Transform : Fundamental theory and fast computation.

David K. Hammond and Pierre Vandergheynst and Rémi Gribonval

Abstract The spectral graph wavelet transform (SGWT) defines wavelet transforms appropriate for data defined on the vertices of a weighted graph. Weighted graphs provide an extremely flexible way to model the data domain for a large number of important applications (such as data defined on vertices of social networks, transportation networks, brain connectivity networks, point clouds, or irregularly sampled grids). The SGWT is based on the spectral decomposition of the $N \times N$ graph Laplacian matrix $\mathscr{L}$, where N is the number of vertices of the weighted graph. Its construction is specified by designing a real-valued function g which acts as a bandpass filter on the spectrum of $\mathscr{L}$, and is analogous to the Fourier transform of the "mother wavelet" for the continous wavelet transform. The wavelet operators at scale s are then specified by $T_g^s = g(s\mathscr{L})$, and provide a mapping from the input data $f \in \mathbb{R}^N$ to the wavelet coefficients at scale s. The individual wavelets $\psi_{s,n}$ centered at vertex n, for scale s, are recovered by localizing these operators by applying them to a delta impulse, i.e. $\psi_{s,n} = T_g^s \delta_n$. The wavelet scales may be discretized to give a graph wavelet transform producing a finite number of coefficients. In this work we also describe a fast algorithm, based on Chebyshev polynomial approximation, which allows computation of the SGWT without needing to compute the full set of eigenvalues and eigenvectors of $\mathscr{L}$.

David K. Hammond
Oregon Institute of Technology - Portland Metro e-mail: david.hammond@oit.edu

Pierre Vandergheynst
Ecole Polytechnique Federale de Lausanne e-mail: pierre.vandergheynst@epfl.ch

Rémi Gribonval
Univ Rennes, Inria, CNRS, IRISA e-mail: remi.gribonval@inria.fr

1 Introduction

Nearly all interesting scientific and engineering problems involve analyzing data. In many cases, data can be described as a real valued function defined on some domain. For example, data sets such as audio recordings, digital photographs, and digital videos may be represented as real valued functions defined on a one, two, or three dimensional Euclidean domains, respectively. Similarly, discrete time-series data may be modeled as a real valued function whose domain is a subset of the integers. As these examples illustrate, many common types of data are defined on domains which are regularly sampled subsets of some Euclidean space. A huge body of signal processing and analysis algorithms have been developed for signals that are defined on such regularly sampled Euclidean domains.

However, a large number of interesting data sets are defined on irregular domains that do not correspond to regularly sampled Euclidean domains. Examples of this included data defined on networks, or on point clouds, or at irregularly sampled points of Euclidean domains. Many topologically complex data domains can be profitably modeled as weighted graphs : i.e. sets of vertices that are connected by edges that each have a non-negative weight, or connection "strength" defined. For some applications, the underlying graph structure for the data domain may be clear. This would be the case for example for analyzing data (such as income, preference for something, or any other scalar value) defined for individuals on a social network, where the underlying graph structure models the relationship strength between individuals. For data defined on point clouds or for irregularly sampled data, generating the underlying graph structure may be calculated in a number of ways based on the proximity of the data points, such as for example using the k-nearest neighbors.

Many signal processing and analysis methods employ some type of transform of the original data, where processing or analysis is performed on the coefficients of the transformed data rather than on the signal in its original domain. A large class of such transforms are linear, where the coefficients of the transform are given by taking inner products of the original signal with some set of transform vectors. For signals defined on regular Euclidean domains, commonly used examples include the Fourier transform, the discrete cosine transform, windowed Fourier transforms, and a large number of different Wavelet transforms. Signal transforms are useful as certain analysis or processing problems may be easier to express, or more powerful, when done in the coefficient domain than in the original signal domain. As a trivial example, the problem of estimating the power spectrum of a discrete time signal averaged over a specified frequency range takes some time to write down in the time domain, but is very easy to express (as an average of the magnitudes of a subset of Fourier coefficients) given the Fourier transform of the signal.

Wavelet transforms in particular have proven to be very effective for a wide variety of signal processing and analysis problems. Wavelet transforms have the property that the transform vectors (in this case called wavelets) are well localized in both space and frequency domains, and are self similar - i.e. related to each other by translation and dilation. Wavelets may be designed so that they provide a sparse representation of signals that consist of relatively smooth regions separated by lo-

calized singularities, such as edges for 2d images, or localized jump discontinuities for 1d signals. Much of the power of many wavelet-based signal processing algorithms arises from exploiting this signal sparsity. Since their initial introduction in the 1980's [1, 2, 3, 4, 5], Wavelet transforms have been very successfully employed for a wide range of signal and image processing applications, including denoising, [6, 7, 8, 9, 10, 11], compression [12, 13, 14, 15, 16] , and inverse problems such as deconvolution [17, 18, 19, 20, 21, 22]. We have included only a sampling of the enormous body of literature in the references given above.

The demonstrated effectiveness of Wavelet transforms, combined with the growing desire to process data defined on non-Euclidean domains, motivates the adaptation of the Wavelet transform to data defined on weighted graphs. The work described in this chapter details one such approach for constructing multiscale Wavelet transforms for data defined on the vertices of weighted graphs. Our approach assumes that all relevant information about the weighted graph is encoded in the symmetric Adjacency matrix A, i.e. no other information about the meaning of the vertices or relationships between them apart from what is stored in the Adjacency matrix is used. Accordingly, for any specific application problem the design of the underlying edge weights for the graph forms a crucial part of determining the overall wavelet transformation. The design choices for constructing the graph weights exactly correspond to modeling the underlying data domain, and thus optimal choices may be highly application dependent. In this chapter we illustrate relatively simple examples of computing a appropriate weighted graph for data defined on point clouds, and on irregularly sampled grids.

In general, one may expect modeling the data domain by a weighted graph to be useful whenever the relationships one may describe between vertices with a weighted graph interact with the underlying process which generated the data. For example, if one were analyzing rates of some disease among different cities, it may be reasonable to assume that infection could be propagated by individuals travelling from one city to another. In this case, using a weighted graph representing a transportation network between the different cities may be helpful. Similarly, if one were analyzing data indicating individuals opinions (favorable or unfavorable) of some particular political candidate, it is reasonable to assume that individuals opinions are affected by discussions with their friends. Accordingly, knowledge of a weighted graph representing acquaintance or friendship relationships between individuals may be useful for analyzing such data.

Classical wavelet analysis is based on the idea of taking a single "mother wavelet" function, and generating an entire set of Wavelet atoms by translating and dilating the mother wavelet. Wavelet coefficients for a given signal are then produced by taking the inner products of the signal with these wavelet atoms at different scales and spatial locations. The success of this classical construction is based on the ability to perform arbitrary dilation and translation operations. On an arbitrary weighted graph, however, it is not possible to define dilation and translation as may be done on Euclidean spaces such as the real line.

The construction described in this chapter resolves this by using the spectral decomposition (i.e. eigenvectors and eigenvalues) of the graph Laplacian matrix $\mathscr{L}$.

Spectral graph theory [23] enables the definition of a Fourier transform for data defined on the vertices of weighted graphs, where the graph Fourier coefficients are given by the inner product of the signal with the eigenvectors of the graph Laplacian. The Spectral Graph Wavelet Transform (SGWT) described here is obtained by considering the mapping from data to coefficients for the classical continuous wavelet transform in the Fourier domain, and constructing the analogous operations using the graph Fourier transform. The SGWT design requires specifying a real-valued kernel function g. This kernel function is used to define the wavelet operators at scale s (i.e. the mappings from the signal to the wavelet coefficients at scale s) as $T_g^s = g(s\mathscr{L})$. The wavelet coefficients at scale s for an input signal $f(t)$ are then given by $T_g^s f$. The individual graph wavelets themselves are obtained by localizing these wavelet operators by applying them to a delta impulse at a single vertex. We employ the notation $\psi_{s,m}$ to denote the wavelet at scale s centered at vertex m, the previous notion implies that $\psi_{s,m} = T_g^s \delta_m$ where δ_m is a signal with zero values at every vertex except m, and unit value at vertex m. The SGWT coefficients are also the inner products of the original data with these wavelets $\psi_{s,m}$.

This chapter describes the basic theory of the SGWT, and gives implementation details and example images illustrating the Wavelets and properties of the overall transform. We show that the SGWT (without discretizing the scale parameter) is analogous to the classical continuous wavelet transform and, subject to an admissibility condition on the kernel function g, may be formally inverted using a similar integral formula. A discrete transform may be obtained by sampling the scale parameter at a discrete set of values, giving a finite number of coefficients organized in distinct wavelet subbands. In this case the SGWT is an overcomplete transform, and we describe how to calculate the corresponding frame bounds.

As the SGWT is defined using a the graph Fourier transform, straightforward computation of the transform requires computing the full set of eigenvectors and eigenvalues of the graph Laplacian $\mathscr{L}$. This limitation would render computing the SGWT infeasible for graphs larger than several thousand vertices, which would severely limit its applicability. In this chapter we describe a method for computing the SGWT based on Chebyshev polynomial approximation of the rescaled kernels $g(s\lambda)$, which does not require explicitly computing the full set of eigenvectors of the graph Laplacian matrix. In particular, this polynomial approximation approach uses $\mathscr{L}$ only through matrix-vector multiplication, and is thus especially computationally efficient for sparse graphs, where only a small number of elements of $\mathscr{L}$ are nonzero. The discrete SGWT may be inverted using the pseudoinverse. We show that this may be done by conjugate-gradients, in a way that is compatible with the Chebyshev polynomial approximation scheme for applying the forward transform.

1.1 Related Work

Other attempts at defining wavelet transforms on graphs have been developed that do not employ spectral graph theory. One approach used by Crovella and Kolaczyk

[24] for analyzing computer network traffic was based on the n-hop distance, where wavelets were defined such that the value of the wavelet at vertex m that was centered at vertex n depended on the n-hop distance from vertex m to n. These functions were chosen so that the wavelets were zero mean. These wavelets were constructed for binary graphs (i.e. making no use of edge weights), additionally no study of the invertibility of the resulting transform was made. These wavelets were used by Smalter et. al [25] as features helping to distinguish chemical structures, as part of a machine learning approach for virtual drug screening.

In [26], Jansen et. al. developed a lifting-based approach for multiscale representation of data defined on graphs. This lifting procedure is based on using the weighted average of each vertices neighbors for the lifting prediction step, where the weightings are based on a set of distances assigned to each edge (playing the part of reciprocals of edge weights), in their paper these edge distances were derived from original Euclidean distances for graphs that arise from irregular sampling of Euclidean space. In contrast with the methods described in this chapter, this lifting scheme is defined directly in the vertex domain, and does not employ spectral graph theory.

Several works have considered wavelet transforms for data defined on trees (i.e. graphs with no loops). These include [27], which developed an adaptation of the Haar wavelet transform appropriate for data defined on rooted binary trees. The treelet transform [28] extended this, including automatic construction of trees for multivariate data.

The "Diffusion Wavelets" of Maggioni and Coifman [29] constructs a wavelet transform based on compressed representations of dyadic powers of a diffusion operator T, which may be flexibly specified. The diffusion wavelets construction involves repeated application of the diffusion operator T, somewhat analogously to how our construction is parametrized by the choice of the graph Laplacian operator $\mathscr{L}$. A key difference between the Diffusion Wavelets and the Spectral Graph Wavelets described here is that the Diffusion Wavelets are designed to be orthonormal. The Diffusion Wavelets approach is based on first identifying the approximation spaces produced by differences of dyadic powers of the operator T; wavelets are produced by locally orthogonalizing these approximation spaces. Our approach is conceptually simpler, and yields an overcomplete representation rather than an orthogonal transform.

The "Diffusion polynomial frames" developed by Maggioni and Mhaskar [30] builds multiscale transforms in a more general quasi-metric measure space setting using polynomials of a differential operator, in a manner that is closely related to our Chebyshev polynomial approximation for computing the SGWT. Geller and Mayeli [31] construct wavelets on differentiable manifolds employing scaling defined by an operator of the form tLe^{-tL}, where L is the manifold Laplace-Beltrami operator. Wavelets are obtained by localizing this operator by applying it to a delta impulse, similar to our theory. However, their work is not directly comparable to ours as it is constructed for functions defined on smooth manifolds.

The work described in this chapter was originally published in [32]. For completeness, we note that since its original publication the SGWT and the related poly-

nomial approximation scheme has been used by a number of authors. We provide here references to a sampling of these applications, include learning dictionaries for signal representation on graphs [33], analysis of cortical thickness measurements [34], shape analysis and surface alignment [35], multiscale community detection [36], 3d mesh compression [37]

2 Classical Continuous Wavelet Transform

As the design of the SGWT is based on examining the the classical Continuous wavelet transform (CWT) in the Fourier domain, we first give an overview of the CWT appropriate for representing $L^2(\mathbb{R})$, i.e. the space of square-integrable signals on the real line.

The CWT is generated by choosing a single "mother" wavelet $\psi(t)$, and then forming a continuous family of wavelets by translating and dilating the mother wavelet. Specifically, for translation by a and dilation by factor $s > 0$, we have the wavelet

$$\psi_{s,a}(t) = \frac{1}{s}\psi\left(\frac{t-a}{s}\right) \tag{1}$$

The factor $\frac{1}{s}$ in front was chosen so that the wavelets all have the same L_1 norm, i.e. $\int_{-\infty}^{\infty}|\psi_{s,a}(t)|dt = \int_{-\infty}^{\infty}|\psi(t)|dt$. The wavelet coefficients are given by the inner products of these wavelets with the signal $f(t)$, as

$$W_f(s,a) = \int_{-\infty}^{\infty}\frac{1}{s}\psi^*\left(\frac{t-a}{s}\right)f(t)dt \tag{2}$$

This expression defines the mapping from the original signal $f(t)$ to the set of wavelet coefficients $W_f(s,a)$. An interesting feature of the CWT is that a single variable function $f(t)$ is represented by the coefficients $W_f(s,a)$ which depend on two parameters s and a. We say the transform may be inverted if it is possible to recover the function $f(t)$ from knowledge of the coefficients $W_f(s,a)$. It has been shown [1] that this is possible if the mother wavelet $\psi(t)$ satisfies the admissibility condition

$$\int_0^{\infty}\frac{|\hat{\psi}(\omega)|^2}{\omega}d\omega = C_\psi < \infty \tag{3}$$

One consequence of the admissibility condition is that for continuously differentiable $\psi(t)$, it must hold that $\hat{\psi}(0) = \int\psi(t)dt = 0$, so $\psi(t)$ must have zero mean. If $\psi(t)$ is admissible, the CWT may be formally inverted as

$$f(t) = \frac{1}{C_\psi}\int_0^{\infty}\int_{-\infty}^{\infty}W_f(s,a)\psi_{s,a}(t)\frac{dads}{s}. \tag{4}$$

Directly forming the analogue of equation 1 on weighted graphs is problematic, as it is unclear how to implement arbitrary dilation or translation on an ar-

bitrary weighted graph. We will show next that for the CWT, scaling can be defined in the Fourier domain, which does provide an expression we can extend to weighted Graphs. We first consider the CWT for discretized set of scale values, where the translation parameter is left continuous. For each value s, we let T^s represent the mapping from signal $f(t)$ to wavelet coefficients $W_f(s,a)$, so that $(T^s f)(a) = W_f(s,a)$ where a is considered as the independent variable.

For convenience in the following, define the scaled, time-reversed wavelet $\overline{\psi}_s$ via

$$\overline{\psi}_s(t) = \frac{1}{s}\psi^*\left(\frac{-t}{s}\right). \tag{5}$$

One may see then that the operator T^s acts on any signal by convolution with $\overline{\psi}_s$. Specifically, we have

$$\begin{aligned}(T^s f)(a) &= \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{t-a}{s}\right) f(t)dt \\ &= \int_{-\infty}^{\infty} \overline{\psi}_s(a-t)f(t)dt \\ &= (\overline{\psi}_s \star f)(a)\end{aligned} \tag{6}$$

By taking the Fourier transform and applying the convolution theorem we see that

$$\widehat{T^s f}(\omega) = \widehat{\overline{\psi}}_s(\omega)\widehat{f}(\omega) \tag{7}$$

Using the properties of the Fourier transform and Equation (5) shows that

$$\widehat{\overline{\psi}}_s(\omega) = \widehat{\psi^*}(s\omega) \tag{8}$$

Inserting Equation 8 into Equation 7 and inverting the Fourier transform gives

$$(T^s f)(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega t}\widehat{\psi}^*(s\omega)\widehat{f}(\omega)d\omega \tag{9}$$

Critically, s appears above only in the argument of $\widehat{\psi}^*$, which is defined in the Fourier domain rather than the original signal domain. We see that the operator T^s mapping $f(t)$ to the set of wavelet coefficients at scale s acts on $f(t)$ by multiplying its Fourier transform by a bandpass filter function $\widehat{\psi}^*(s\omega)$ which is scaled (in the Fourier domain) by s. Equation (9) will form the basis for us to later define the SGWT, where we will replace the Fourier transform by the graph Fourier transform.

We may express the individual wavelets by applying the operator T^s to a translated delta impulse function $\delta_a(t) = \delta(t-a)$. From Equation (6) it follows that

$$(T^s\delta_a)(t) = \frac{1}{s}\psi^*\left(\frac{t-a}{s}\right), \tag{10}$$

which for even and real-valued $\psi(t)$ simplifies to $(T^s\delta_a)(t) = \psi_{a,s}(t)$.

3 Spectral Graph Theory

In this section we introduce the tools from Spectral Graph theory needed to define the graph Fourier transform. This will provide the ability to define scaling of an operator, in the spectral domain, which is at the core of the SGWT construction. We first fix our notation for weighted graphs.

3.1 Notation for Weighted graphs

A weighted graph G consists of a finite vertex set V, a set of edges E (which is a subset of the set of all unordered pairs of vertices), and a non-negative valued weight function $w : E \to \mathbb{R}$ which gives a weight associated with each edge. We let $N = |V|$ denote the number of vertices. A finite weighted graph may also be unambiguously described by an $N \times N$ weighted adjacency matrix A, where $A_{i,j}$ equals zero if the the edge $(i, j) \notin E$, and $A_{i,j} = w((i, j))$ if the edge $(i, j) \in E$. We consider only symmetric (i.e. undirected) graphs.

The degree of each vertex is the sum of all the edge weights of edges incident to that vertex. In terms of the adjacency matrix, may write $d(m) = \sum_n A_{m,n}$ for the degree of vertex m. The diagonal degree matrix D is defined by

$$D_{i,j} = \begin{cases} d(i) & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{11}$$

Once a specific numbering of the vertices has been fixed, any function $f : V \to \mathbb{R}$ defined on the vertices can be naturally associated with a vector in $f \in \mathbb{R}^N$, where f_i is simply the value of the function on vertex i. We will also denote $f(i)$ for the value on vertex i.

In this work we use the non-normalized graph Laplacian operator $\mathscr{L}$, defined by $\mathscr{L} = D - A$. For any $f \in \mathbb{R}^N$, it is straightforward to show that

$$(\mathscr{L} f)(m) = \sum_{m \sim n} A_{m,n} \cdot (f(m) - f(n)) \tag{12}$$

where by $m \sim n$ we mean that the sum is taken over all vertices n which are connected to vertex m. This expression shows that the graph Laplacian operator applied to any function gives a weighted difference of the function values, summed over all edges incident to a given vertex.

Another form of the graph Laplacian that is commonly used elsewhere in the literature is the normalized form, given by

$$\mathscr{L}^{norm} = D^{-1/2} \mathscr{L} D^{-1/2} = I - D^{-1/2} A D^{-1/2} \tag{13}$$

The eigenvalues of this $\mathscr{L}^{norm}$ all lie in the interval $[0, 2]$. We note that the normalized and non-normalized Laplacian matrices are not similar matrices, and their

eigenvectors are different. The entire SGWT machinery could be defined using either form of the graph Laplacian, which would produce two different transforms. We will use the non-normalized form exclusively in the remainder of this work.

For the non-normalized Laplacian $\mathscr{L}$ constructed for a graph that corresponds to a regularly sampled grid, we note that $\mathscr{L}$ is proportional (with a difference in sign) to a standard finite difference approximation of the continuous Laplacian operator. For example, consider a regular two-dimensional grid with unit weights on the edges, where $v_{m,n}$ represents the vertex at index position (m,n) on the grid. Using these two-dimensional indices, one sees that for a function $f = f_{m,n}$ defined on the vertices, applying $\mathscr{L}$ yields (for (m,n) away from the boundary of the grid)

$$(\mathscr{L}f)_{m,n} = 4f_{m,n} - f_{m+1,n} - f_{m-1,n} - f_{m,n+1} - f_{m,n-1}. \tag{14}$$

Aside from a missing factor of h^2, where h is the mesh spacing, this is the standard 5-point stencil for computing $-\nabla^2 f$.

3.2 Graph Fourier Transform

The standard Fourier transform on the real line is

$$\widehat{f}(\omega) = \int e^{-it\omega} f(t) dt, \tag{15}$$

with inverse transform

$$f(t) = \frac{1}{2\pi} \int \widehat{f}(\omega) e^{i\omega t} d\omega. \tag{16}$$

The complex exponentials $e^{i\omega t}$ are eigenfunctions of the one-dimensional Laplacian $\frac{d}{dx^2}$. We may view the forward transform as computing the Fourier coefficient $\widehat{f}(\omega)$ as the inner product of the signal $f(t)$ with a Laplacian eigenfunction. Similarly, the inverse transform may be viewed as expanding the signal $f(t)$ as a weighted sum of Laplacian eigenfunctions.

The graph Fourier transform is obtained by analogy from the previous statements, by replacing the continous Laplacian $\frac{d}{dx^2}$ by the graph Laplacian $\mathscr{L}$. The matrix $\mathscr{L}$ is symmetric, and so has a set of orthonormal eigenvectors which span $\mathbb{R}^N$. We write these as χ_ℓ for $0 \leq \ell \leq N-1$. The corresponding eigenvalues λ_ℓ satisfy $\mathscr{L}\chi_\ell = \lambda_\ell \chi_\ell$. The symmetry of $\mathscr{L}$ implies that these eigenvalues are real, so we may organize them in ascending order. In addition, for the graph Laplacian $\mathscr{L}$, it holds that the eigenvalues are all non-negative, the smallest eigenvalue is 0, and the multiplicity of the 0 eigenvalue is equal to the number of connected components of the weighted graph G [23]. Assuming that G is connected, the eigenvalues λ_ℓ satisfy

$$0 = \lambda_0 < \lambda_1 \leq \lambda_2 ... \leq \lambda_{N-1} \tag{17}$$

We now define the graph Fourier transform. For any signal $f \in \mathbb{R}^N$, the graph Fourier transform $\widehat{f}$ is given by

$$\widehat{f}(\ell) = \langle \chi_\ell, f \rangle = \sum_{n=1}^{N} \chi_\ell^*(n) f(n). \tag{18}$$

The inverse graph Fourier transform expresses the original signal f as an expansion using the graph Fourier coefficients $\widehat{f}$ as

$$f(n) = \sum_{\ell=0}^{N-1} \widehat{f}(\ell) \chi_\ell(n) \tag{19}$$

The validity of the inverse Fourier transform is a straightforward consequence of the orthonormality of the eigenvectors χ_ℓ. It can similarly be shown that the graph Fourier coefficients satisfy the Parseval relation, i.e. for any two signals $f, g \in \mathbb{R}^N$ one has

$$\langle f, g \rangle = \left\langle \widehat{f}, \widehat{g} \right\rangle. \tag{20}$$

4 Spectral Graph Wavelets

Equipped with the graph Fourier transform, we are now prepared to describe the Spectral Graph Wavelet Transform. As alluded to earlier, specification of the SGWT requires fixing a non-negative real-valued kernel function g, which behaves as a band-pass filter and is analogous to the Fourier transform of the mother wavelet $\widehat{\psi}^*$ from Equation (9). We will require that $g(0) = 0$ and that $\lim_{\lambda \to \infty} g(\lambda) = 0$. Specific choices for the kernel g will be discussed later.

4.1 Wavelets

The wavelet operators producing the SGWT coefficients at each scale are obtained as rescaled kernel functions of the graph Laplacian operator. One may define a function of a self-adjoint operator by using the continuous functional calculus [38] based on the Spectral representation of the operator. For the finite dimensional graph Laplacian, this is afforded by the eigenvectors and eigenvalues of the Laplacian matrix $\mathscr{L}$. Specifically, we set the wavelet operator by $T_g = g(\mathscr{L})$. T_g is a mapping from $\mathbb{R}^N$ to $\mathbb{R}^N$, and $T_g f$ gives the wavelet coefficients for the signal f at unit scale ($s = 1$). This operator is defined by its action on the eigenvectors χ_ℓ, specifically as

$$T_g \chi_\ell = g(\lambda_\ell) \chi_\ell \tag{21}$$

This implies that for any graph signal f, the operator T_g acts on f by modulating each of its graph Fourier coefficients, according to

$$\widehat{T_g f}(\ell) = g(\lambda_\ell)\hat{f}(\ell) \tag{22}$$

Applying the inverse Fourier transform then shows

$$(T_g f)(m) = \sum_{\ell=0}^{N-1} g(\lambda_\ell)\hat{f}(\ell)\chi_\ell(m) \tag{23}$$

This relation should be compared with Equation (9) describing the mapping from signal to Wavelet coefficients for the Continuous Wavelet Transform.

We next define T_g^s, the Wavelet operator at scale s, as $T_g^s = g(s\mathscr{L})$. The crucial point enabling this definition of scaling is that while the original spatial domain (set of vertices) is discrete, the domain of the kernel function $g(\lambda)$ is continuous, which enables proper definition of T_g^s, for any $s > 0$.

The individual wavelets are obtained by localizing these operators by applying them to δ_n, where $\delta_n \in \mathbb{R}^N$ is the signal with a 1 on vertex n and zeros elsewhere. This reads as

$$\psi_{s,n} = T_g^s \delta_n, \tag{24}$$

so that $\psi_{s,n}$ is the Spectral Graph Wavelet at scale s, centered on vertex n. We now observe that

$$\widehat{\delta_n}(\ell) = \sum_{m=1}^{N} \chi_\ell^*(m)\delta_n(m) = \chi_\ell^*(n). \tag{25}$$

Using this with Equation (23) then implies that

$$\psi_{s,n}(m) = \sum_{\ell=0}^{N-1} g(s\lambda_\ell)\chi_\ell^*(n)\chi_\ell(m) \tag{26}$$

The wavelet coefficients $W_f(s,n)$ may then be considered as the inner products of f with the wavelet $\psi_{s,n}$, i.e. via

$$W_f(s,n) = \langle \psi_{s,n}, f \rangle . \tag{27}$$

Equivalently, we have $W_f(s,n)$ as the value of $(T_g^s f)_n$. Using Equation (23) this may be expanded as

$$W_f(s,n) = \left(T_g^s f\right)(n) = \sum_{\ell=0}^{N-1} g(s\lambda_\ell)\hat{f}(\ell)\chi_\ell(n) \tag{28}$$

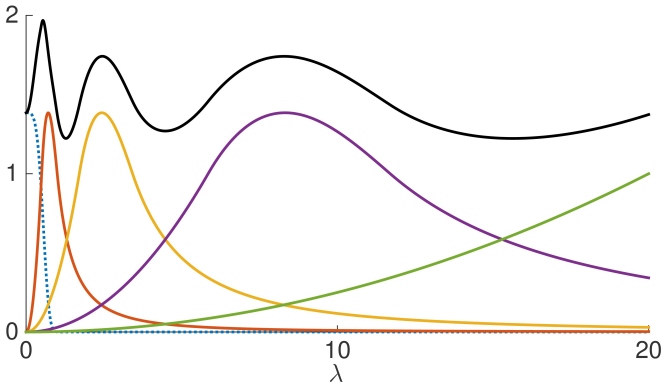

Fig. 1 Scaling function $h(\lambda)$ (dotted blue curve), wavelet generating kernels $g(t_j\lambda)$ (green,magenta,yellow,orange curves), and sum of squares G (black curve), for $J = 4$ scales, $\lambda_{max} = 20$. Scale values are $t_1 = 2.0$, $t_2 = 0.5848$, $t_3 = 0.171$, $t_4 = 0.05$. Details for the functional form of h and g are in Section 7.

4.2 Scaling Functions

As the wavelet kernel g satisfies $g(0) = 0$, the wavelets $\psi_{s,n}$ are all orthogonal to the eigenvector χ_0, and are close to orthogonal to χ_ℓ for eigenvectors where λ_ℓ is close to zero. In order to stably represent the lower frequency content of signals, it is helpful to introduce a set of spectral graph scaling functions. These are defined in analogy with scaling functions for the classical wavelet transform, which are needed to represent low frequency content of signals when the scale parameter is not allowed to become arbitrarily large. We define the spectral graph scaling functions similarly to the wavelets, using a non-negative valued scaling function kernel $h(\lambda)$ which may be viewed as a low-pass filter. The scaling function kernel h satisfies $h(0) > 0$ and $\lim_{\lambda\to\infty} h(\lambda) = 0$, from which the scaling function operator $T_h = h(\mathscr{L})$ is defined. The scaling functions centered at vertex n are given by $\phi_n = T_h\delta_n$, and the scaling function coefficients are given by $S_f(n) = \langle \phi_n, f \rangle$.

In figure 1 we show the graphs of representative scaled wavelet kernels and the scaling function kernel, for a chosen set of discrete scale values s_j. Details of the choices for h and g are deferred until later. We will show later in Section 5.2 that stable recovery of f from its wavelet and scaling function coefficients is possible if the function $G(\lambda) = h(\lambda)^2 + \sum_j g(s_j\lambda)^2$ is nonzero for all λ in the spectrum of $\mathscr{L}$. Clearly, as each of the scaled wavelet kernels $g(s_j\lambda)$ approach zero as $\lambda \to 0$, this condition can only hold if the scaling function $h(\lambda)$ satisfies $h(0) > 0$.

The scaling functions defined here are used to represent low frequency content of the signal f. We note that the kernels h and g used here do not satisfy the two-scale relation as in classical orthogonal wavelet design [3]. We thus have much freedom in choosing the form of h, provided that the resulting G does not become close to zero over the spectrum of $\mathscr{L}$.

5 Properties of the SGWT

We next describe several properties of the spectral graph wavelet transform, including the inverse of the continuous transform, the localization properties in the small-scale limit, and the frame bounds for the scale-discretized transform.

5.1 Inverse for Continuous SGWT

For any type of signal transform to be useful for signal processing (rather than only signal analysis), one must be able to invert the transform, i.e. to reconstruct a signal corresponding to a given set of transform coefficients. The continuous SGWT (i.e. where the scale parameter is not discretized) admits an inverse formula that has a very similar form to the inverse expression for the continuous wavelet transform in Equation (4).

Each wavelet coefficient $W_f(s,n)$ may be viewed as measuring the "amount" of the wavelet $\psi_{s,n}$ present in the original signal f. The continuous SGWT inverse uses these measurements to reconstruct the original signal, with a weighting ds/s. As mentioned previously, however, as all of the wavelets are orthogonal to the eigenvector χ_0, the subspace spanned by χ_0 must be handled separately.

Lemma 1. *Let $f \in \mathbb{R}^N$ be a signal, and let $f^\#$ be the projection of f onto the orthogonal complement of the span of χ_0, i.e. $f^\# = f - \langle \chi_0, f \rangle \chi_0$. Let g be a kernel function satisfying $g(0) = 0$, and the admissibility condition*

$$\int_0^\infty \frac{g^2(x)}{x} dx = C_g < \infty. \tag{29}$$

Then the continuous reconstruction formula holds :

$$\frac{1}{C_g} \sum_{n=1}^{N} \int_0^\infty W_f(s,n) \psi_{s,n}(m) \frac{ds}{s} = f^\#(m) \tag{30}$$

The complete reconstruction of f is then given by $f = f^\# + \hat{f}(0)\chi_0$.

Proof. We first expand the left side of the above, using Equations (26) and (28) to write $\psi_{t,n}$ and $W_f(t,n)$ in terms of the Laplacian eigenvectors χ_ℓ. This gives

$$\frac{1}{C_g} \int_0^\infty \frac{1}{s} \sum_n \left(\sum_\ell g(s\lambda_\ell) \chi_\ell(n) \hat{f}(\ell) \sum_{\ell'} g(s\lambda_{\ell'}) \chi^*_{\ell'}(n) \chi_{\ell'}(m) \right) ds \tag{31}$$

$$= \frac{1}{C_g} \int_0^\infty \frac{1}{s} \left(\sum_{\ell,\ell'} g(s\lambda_{\ell'}) g(s\lambda_\ell) \hat{f}(\ell) \chi_{\ell'}(m) \sum_n \chi^*_{\ell'}(n) \chi_\ell(n) \right) ds \tag{32}$$

We have that $\sum_n \chi^*_{\ell'}(n)\chi_\ell(n) = \delta_{\ell,\ell'}$, applying this and summing over ℓ' gives

$$= \frac{1}{C_g} \sum_{\ell} \left(\int_0^{\infty} \frac{g^2(s\lambda_\ell)}{s} ds \right) \hat{f}(\ell) \chi_\ell(m) \tag{33}$$

Using the substitution $u = s\lambda_\ell$, provided that $\lambda_\ell \neq 0$, reduces the integral appearing above to $\int \frac{g^2(u)}{u} du$, which is finite and equals C_g by the admissibility condition for g. If $\lambda_\ell = 0$, which holds only for $\ell = 0$, then the integral is 0 as $g(0) = 0$. This implies that Equation (33) is precisely the inverse Fourier transform evaluated at vertex m, with the $\ell = 0$ omitted from the sum. As the omitted $\ell = 0$ term is precisely $\hat{f}(0)\chi_0 = \langle \chi_0, f \rangle \chi_0$, the lemma is proved.

This expression for the inverse of the continuous transform is of theoretical interest, however any practical implementation of the SGWT must use a finite number of wavelet scales. We shall discuss reconstruction from the scale discretized SGWT later in this chapter.

5.2 Frame bounds for SGWT

As alluded to previously, practical computation of the SGWT must involve discretizing the scale parameter s to a finite set of values. Fixing our notation, we let J be the number of scales chosen and let $\{s_1, s_2, \ldots, s_J\}$ denote the specific scale values. The SGWT at each scale produces a set (often termed a "subband") of N coefficients $W_{s_j,n}$ for $1 \leq n \leq N$. Together with the N scaling function coefficients, the full transform with J scales may be considered as a mapping from $\mathbb{R}^N$ to $\mathbb{R}^{N(J+1)}$, producing $N(J+1)$ coefficients.

Some insight into how stable the coefficients for the entire set of NJ wavelets and N scaling functions are for representing signals may be gained by considering the frame formed by the entire set of wavelets and scaling functions. Briefly, for a Hilbert space $\mathscr{H}$, a set of vectors $\Gamma_n \in \mathscr{H}$ is said to be a frame with frame bounds A and B if for all $f \in \mathscr{H}$ it is true that

$$A\,||f||^2 \leq \sum_n |c_n|^2 \leq B\,||f||^2 . \tag{34}$$

where the coefficients c_n are given by $c_n = \langle \Gamma_n, f \rangle$. These constants A and B describe the numerical stability of recovering the original signal f from the coefficients c_n. In particular, if $A = B$, then the set $\{\Gamma_n\}$ is called a tight frame, and the signal may be recovered simply by

$$f = \frac{1}{A} \sum_n c_n \Gamma_n \tag{35}$$

In general A and B will not be equal, however the guiding principal that the frame is easier to invert if B/A is close to 1 still holds. In fact, as discussed in section 6.3, these frame bounds provide a precise estimate for the speed of convergence of the conjugate-gradients algorithm for inverting the discrete SGWT. For further details of the fundamentals of the theory of frames, see [39] or [40].

For the scale-discretized SGWT, the frame bounds are given by the following.

Theorem 1. *Fix a choice of a set of scales $\{s_1,...,s_J\}$. Set $G(\lambda)=h^2(\lambda)+\sum_j g(s_j\lambda)^2$, where h and g are the scaling function and wavelet kernels. Then the set $\Gamma=\{\phi_n\}_{n=1}^N\cup\{\psi_{s_j,n}\}_{j=1\ n=1}^{J\ \ N}$ is frame with frame bounds A, B given by*

$$\begin{aligned} A &= \min_{\lambda\in[0,\lambda_{N-1}]} G(\lambda) \\ B &= \max_{\lambda\in[0,\lambda_{N-1}]} G(\lambda). \end{aligned} \tag{36}$$

Proof. Expression (28), shows that, for a fixed signal f, we may write

$$\begin{aligned} \sum_n |W_f(s,n)|^2 &= \sum_n \sum_\ell g(s\lambda_\ell)\chi_\ell(n)\hat{f}(\ell)\sum_{\ell'}\left(g(s\lambda_{\ell'})\chi_{\ell'}(n)\hat{f}(\ell')\right)^* \\ &= \sum_\ell |g(s\lambda_\ell)|^2|\hat{f}(\ell)|^2, \end{aligned} \tag{37}$$

where we have used the orthonormality of the χ_n. For the scaling function coefficients we have, similarly,

$$\sum_n |S_f(n)|^2 = \sum_\ell |h(\lambda_\ell)|^2|\hat{f}(\ell)|^2 \tag{38}$$

Fix an ordering of the elements of Γ, so that $\Gamma=\cup_{k=1}^{N(J+1)}\gamma_k$. Note that $\langle\gamma_k,f\rangle$ may be either a scaling function coefficient or wavelet coefficient, depending on k. Equations (37) and (38) show that

$$\sum_{k=1}^{N(J+1)} |\langle\gamma_k,f\rangle|^2 = \sum_\ell \left(|h(\lambda_\ell)|^2+\sum_{j=1}^J |g(s_j\lambda_\ell)|^2\right)|\hat{f}(\ell)|^2 = \sum_\ell G(\lambda_\ell)|\hat{f}(\ell)|^2 \tag{39}$$

Then by the definition of A and B, we have

$$A\sum_{\ell=0}^{N-1}|\hat{f}(\ell)|^2 \le \sum_{k=1}^{N(J+1)} |\langle\gamma_k,f\rangle|^2 \le B\sum_{\ell=0}^{N-1}|\hat{f}(\ell)|^2 \tag{40}$$

The Parseval relation $||f||^2=\sum_\ell|\hat{f}(\ell)|^2$ then implies that A and B are frame bounds for the frame Γ.

5.3 Limit of small scales

Much of the effectiveness of classical wavelets for signal processing follows as the wavelets may be designed to be localized in both the spatial domain and the frequency domain. The spectral graph wavelets may be designed to be localized in the

frequency domain, provided that g is chosen as a band-pass filter. However, we have not yet demonstrated localization of the spectral graph wavelets in the spatial (i.e. vertex) domain.

The spatial localization properties of classical wavelets derived from a single wavelet via dilation and translation are straightforward to infer from the mother wavelet $\psi(t)$ itself. If $\psi(t)$ is well localized on the interval $[-d,d]$, then the derived wavelet $\psi_{s,a}(t)$ will be well localized on $[a-ds,a+ds]$. In the limit of small scales as $s \to 0$, this implies that $\psi_{s,a}(t) \to 0$ for all $t \neq a$, as long as the original mother $\psi(t)$ wavelet decays to zero as $t \to \infty$.

As scaling for the spectral graph wavelets is defined in the graph Fourier domain, localization in the limit as $s \to 0$ is not as straightforward to infer. We will demonstrate that normalized spectral graph wavelets $\frac{\psi_{s,n}}{||\psi_{s,n}||}$ will approach zero for vertices far enough away from the central vertex n, as $s \to 0$. Our result is based on the fact that powers of $\mathscr{L}$ are localized, and that T_g^s may be approximated as proportional to a power of $\mathscr{L}$ in the limit of small scales.

As noted previously, the operator T_g^s depends only on the values of $g(s\lambda)$ for λ in the spectrum of $\mathscr{L}$, in particular the values of $g(s\lambda)$ for $\lambda > \lambda_{N-1}$ have no effect on T_g^s. As the graph of $g(s\lambda)$ is obtained from the graph of $g(\lambda)$ by zooming in by a factor $1/s$, the operator T_g^s is determined by the values of $g(\lambda)$ over the small interval $[0, \lambda_{N-1}s]$. Our approach will be to approximate $g(\lambda)$ in a neighborhood of 0 by its Taylor polynomial, which will let us transfer the study of localization of T_g^s to studying localization of the first nonzero power of $\mathscr{L}$ appearing in the Taylor series.

The Taylor polynomial for g at the origin will provide an approximation of $g(s\lambda)$ that we will use for small s. In order to study the resulting approximate wavelets, we first establish a bound on how perturbations of the kernel function g affect the resulting wavelets. If two kernel functions g and $\tilde{g}$ are close to each other, then their resulting wavelets should also be close to each other.

Lemma 2. *Let g and $\tilde{g}$ be two kernel functions, and $\psi_{s,n} = T_g^s \delta_n$ and $\tilde{\psi}_{s,n} = T_{\tilde{g}}^s \delta_n$ be their corresponding wavelets at scale s. Suppose that there is a bound $M(s)$ so that $|g(s\lambda) - \tilde{g}(s\lambda)| \leq M(s)$ for all $\lambda \in [0, \lambda_{N-1}]$. It then follows that for each value of s and for each vertex m, $|\psi_{s,n}(m) - \tilde{\psi}_{s,n}(m)| \leq M(s)$, and that $||\psi_{s,n} - \tilde{\psi}_{s,n}||_2 \leq \sqrt{N}M(s)$.*

Proof. As by definition $\psi_{s,n}(m) = \langle \delta_m, g(s\mathscr{L})\delta_n \rangle$, we may write

$$|\psi_{s,n}(m) - \tilde{\psi}_{s,n}(m)| = |\langle \delta_m, (g(s\mathscr{L}) - \tilde{g}(s\mathscr{L}))\,\delta_n \rangle| \tag{41}$$

Using the Parseval relation for the graph Fourier transform (20) shows this may be written as

$$|\psi_{s,n}(m) - \tilde{\psi}_{s,n}(m)| = \left| \sum_\ell \chi_\ell(m)(g(s\lambda_\ell) - \tilde{g}(s\lambda_\ell))\chi_\ell^*(n) \right|$$
$$\leq M(s) \sum_\ell |\chi_\ell(m)\chi_\ell(n)^*| \tag{42}$$

Using the Cauchy-Schwartz inequality shows the above sum over ℓ is bounded by 1, as

$$\sum_\ell |\chi_\ell(m)\chi_\ell^*(n)| \leq \left(\sum_\ell |\chi_\ell(m)|^2 \right)^{1/2} \left(\sum_\ell |\chi_\ell^*(n)|^2 \right)^{1/2}, \tag{43}$$

and $\sum_\ell |\chi_\ell(m)|^2 = 1$ for all m, as the χ_ℓ are a complete orthonormal basis. Applying this to (42) proves $|\psi_{s,n}(m) - \tilde{\psi}_{s,n}(m)| \leq M(s)$. We may then write

$$||\psi_{s,n} - \tilde{\psi}_{s,n}||_2^2 = \sum_m (\psi_{s,n}(m) - \tilde{\psi}_{s,n}(m))^2 \leq \sum_m M(s)^2 = NM(s)^2 \tag{44}$$

which proves the statement $||\psi_{s,n} - \tilde{\psi}_{s,n}||_2 \leq \sqrt{N}M(s)$.

Our localization results will be stated using a notion of distance between vertices. We employ the shortest-path distance, which defines the distance between two vertices as the number of edges in the shortest path connecting them, i.e.

$$d_G(m,n) = \operatorname*{argmin}_s \{k_1, k_2, ..., k_s\} \tag{45}$$
$$\text{s.t. } m = k_1,\ n = k_s, \text{ and } A_{k_r,k_{r+1}} > 0 \text{ for } 1 \leq r < s. \tag{46}$$

This distance measure treats the graph G as a binary graph, i.e. the particular values of the nonzero edge weights are not used.

For integer powers of $\mathscr{L}$, we have the following localization result. Note that this holds for both the normalized and non-normalized forms of the Laplacian.

Lemma 3. *Let G be a weighted graph, and $\mathscr{L}$ the graph Laplacian of G. Fix an integer $s > 0$, and pick vertices m and n. Then $(\mathscr{L}^s)_{m,n} = 0$ whenever $d_G(m,n) > s$.*

Proof. By the construction of $\mathscr{L}$, we have that $\mathscr{L}_{r,s} = 0$ for any vertices $r \neq s$ that are not connected (i.e. where $A_{r,s} = 0$). Writing out repeated matrix multiplication, we see

$$(\mathscr{L}^s)_{m,n} = \sum_{k_1=1}^{N} \sum_{k_2=1}^{N} \cdots \sum_{k_{s-1}=1}^{N} \mathscr{L}_{m,k_1} \mathscr{L}_{k_1,k_2} ... \mathscr{L}_{k_{s-1},n} \tag{47}$$

Suppose for sake of contradiction that $(L^s)_{m,n} \neq 0$. This implies that at least one of the terms in the above sum is nonzero, which demonstrates the existence of a sequence of vertices $k_1, k_2, ..., k_{s-1}$ with $\mathscr{L}_{m,k_1} \neq 0$, $L_{k_1,k_2} \neq 0$, ... , $\mathscr{L}_{k_{s-1}} \neq 0$. However, this is precisely a path of length s from vertex m to vertex n, with possible repeats of vertices. Removing these possible repeated vertices gives a path of length $k \leq s$ from vertex m to n, which implies $d_G(m,n) \leq s$, which is a contradiction.

This result implies that any kernel function that can be approximated by an integer power of $\mathscr{L}$ will produce localized wavelets. Every valid kernel g satisfies

$g(0) = 0$, if g is smooth in a neighborhood of 0 then we may approximate $g(s\lambda)$ using the first nonzero term of the Taylor series for g, which will allow us to use Lemma 3. We first clarify this truncated Taylor approximation for kernels $g(x)$ that have a zero with integer multiplicity at $x = 0$.

Lemma 4. *Suppose g satisfies $g(0) = 0$, $g^{(r)}(0) = 0$ for all $r < K$, and $g^{(K)}(0) = C \neq 0$. Let there be some $s' > 0$ so that g is $K+1$ times continuously differentiable on $[0, s'\lambda_{N-1}]$ and that $|g^{(K+1)}(\lambda)| \leq B$ for all $\lambda \in [0, s'\lambda_{N-1}]$. Define the monomial approximation kernel $\tilde{g}$ by $\tilde{g}(x) = (C/K!)x^K$, and set*

$$M(s) = \sup_{\lambda \in [0, \lambda_{N-1}]} |g(s\lambda) - \tilde{g}(s\lambda)|. \tag{48}$$

Then for all $s < s'$, the error $M(s)$ is bounded by

$$M(s) \leq s^{K+1} \frac{\lambda_{N-1}^{K+1}}{(K+1)!} B \tag{49}$$

Proof. Using the assumed information about the derivatives of g at $x = 0$, Taylor's formula with remainder shows that for any $x < s'\lambda_{N-1}$,

$$\begin{aligned} g(x) &= C\frac{x^K}{K!} + g^{(K+1)}(x^*)\frac{x^{K+1}}{(K+1)!} \\ &= \tilde{g}(x) + g^{(K+1)}(x^*)\frac{x^{K+1}}{(K+1)!} \end{aligned} \tag{50}$$

for some $x^* \in [0, x]$. By assumption we have $g^{(K+1)}(x^*) < B$. Now fix $s < s'$, and set $x = s\lambda$. We then have for all $0 \leq \lambda \leq \lambda_{N-1}$ that

$$|g(s\lambda) - \tilde{g}(s\lambda)| \leq B\frac{s^{K+1}\lambda^{K+1}}{(K+1)!} \leq B\frac{s^{K+1}\lambda_{N-1}^{K+1}}{(K+1)!}, \tag{51}$$

which implies $M(s) \leq s^{K+1}\frac{\lambda_{N-1}^{K+1}}{(K+1)!}B$.

We are now equipped to state our localization result for the spectral graph wavelets in the limit of small scales. We note that simply due to the definition of the SGWT, if $g(0) = 0$ and g is continuous it follows that $\lim_{s\to 0} \psi_{s,n}(m) = 0$ for all m, n. This explains why the statement of our result includes normalization by $||\psi_{s,n}||$.

Theorem 2. *Let g be a kernel function satisfying $g^{(r)}(0) = 0$ for $0 \leq r < K$, and $g^{(K)}(0) \neq 0$, and let s' and B be such that $|g^{(K+1)}(\lambda)| \leq B$ for all $0 \leq \lambda \leq s'\lambda_{N-1}$. Let m and n be vertices separated by distance greater than K, i.e. with $d_G(m,n) > K$. Then there are constants D and s'' so that*

$$\frac{\psi_{s,n}(m)}{||\psi_{s,n}||} \leq Ds \tag{52}$$

holds for all sufficiently small scales $s < \min(s', s'')$.

Proof. Define $\tilde{g}(\lambda) = \frac{g^{(K)}(0)}{K!}\lambda^K$ *and set* $\tilde{\psi}_{s,n} = T_{\tilde{g}}^s \delta_n$. *We expand* $\tilde{\psi}_{s,n} = \frac{g^{(K)}(0)}{K!} s^K \mathscr{L}^K \delta_n$, *so that*

$$\begin{aligned}\tilde{\psi}_{s,n}(m) &= \frac{g^{(K)}(0)}{K!} s^K \left\langle \delta_m, \mathscr{L}^K \delta_n \right\rangle \\ &= \frac{g^{(K)}(0)}{K!} s^K \left(\mathscr{L}^K\right)_{m,n} \\ &= 0 \end{aligned} \tag{53}$$

as $(\mathscr{L}^K)_{m,n} = 0$ *by Lemma 3. Combining Lemmas 2 and 4 shows*

$$|\psi_{s,n}(m) - \tilde{\psi}_{s,n}(m)| = |\psi_{s,n}(m)| \leq s^{K+1} E \tag{54}$$

with $E = \frac{\lambda_{N-1}^{K+1}}{(K+1)!} B$.

We next need to bound $||\psi_{s,n}||$ *away from 0. The triangle inequality applied to* $\tilde{\psi}_{s,n} = \psi_{s,n} + (\tilde{\psi}_{s,n} - \psi_{s,n})$ *directly gives* $||\tilde{\psi}_{s,n}|| \leq ||\psi_{s,n}|| + ||\tilde{\psi}_{s,n} - \psi_{s,n}||$, *so*

$$||\tilde{\psi}_{s,n}|| - ||\psi_{s,n} - \tilde{\psi}_{s,n}|| \leq ||\psi_{s,n}|| \tag{55}$$

Lemma 4 shows that $||\psi_{s,n} - \tilde{\psi}_{s,n}|| \leq \sqrt{N} s^{K+1} \frac{\lambda_{N-1}^{K+1}}{(K+1)!} B$, *while we may simply calculate* $||\tilde{\psi}_{s,n}|| = s^K \frac{g^{(K)}(0)}{K!} \left|\left|\mathscr{L}^K \delta_n\right|\right|$. *These show that*

$$s^K \left(\frac{g^{(K)}(0)}{K!} \left|\left|\mathscr{L}^K \delta_n\right|\right| - s\sqrt{N} \frac{\lambda_{N-1}^{K+1}}{(K+1)!} B \right) \leq ||\tilde{\psi}_{s,n}|| - ||\psi_{s,n} - \tilde{\psi}_{s,n}|| \tag{56}$$

Equations (56) and (54) together yield

$$\frac{\psi_{s,n}(m)}{||\psi_{s,n}||} \leq \frac{sE}{q - sp} \tag{57}$$

where we define $q = \frac{g^{(K)}(0)}{K!} \left|\left|\mathscr{L}^K \delta_n\right|\right|$ *and* $p = \sqrt{N} \frac{\lambda_{N-1}^{K+1}}{(K+1)!} B$. *Straightforward computation demonstrates that* $\frac{sE}{q-sp} \leq \frac{2E}{q} s$ *whenever* $s \leq \frac{q}{2p}$. *This implies the stated theorem once we define* $D = \frac{2EK!}{g^{(K)}(0)\left|\left|\mathscr{L}^K \delta_n\right|\right|}$ *and* $s'' = \frac{g^{(K)}(0)\left|\left|\mathscr{L}^K \delta_n\right|\right|(K+1)}{2\sqrt{N}\lambda_{N-1}^{K+1} B}$.

The localization result stated in Theorem 2 uses the shortest-path distance, and thus as stated is really only meaningful for graphs where the shortest-path distance (which treats all non-zero edges the same, even if the edge weights are close to zero) is a useful measure of distance. This will be the case if a significant number of edge weights are exactly zero. We note that many large graphs arising in practice are sparse (i.e. the number of nonzero edges is small relative to the total number

of possible edges), for such sparse weighted graphs the shortest-path distance does provide a meaningful notion of distance.

6 Polynomial Approximation

The SGWT is defined using the eigenvectors χ_ℓ and eigenvalues λ_ℓ of the $N \times N$ matrix $\mathscr{L}$. Directly computing the transform according to Equation 28 requires diagonalizing $\mathscr{L}$, i.e. computing the full set of eigenvectors and eigenvalues. This is computationally intensive, requiring $O(N^3)$ operations for the commonly used QR algorithm [41]. This computational complexity renders the direct computation of the entire set of eigenvectors impractical for graphs with more than a several thousand vertices. However, signal processing problems routinely involve data with hundreds of thousands or millions of dimensions. The SGWT cannot be a practical tool for such larger problems if its computation relies on fully diagonalizing $\mathscr{L}$.

In this section we describe a fast algorithm for the SGWT that avoids the need to diagonalize the graph Laplacian. This is a achieved by directly approximating the scaled wavelet kernels $g(s_j\lambda)$ by polynomials. A polynomial function of $\mathscr{L}$ may be applied to a signal f in a manner which uses only matrix-vector multiplication. In general, multiplying a vector by $\mathscr{L}$ requires a number of operations equal to the number of nonzero edges in the graph. For sparse graphs, where the number of nonzero edges is small, this yields an efficient procedure.

6.1 Chebyshev polynomial approximation

As mentioned previously, the wavelet operator T_g^s depends on the values of $g(s\lambda)$ only for λ within the spectrum of $\mathscr{L}$. This implies that the polynomial approximations we seek need only be valid on an interval containing the spectrum of $\mathscr{L}$.

Lemma 5. *Let λ_{max} be an upper bound on the spectrum of $\mathscr{L}$, so that $\lambda_{max} \geq \lambda_{N-1}$. Let $p(\lambda)$ be a polynomial such that, for fixed scale s, $\max_{\lambda \in [0,\lambda_{max}]} |g(s\lambda) - p(\lambda)| = B$, and define the approximate wavelet coefficients by $\tilde{W}_f(s,n) = (p(\mathscr{L})f)_n$. Then the error in the approximate wavelet coefficients satisfies*

$$|W_f(s,n) - \tilde{W}_f(s,n)| \leq B\,||f|| \tag{58}$$

Proof. Equation (28) shows

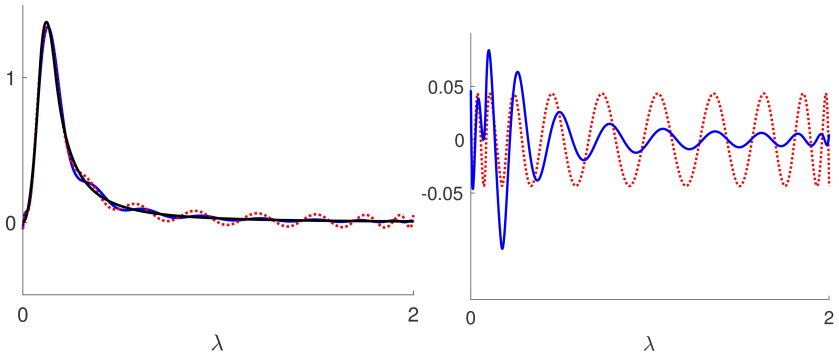

Fig. 2 (a) Wavelet kernel $g(\lambda)$ (black), truncated Chebyshev expansion (blue) and minimax polynomial approximation (red, dashed) for degree $m = 20$, shown for $[0, \lambda_{max}] = [0, 2]$. Approximation errors shown in (b), the truncated Chebyshev expansion has maximum error 0.1023, the minimax polynomial has maximum error 0.0434.

$$\begin{aligned} |W_f(s,n) - \tilde{W}_f(s,n)| &= \left| \sum_\ell g(s\lambda_\ell)\hat{f}(\ell)\chi_\ell(n) - \sum_\ell p(\lambda_\ell)\hat{f}(\ell)\chi_\ell(n) \right| \\ &\leq \sum_\ell |g(s\lambda_\ell) - p(\lambda_\ell)||\hat{f}(\ell)\chi_\ell(n)| \\ &\leq B \sum_\ell |\hat{f}(\ell)\chi_\ell(n)| \end{aligned} \tag{59}$$

The Cauchy-Schwartz inequality applied on the last sum above shows

$$\sum_\ell |\hat{f}(\ell)\chi_\ell(n)| \leq \sum_\ell (\hat{f}(\ell))^2 \sum_\ell (\chi_\ell(n))^2 = ||f||\,, \tag{60}$$

using the Parseval equality and the orthonormality of the χ_ℓ's. Together (59) and (60) imply Equation (58).

An upper bound λ_{max} such as used in the above lemma can be found by calculating the largest eigenvalue of $\mathscr{L}$. It is important to note that good algorithms exist for finding the largest eigenvalue of a symmetric matrix that access the matrix only via matrix-vector multiplication. Examples include Arnoldi iteration and the Jacobi-Davidson method [41, 42]. These algorithms are able to compute accurate estimates of λ_{N-1} with much lower computational cost than needed to find the entire spectrum of $\mathscr{L}$. Additionally, as only a rough estimate of λ_{N-1} is needed to form an upper bound (the rough estimate may simply be increased to ensure a valid upper bound), the Arnoldi iteration need not be run until close convergence is achieved for computing λ_{N-1}.

In this work we will use polynomial approximations computed from the truncated Chebyshev polynomial expansion of the scaled wavelet kernels $g(s\lambda)$, over the interval $[0, \lambda_{max}]$. Lemma 5 suggests that polynomial approximations $p(\lambda)$ should be

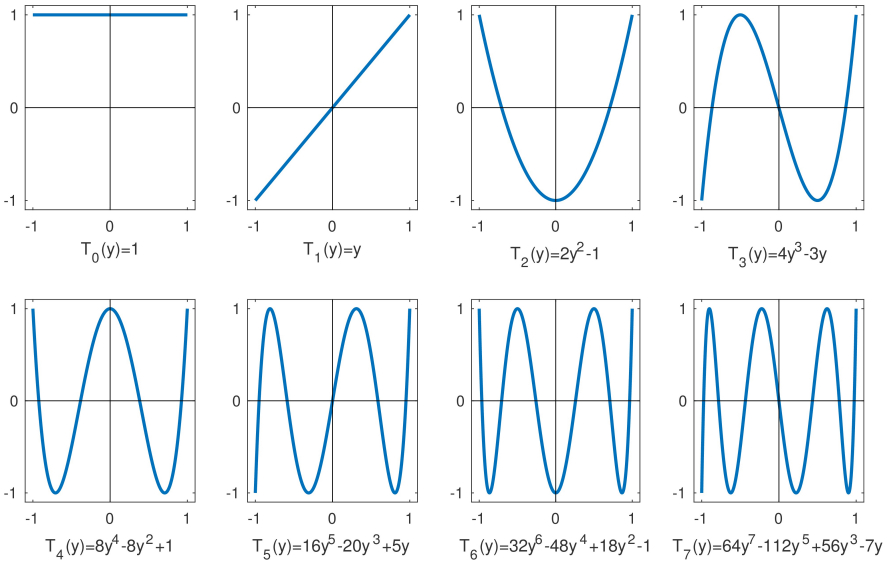

Fig. 3 Graphs of the Chebyshev polynomials $T_k(y)$ for $0 \leq k \leq 7$, plotted on the interval [-1,1].

chosen to minimize the supremum norm $B = \max_{\lambda \in [0,\lambda_{max}]} |p(\lambda) - g(s\lambda)|$. Truncated Chebyshev polynomial expansions give polynomials that in many cases are a close approximation of the so-called minimax polynomials that exactly minimize the supremum norm [43]. The minimax polynomial $p(x)$ of degree M for approximating $g(sx)$ has the property that the error $|p(x) - g(sx)|$ reaches the same maximum value at $M+2$ points across the domain. This is illustrative of the fact that the minimax polynomials distribute the approximation error evenly over the entire interval. In contrast, for the wavelet kernels used in this work we have observed that the truncated Chebyshev polynomials have a maximum error only slightly greater than that of the minimax polynomials, and typically have significantly lower approximation error in regions where $g(s\lambda)$ is smooth. We have also observed that for graphs that are small enough where the SGWT may be computed exactly using Equation (28), the polynomial approximation using truncated Chebyshev expansions produces a slightly lower approximation error than that based on the minimax polynomials. We illustrate a scaled wavelet kernel and both the truncated Chebyshev and minimax polynomial approximations (computed using the Remez exchange algorithm [44]) in Figure 2.

Another reason we adopt the truncated Chebyshev approximation is that we can use the recurrence properties of the Chebyshev polynomials to conveniently evaluate Chebyshev polynomials of $\mathscr{L}$ applied to an input signal f via repeated matrix-vector multiplication. Chebyshev polynomial approximation is a classical topic, a good overview is [45]. For completeness we briefly describe a few key properties of the Chebyshev polynomials here.

The (unscaled) Chebyshev polynomials are a set of polynomials convenient for representing functions on the interval $[-1,1]$. On this interval they satisfy $T_k(y) = \cos(k \arccos(y))$, showing that they oscillate between $[-1,1]$, and that the k$^{\text{th}}$ order polynomial $T_k(y)$ has zeros at the points $y = \cos(\frac{\pi}{k}(n+\frac{1}{2}))$ for $n = 0,1,2,...k-1$. The Chebyshev polynomials satisfy the two-term recurrence relation

$$T_k(y) = 2yT_{k-1}(y) - T_{k-2}(y), \tag{61}$$

which together with the starting expressions $T_0(y) = 1$ and $T_1(y) = y$ can be used to generate the entire sequence. The graphs of the first 8 Chebyshev polynomials are shown in Figure 3.

Many of the approximation properties of the Chebyshev polynomials follow from them being an orthogonal set, with respect to the inner product defined with the measure $\frac{dy}{\sqrt{1-y^2}}$. Specifically, one has

$$\int_{-1}^{1} \frac{T_l(y)T_m(y)}{\sqrt{1-y^2}} dy = \begin{cases} \delta_{l,m}\pi/2 & \text{if } m,l > 0 \\ \pi & \text{if } m = l = 0 \end{cases} \tag{62}$$

Any function h which is square-integrable on $[-1,1]$ with respect to the measure $dy/\sqrt{1-y^2}$ has a convergent Chebyshev series, given by

$$h(y) = \frac{1}{2}c_0 + \sum_{k=1}^{\infty} c_k T_k(y) \tag{63}$$

where the Chebyshev coefficients c_k are given by

$$c_k = \frac{2}{\pi}\int_{-1}^{1} \frac{T_k(y)h(y)}{\sqrt{1-y^2}} dy = \frac{2}{\pi}\int_0^{\pi} \cos(k\theta)h(\cos(\theta))d\theta. \tag{64}$$

We now detail the polynomial approximation scheme used for the fast computation of the SGWT. We first rescale the argument of the Chebyshev polynomials by the change of variables $x = \lambda_{max}(y+1)/2$, which transforms the interval $[-1,1]$ onto $[0,\lambda_{max}]$. We write $\overline{T}_k(x)$ for these shifted Chebyshev polynomials, which satisfy

$$\overline{T}_k(x) = T_k(\frac{2x-\lambda_{max}}{\lambda_{max}}) \tag{65}$$

We let M denote the degree of the polynomial approximations for each of the scaled wavelet kernels, and assume we have fixed some set of scales s_j. Larger values of M will yield more accurate approximations, at the expense of higher computational cost. For each scale s_j, the truncated Chebyshev polynomial $p_j(x)$ which approximates $g(s_j x)$ has the expression

$$p_j(x) = \frac{1}{2}c_{j,0} + \sum_{k=1}^{M} c_{j,k}\overline{T}_k(x), \tag{66}$$

where the Chebyshev coefficients are given by

$$c_{j,k} = \frac{2}{\pi}\int_0^\pi \cos(k\theta) g(s_j \tfrac{\lambda_{max}}{2}(\cos(\theta)+1))d\theta. \tag{67}$$

Exactly the same scheme is used to construct the M degree polynomial $p_0(x)$ for approximating the scaling function kernel h.

These Chebyshev coefficients may be computed independent of any particular knowledge of the graph signal f, beyond of knowing an appropriate spectral bound λ_{max}. Once they are obtained, the wavelet and scaling function coefficients for the fast SGWT are :

$$\begin{aligned}\tilde{W}_f(s_j,n) &= \left(\frac{1}{2}c_{j,0}f + \sum_{k=1}^{M} c_{j,k}\overline{T}_k(\mathscr{L})f\right)_n \\ \tilde{S}_f(n) &= \left(\frac{1}{2}c_{0,0}f + \sum_{k=1}^{M} c_{0,k}\overline{T}_k(\mathscr{L})f\right)_n\end{aligned} \tag{68}$$

The fast SGWT relies on computation of the terms $\overline{T}_k(\mathscr{L})f$ using the recurrence relation satisfied by the shifted Chebyshev polynomials. Equation (61) and the inverse change of variables $y = \frac{2}{\lambda_{max}} - 1$ shows $\overline{T}_k(x) = \frac{4}{\lambda_{max}}(x-1)\overline{T}_{k-1}(x) - \overline{T}_{k-2}(x)$, which immediately implies

$$\overline{T}_k(\mathscr{L})f = \frac{4}{\lambda_{max}}(\mathscr{L}-I)\left(\overline{T}_{k-1}(\mathscr{L})f\right) - \overline{T}_{k-2}(\mathscr{L})f \tag{69}$$

Critically, this recurrence relation can be calculated using only matrix-vector multiplication, storing only the vector result $\overline{T}_k(\mathscr{L})f$ for each $k \leq M$ and never explicitly computing the matrix $\overline{T}_k(\mathscr{L})$. The above recurrence shows that the vector $\overline{T}_k(\mathscr{L})f$ can be computed from the vectors $\overline{T}_{k-1}(\mathscr{L})f$ and $\overline{T}_{k-2}(\mathscr{L})f$, with computational cost dominated by matrix-vector multiplication by $\mathscr{L}-I$.

We estimate the computational complexity of computing the approximate SGWT this way, for a graph with a total number of nonzero edges $|E|$. If $\mathscr{L}$ is stored using a sparse matrix representation, then the cost of the matrix-vector produce $\mathscr{L}v$ for any $v \in \mathbb{R}^N$ is $O(|E|)$ (as opposed to $O(N^2)$ for full matrix-vector multiplication). For sparse graphs, where $|E|$ is small compared to N^2, this difference may be very significant. Computing all of the terms $\overline{T}_k(\mathscr{L})f$ for $k \leq M$ requires $O(M|E|)$ operations. We compute the wavelet and scaling function coefficients according to Equation (68), this may be done by adding the term $c_{j,k}\overline{T}_k(\mathscr{L})f$ for $j = 0,\ldots,J$ to a vector containing the j^{th} set of coefficients, as the terms $\overline{T}_k(\mathscr{L})f$ are computed. Computing the scalar-vector product $c_{j,k}\overline{T}_k(\mathscr{L})f$ and adding it to the running total vector requires $O(N)$ operations, this cost is incurred $M(J+1)$ times for computing each of the $J+1$ wavelet or scaling function bands, up to polynomial order M. All together, this implies a total computational cost of $O(M|E| + MN(J+1))$ to compute the SGWT via polynomial approximation.

As the recurrence relation (69) involves only three terms, computing all of the $\overline{T}_k(\mathscr{L})f$ may be done with memory of size $3N$ if the lower degree terms are overwritten once they are no longer needed for the recurrence. A straightforward implementation of the fast SGWT would also need enough memory to hold each of the $J+1$ wavelet or scaling function bands, implying a total memory size requirement of $N(J+1)+3N = N(J+4)$.

6.2 Polynomial approximation for the SGWT Adjoint operator

The SGWT wavelet and scaling function operators define linear mappings from $\mathbb{R}^N$ to the corresponding wavelet or scaling function coefficients. Once a set of J scales s_j is fixed, one may form the overall SGWT operator $W : \mathbb{R}^N \to \mathbb{R}^{N(J+1)}$ by concatenating all of the scaling function and wavelet coefficients into a single $N(J+1)$ length vector, as $Wf = \left((T_h f)^T, (T_g^{s_1} f)^T, \cdots, (T_g^{s_J} f)^T\right)^T$. Letting $\tilde{W} : \mathbb{R}^N \to \mathbb{R}^{N(J+1)}$ be the SGWT operator computed by polynomial approximation, we have

$$\tilde{W} f = \left((p_0(\mathscr{L})f)^T, (p_1(\mathscr{L})f)^T, \cdots, (p_J(\mathscr{L})f)^T\right)^T. \tag{70}$$

where the approximating polynomials p_j are defined in Equation (66).

Both the adjoint operator $\tilde{W}^T : \mathbb{R}^{N(J+1)} \to \mathbb{R}^N$ and the operator $W^T W : \mathbb{R}^N \to \mathbb{R}^N$ can be computed using Chebyshev polynomial approximation. These operators are used in the method we will detail in Section 6.3 for computing the inverse transformation. In addition, many wavelet based signal processing algorithms (in particular iterative algorithms for solving minimization problems arising from regularization using sparsity-promoting penalty functions of wavelet coefficients, see for example [46]) are described using the adjoint operator, so knowing that the adjoint may be efficiently computed is important for adapting such algorithms to graph signal processing using the SGWT.

The adjoint W^T is the linear operator such that $\left\langle W^T u, v\right\rangle = \langle u, Wv\rangle$ for every $u \in \mathbb{R}^{N(J+1)}$ and $v \in \mathbb{R}^N$. Below, we write $u \in \mathbb{R}^{N(J+1)}$ as the partitioned vector $u = (u_0^T, u_1^T, ..., u_J^T)$. We may then write

$$\begin{aligned}\langle u, Wv\rangle_{N(J+1)} &= \langle u, T_h v\rangle + \sum_{j=1}^{J} \left\langle u_j, T_g^{s_j} v\right\rangle_N \\ &= \left\langle T_h^T u_0, v\right\rangle + \left\langle \sum_{j=1}^{J} (T_g^{s_j})^T u_j, v\right\rangle_N = \left\langle T_h u_0 + \sum_{j=1}^{J} T_g^{s_j} u_j, f\right\rangle_N \end{aligned} \tag{71}$$

as the operators T_h and $T_g^{s_j}$ are all symmetric. Equation 71 shows that for any $u \in \mathbb{R}^{N(J+1)}$ viewed as the concatenation of $J+1$ coefficient subbands, application of the adjoint $W^T u$ is given by $W^T u = T_h u_0 + \sum_{j=1}^{J} T_g^{s_j} u_j$. Similarly, the adjoint of the approximate SGWT operator $\tilde{W}$ from Equation (66) is given as

$$\tilde{W}^T u = \sum_{j=0}^{J} p_j(\mathscr{L}) u_j. \tag{72}$$

This can be computed efficiently, using only matrix-vector multiplication, using exactly the same approach as described in Section 6.1.

From Equations (70) and (72) we see that

$$\tilde{W}^T \tilde{W} f = \sum_{j=0}^{J} p_j(\mathscr{L}) p_j(\mathscr{L}) f = \left(\sum_{j=0}^{J} (p_j(\mathscr{L}))^2 \right) f \tag{73}$$

This expression may be computed efficiently by determining the Chebyshev coefficients d_k for $0 \leq k \leq 2M$ for the 2M degree sum-of-squares polynomial $P(x) = \sum_{j=0} (p_j(x))^2$. Once these are calculated, we compute

$$\tilde{W}^T \tilde{W} f = \sum_{k=0}^{2M} d_k \overline{T}_k(\mathscr{L}) f. \tag{74}$$

We detail the determination of the coefficients d_k below.

The expression $T_k(x) = \cos(k \arccos(x))$, together with the trigonometric identity $\cos(\alpha)\cos(\beta) = \frac{1}{2}(\cos(\alpha+\beta) + \cos(\alpha-\beta))$ implies the the product relation

$$T_k(x) T_l(x) = \frac{1}{2} \left(T_{k+l}(x) + T_{|k-l|}(x) \right). \tag{75}$$

We will use this to express the $d'_k s$ in terms of the $c'_{j,k} s$. For convenience below we denote $c'_{j,k} = c_{j,k}$ for $k \geq 1$ and $c'_{j,0} = \frac{1}{2} c_{j,0}$, so that we may write $p_j(x) = \sum_{k=0}^{M} c'_{j,k} \overline{T}_k(x)$, without the factor of $\frac{1}{2}$ as in (66).

Similarly we define the coefficients $d'_{j,k}$ so that $(p_j(x))^2 = \sum_{k=0}^{2M} d'_{j,k} \overline{T}_k(x)$. Expanding $(p_j(x))^2$ and using (66) yields, after a lengthy but straightforward calculation, that

$$d'_{j,k} = \begin{cases} \frac{1}{2} \left({c'_{j,0}}^2 + \sum_{i=0}^{M} {c'_{j,i}}^2 \right) & \text{if } k = 0 \\ \frac{1}{2} \left(\sum_{i=0}^{k} c'_{j,i} c'_{j,k-i} + \sum_{i=0}^{M-k} c'_{j,i} c'_{j,k+i} + \sum_{i=k}^{M} c'_{j,i} c'_{j,i-k} \right) & \text{if } 0 < k \leq M \\ \frac{1}{2} \left(\sum_{i=k-M}^{M} c'_{j,i} c'_{j,k-i} \right) & \text{if } M < k \leq 2M \end{cases} \tag{76}$$

Defining $d_{n,0} = 2 d'_{j,0}$ and $d_{j,k} = d'_{j,k}$ for $k \geq 1$, we have $d_k = \sum_{j=0}^{J} d_{j,k}$. These are used with Equation 74 to compute $\tilde{W}^T \tilde{W} f$.

6.3 SGWT inverse transform

Many wavelet based signal processing algorithms function by computing wavelet coefficients of the original signal, manipulating the signal in the coefficient domain, and then inverting the wavelet transform. To be useful for signal processing, it is important to be able to invert the SGWT, i.e. to reconstruct a signal from a set of spectral graph wavelet coefficients. The scale discretized SGWT operator W computes $N(J+1)$ wavelet and scaling function coefficients for each N dimensional signal f. As the number of coefficients is greater than the dimension of the original signal, the SGWT is an overcomplete transform, and thus cannot have a unique linear inverse. Provided that the entire set of wavelet and scaling functions (equivalently, the columns of W^T) span $\mathbb{R}^N$, there will be infinitely many different left inverse matrices M satisfying $MW = I$. We note that this condition holds if the frame bound A from Theorem 1 is positive.

We use the pseudoinverse, formally given by $M = (W^T W)^{-1} W^T$, as the inverse of the SGWT. For a given set of SGWT coefficients $c \in \mathbb{R}^{N(J+1)}$, the inverse SGWT will be the signal $f \in \mathbb{R}^N$ obtained by solving the linear system $(W^T W) f = W^T c$. For most applications this system is too large to be solved directly (for instance by the LU factorization and back substitution). Instead, we employ the well-known conjugate gradients algorithm [47]. This is an iterative algorithm, the computational cost at each step is dominated by computing the product of $W^T W$ with a single vector. We use the Chebyshev polynomial approximation scheme for computing $\tilde{W}^T \tilde{W}$ in each step of the conjugate gradients algorithm.

We note that the frame bounds from Theorem 1 may be used to estimate the convergence speed of the conjugate gradients iteration. The remaining error in the conjugate gradients algorithm after i iterations (as measured by the norm of the residual) is bounded by the norm of the first residual times

$$2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i, \tag{77}$$

where κ is the ratio of the largest and smallest eigenvalues of $W^T W$. The frame bounds A and B are bounds on the spectrum of $W^T W$ (see [39]), and thus $\kappa < A/B$. This explicitly shows that the convergence properties for the conjugate gradients reconstruction depend on the frame bounds, with faster convergence for smaller A/B.

7 SGWT kernel design details

The described theory of the SGWT places few constraints on the wavelet kernel g, scaling function kernel h, or the selection of scales. We give details of the design choices described in the original paper [32], which are also those used in the example illustrative images included later in this chapter. The wavelet kernel g is chosen

to give exact localization in the limit of small scales. By Theorem 2, this will occur if $g(x)$ behaves like a power of x near the origin. We ensure this by choosing g to be an exact monic polynomial for x in a neighborhood of the origin. For large x, g should decay to zero. This is enforced by setting g to decay as a negative power of x, for x larger than some fixed value. The final design connects these two regions by a cubic spline ensuring continuity of g and its derivative. Specifically, we set

$$g(x;\alpha,\beta,x_1,x_2) = \begin{cases} x_1^{-\alpha}x^{\alpha} & \text{for } x < x_1 \\ s(x) & \text{for } x_1 \leq x \leq x_2 \\ x_2^{\beta}x^{-\beta} & \text{for } x > x_2 \end{cases} \tag{78}$$

where α and β are integers, and x_1 and x_2 specify the transition region between the monic polynomial and decaying regions. The examples included in this chapter used α and β both set to 2, $x_1 = 1$ and $x_2 = 2$. In this case the cubic polynomial $s(x) = -5 + 11x - 6x^2 + x^3$.

The discrete scale values s_j are chosen by first specifying the maximum scale s_1 and the minimum scale s_J, then setting the intermediate scales decreasing and logarithmically equally spaced as $s_j = s_1(\frac{s_J}{s_1})^{\frac{j-1}{J-1}}$ for $1 \leq j \leq J$.

The minimum and maximum scales are adapted to the spectrum of $\mathscr{L}$ as follows. Given an upper bound λ_{max} on the spectrum of $\mathscr{L}$, and a value K that is considered a design parameter of the transform, we set $\lambda_{min} = \lambda_{max}/K$. The scales s_1 and s_J are chosen so that the smallest scale kernel $g(s_J x)$ is a monic polynomial over the interval $[0, \lambda_{max}]$, and so that the largest scale kernel $g(s_1 x)$ decays as $x^{-\beta}$ over the interval $[\lambda_{min}, \infty)$. This is ensured by setting $s_1 = x_2/\lambda_{min}$ and $s_J = x_1/\lambda_{max}$.

The scaling function kernel $h(x)$ is set as $h(x) = \gamma\exp(-(\frac{x}{0.6\lambda_{min}})^4)$. Here γ is determined by the condition that that $h(0)$ has the same value as the maximum value of g. An illustration of these choices for the scaling function and scaled wavelet kernels, for $\lambda_{max} = 20$, $K = 20$, $J = 4$, $\alpha = 2$, $\beta = 2$, $x_1 = 1$ and $x_2 = 2$ is given in Figure 1.

We note that many other design choices for the wavelet and scaling function kernels are possible. In particular, Leonardi and Van De Ville have developed a design leading to a SGWT that is a tight frame, i.e. where the bounds A and B from (34) are equal [48].

8 Illustrative Examples

In order to illustrate the SGWT, we provide several examples of weighted graphs arising from different application areas, and present images of some wavelets and scaling functions for these graphs. As a first example, we consider a point cloud sampled randomly from the "swiss roll", a 2 dimensional manifold embedded that is widely used as a benchmark example for dimensionality reduction and manifold learning algorithms [49]. Our example is based on sampling 500 points from the

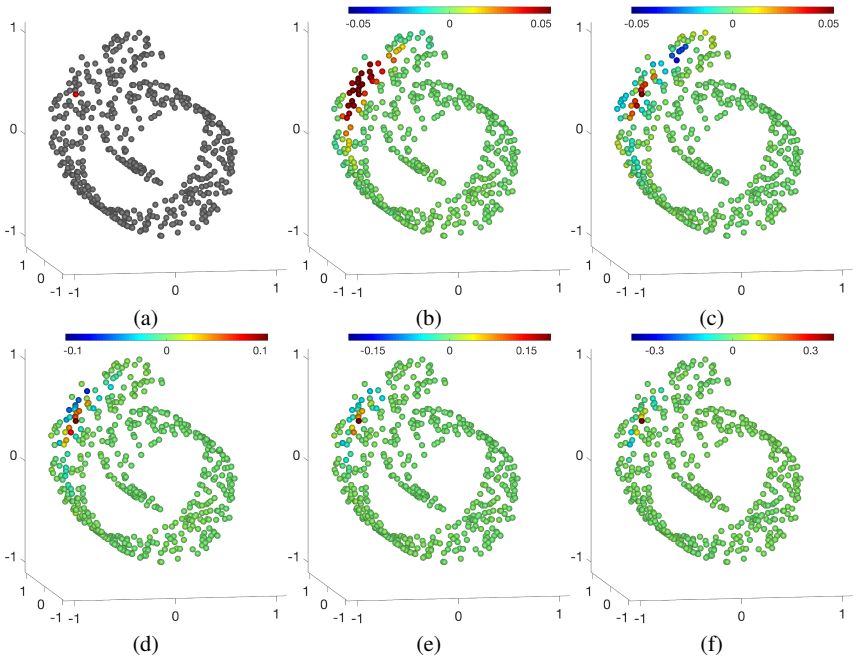

Fig. 4 Spectral graph wavelets on Swiss Roll data cloud, for transform with $J = 8$ wavelet scales (showing only wavelets for the 4 coarsest scales) . (a) vertex at which wavelets are centered (b) scaling function (c)-(f) wavelets, scales 1-4. Figure adapted from [32].

embedding in $\mathbb{R}^3$ given parametrically by $x(u,v) = (v\cos(v)/4\pi, u, v\sin(v)/4\pi)$ for $-1 \leq u \leq 1$, $\pi \leq v \leq 4\pi$.

The adjacency for the weighted graph A is computed from the points x_i by assigning a greater edge weight to edges connecting points that are close in $\mathbb{R}^3$. Specifically, we set $A_{i,j} = \exp(-\left|\left|x_j - x_j\right|\right|^2/2\sigma^2)$, with $\sigma = 0.1$. We show the point cloud, a scaling function and four wavelets all centered on the same vertex, in Figure 4.

The swiss roll point cloud is a toy example of data constrained to a lower dimensional manifold that are embedded in a higher dimensional space, a situation which commonly arises in many examples relevant to machine learning. We note in particular that the support of the wavelets and the scaling function automatically adapts to the structure of the underlying 2d manifold, and does not jump across to the upper portion of the roll, even though the geometric separation in the 3d embedding space is smaller in some cases than the diameter of the support of the wavelet or scaling function.

We next consider an example transportation network, arising from a graph describing the road network in Minnesota. Here edges correspond to major roads, each vertex is the intersection of two roads. The vertices thus do not always exactly correspond to population centers (incorporated towns or cities), although many do. For this particular dataset the edges are unweighted, and do not for instance reflect

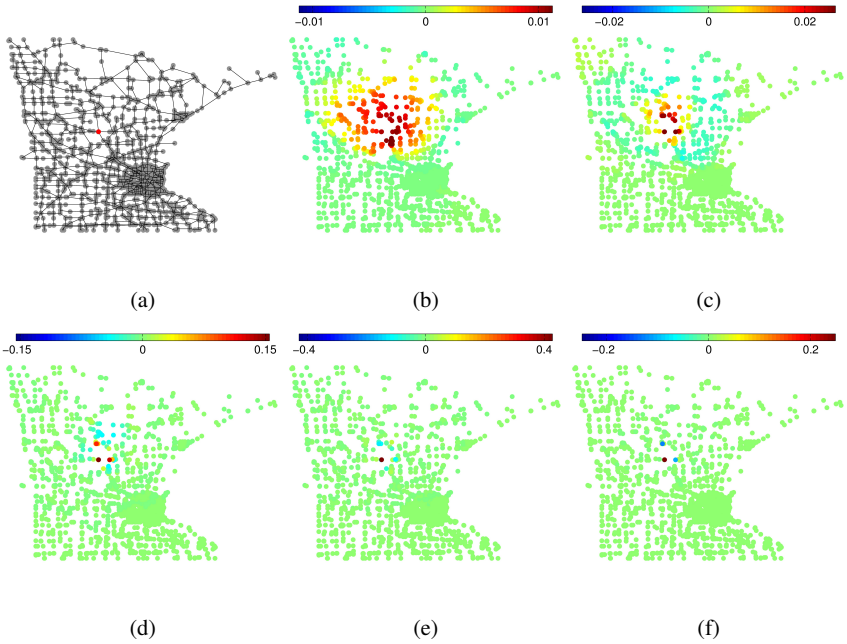

Fig. 5 Spectral graph wavelets on Minnesota road graph, with $K = 100$, $J = 4$ scales. (a) vertex at which wavelets are centered (b) scaling function (c)-(f) wavelets, scales 1-4. Reproduced with permission from [32].

the capacity of the road. Figure 5 shows a set of wavelets and scaling function centered at a single vertex, for the SGWT computed with parameter $K = 100$ and $J = 4$ scales. We note that for display purposes each vertex has associated 2d coordinates, however these were used only for rendering the figure and were not used for the actual computation of the SGWT.

With an eye towards applications, we note that the SGWT be useful for analysis of data measured at vertices of a transportation network where the phenomena generating the measured data was influenced in some way by the transportation network. Possible examples could include analysis of data describing disease rates during an epidemic (if it were expected that transportation network could influence patterns of disease transmission), or analysis of inventory data for goods that are moved along the transportation network.

A third example shows the SGWT appropriate for data measured on irregularly shaped domains. We take as an example irregular domain the geometry of the surface of Lake Geneva. The SGWT for this case could be used for analysis or processing of some physical measurement (such as water temperature, or concentration of some solute) that was taken at regularly spaced points on the surface of the body of water. Using classical wavelet analysis for such data would require some special handling of the geometrically complex boundary between land and water. In con-

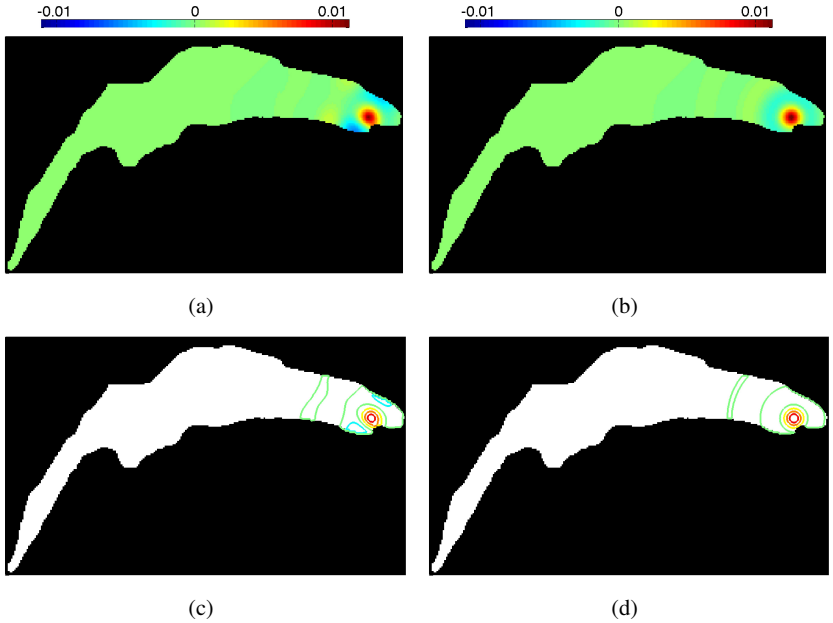

Fig. 6 Spectral graph wavelets on lake Geneva domain, (spatial map (a), contour plot (c)); compared with truncated wavelets from graph corresponding to complete mesh (spatial map (b), contour plot (d)). Note that the graph wavelets adapt to the geometry of the domain. Reproduced with permission from [32].

trast, the SGWT implicitly handles the boundary, and needs no special adaptation beyond encoding the domain with the adjacency matrix.

In this example the geometry of the lake surface is described as a binary function on a regular rectangular grid. The graph is constructed by retaining only vertices corresponding to grid points within the lake interior, and with edges connecting the (at most 4) neighboring vertices. We use a binary graph (all edge weights set to 1). At interior points of the domain, the graph Laplacian thus corresponds to the standard 5-point stencil for approximating $-\nabla^2$ (see Equation (14)), while at points on the boundary the Laplacian is modified by the deletion of grid points outside of the lake domain. We constructed our lake geometry mask using shoreline data from the GSHHS database [50]. The lake mask was calculated on a 256 x 153 pixel grid (corresponding to a physical scale of 232 meteres per pixel). We show a single wavelet for the largest scale value in Figure 6, for the design with parameter $K = 100$ and $J = 5$ scales.

To illustrate the implicit adaption of the wavelet to the geometry of the domain, we compare it with a SGWT wavelet computed with the same wavelet kernel and scale parameter for a large regular grid, that is simply truncated. These true and truncated wavelets will coincide for central vertices that are far from the boundary, however they may be very different for wavelets centered on vertices near the

boundary. This can be seen clearly in Figure 6, which illustrates how the SGWT adapts to an irregular boundary.

9 Conclusion

We have described a wavelet transform for data defined on the vertices of arbitrary weighted graphs. Our approach uses spectral graph theory, based on the eigenvectors and eigenvalues of the graph Laplacian matrix, to define a notion of scaling that is analogous to classical wavelet operators. Our graph wavelet operators are defined by taking kernel functions of the graph Laplacian, where the kernel functions are formed by rescaling a single bandpass function. We have shown that defining the graph wavelets by applying these wavelet operators to a delta impulse centered on a single vertex gives wavelets that are localized in the limit of small scales. A fast algorithm based on Chebyshev polynomial approximation was described, demonstrating that the SGWT can be applied to large graphs without the need for explicit computation of the eigenvectors and eigenvalues of the graph Laplacian matrix. We studied the frame bounds of the SGWT, and described a computation of the inverse transform. Finally, we showed a series of example images of the SGWT computed for several different graphs, highlighting potential applications of the transform.

Software Implementation

A MATLAB toolbox with complete functionality for computing the SGWT may be found online at wiki.epfl.ch/sgwt. Much of this functionality has also been incorporated into the larger Graph Signal Processing toolbox [51] (GSPBox), a MATLAB implementation is available at epfl-lts2.github.io/gspbox-html, and a Python implementation may be found at https://pygsp.readthedocs.io/en/stable.

References

1. A. Grossmann and J. Morlet, "Decomposition of Hardy functions into square integrable wavelets of constant shape," *SIAM Journal on Mathematical Analysis*, vol. 15, no. 4, pp. 723–736, 1984.
2. I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 41, no. 7, pp. 909–996, 1988.
3. S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, Jul 1989.
4. Y. Meyer, *Orthonormal Wavelets. In : Wavelets. Inverse Problems and Theoretical Imaging*. Springer, 1989.

5. G. Beylkin, R. Coifman, and V. Rokhlin, "Fast wavelet transforms and numerical algorithms i," *Communications on Pure and Applied Mathematics*, vol. 44, no. 2, pp. 141–183, 1991.
6. D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, 1994.
7. S. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *Image Processing, IEEE Transactions on*, vol. 9, pp. 1532–1546, Sep 2000.
8. L. Sendur and I. Selesnick, "Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency," *Signal Processing, IEEE Transactions on*, vol. 50, pp. 2744–2756, Nov 2002.
9. J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 12, pp. 1338–1351, 2003.
10. I. Daubechies and G. Teschke, "Variational image restoration by means of wavelets: Simultaneous decomposition, deblurring, and denoising," *Applied and Computational Harmonic Analysis*, vol. 19, no. 1, pp. 1 – 16, 2005.
11. F. Luisier, C. Vonesch, T. Blu, and M. Unser, "Fast interscale wavelet denoising of poisson-corrupted images," *Signal Processing*, vol. 90, no. 2, pp. 415 – 427, 2010.
12. J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *Signal Processing, IEEE Transactions on*, vol. 41, pp. 3445–3462, Dec 1993.
13. A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 6, pp. 243–250, Jun 1996.
14. M. Hilton, "Wavelet and wavelet packet compression of electrocardiograms," *Biomedical Engineering, IEEE Transactions on*, vol. 44, pp. 394–402, May 1997.
15. R. Buccigrossi and E. Simoncelli, "Image compression via joint statistical characterization in the wavelet domain," *Image Processing, IEEE Transactions on*, vol. 8, pp. 1688–1701, Dec 1999.
16. D. Taubman and M. Marcellin, *JPEG2000 : Image compression fundamentals, standards and practice*. Kluwer Academic Publishers, 2002.
17. J.-L. Starck and A. Bijaoui, "Filtering and deconvolution by the wavelet transform," *Signal Processing*, vol. 35, no. 3, pp. 195 – 211, 1994.
18. D. L. Donoho, "Nonlinear solution of linear inverse problems by wavelet-vaguelette decomposition," *Applied and Computational Harmonic Analysis*, vol. 2, no. 2, pp. 101 – 126, 1995.
19. E. Miller and A. S. Willsky, "A multiscale approach to sensor fusion and the solution of linear inverse problems," *Applied and Computational Harmonic Analysis*, vol. 2, no. 2, pp. 127 – 147, 1995.
20. R. Nowak and E. Kolaczyk, "A statistical multiscale framework for Poisson inverse problems," *Information Theory, IEEE Transactions on*, vol. 46, pp. 1811–1825, Aug 2000.
21. J. Bioucas-Dias, "Bayesian wavelet-based image deconvolution: a GEM algorithm exploiting a class of heavy-tailed priors," *Image Processing, IEEE Transactions on*, vol. 15, pp. 937–951, April 2006.
22. J. R. Wishart, "Wavelet deconvolution in a periodic setting with long-range dependent errors," *Journal of Statistical Planning and Inference*, vol. 143, no. 5, pp. 867 – 881, 2013.
23. F. K. Chung, *Spectral Graph Theory*, vol. 92 of *CBMS Regional Conference Series in Mathematics*. AMS Bookstore, 1997.
24. M. Crovella and E. Kolaczyk, "Graph wavelets for spatial traffic analysis," *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, vol. 3, pp. 1848 – 1857 vol.3, Jan 2003.
25. A. Smalter, J. Huan, and G. Lushington, "Graph wavelet alignment kernels for drug virtual screening," *Journal of Bioinformatics and Computational Biology*, vol. 7, pp. 473–497, 2009.
26. M. Jansen, G. P. Nason, and B. W. Silverman, "Multiscale methods for data on graphs and irregular multidimensional situations," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 71, no. 1, pp. 97–125, 2009.
27. F. Murtagh, "The Haar wavelet transform of a dendrogram," *Journal of Classification*, vol. 24, pp. 3–32, Jun 2007.

28. A. B. Lee, B. Nadler, and L. Wasserman, "Treelets - an adaptive multi-scale basis for sparse unordered data," *Annals of Applied Statistics*, vol. 2, pp. 435–471, 2008.
29. R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Applied and Computational Harmonic Analysis*, vol. 21, pp. 53–94, Jan 2006.
30. M. Maggioni and H. Mhaskar, "Diffusion polynomial frames on metric measure spaces," *Applied and Computational Harmonic Analysis*, vol. 24, no. 3, pp. 329–353, 2008.
31. D. Geller and A. Mayeli, "Continuous wavelets on compact manifolds," *Mathematische Zeitschrift*, vol. 262, pp. 895–927, 2009.
32. D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129 – 150, 2011.
33. D. Thanou, D. I. Shuman, and P. Frossard, "Learning parametric dictionaries for signals on graphs," *IEEE Transactions on Signal Processing*, vol. 62, pp. 3849–3862, Aug 2014.
34. W. H. Kim, D. Pachauri, C. Hatt, M. K. Chung, S. Johnson, and V. Singh, "Wavelet based multi-scale shape features on arbitrary surfaces for cortical thickness discrimination," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1241–1249, Curran Associates, Inc., 2012.
35. W. H. Kim, M. K. Chung, and V. Singh, "Multi-resolution shape analysis via non-euclidean wavelets: Applications to mesh segmentation and surface alignment problems," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2139–2146, June 2013.
36. N. Tremblay and P. Borgnat, "Multiscale community mining in networks using spectral graph wavelets," in *21st European Signal Processing Conference (EUSIPCO 2013)*, pp. 1–5, Sept 2013.
37. M. Zhong and H. Qin, "Sparse approximation of 3d shapes via spectral graph wavelets," *The Visual Computer*, vol. 30, pp. 751–761, Jun 2014.
38. M. Reed and B. Simon, *Methods of Modern Mathematical Physics Volume 1 : Functional Analysis*. Academic Press, 1980.
39. I. Daubechies, *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.
40. C. E. Heil and D. F. Walnut, "Continuous and discrete wavelet transforms," *SIAM Review*, vol. 31, no. 4, pp. 628–666, 1989.
41. D. Watkins, *The Matrix Eigenvalue Problem - GR and Krylov subspace methods*. Society for Industrial and Applied Mathematics, 2007.
42. G. L. G. Sleijpen and H. A. V. der Vorst, "A Jacobi–Davidson iteration method for linear eigenvalue problems," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 2, pp. 401–425, 1996.
43. K. O. Geddes, "Near-minimax polynomial approximation in an elliptical region," *SIAM Journal on Numerical Analysis*, vol. 15, no. 6, pp. 1225–1233, 1978.
44. W. Fraser, "A survey of methods of computing minimax and near-minimax polynomial approximations for functions of a single independent variable," *J. Assoc. Comput. Mach.*, vol. 12, pp. 295–314, 1965.
45. G. M. Phillips, *Interpolation and Approximation by Polynomials*. CMS Books in Mathematics, Springer-Verlag, 2003.
46. I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.
47. J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," tech. rep., Pittsburgh, PA, USA, 1994.
48. N. Leonardi and D. V. D. Ville, "Tight wavelet frames on multislice graphs," *IEEE Transactions on Signal Processing*, vol. 61, pp. 3357–3367, July 2013.
49. J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
50. P. Wessel and W. H. F. Smith, "A global, self-consistent, hierarchical, high-resolution shoreline database," *J Geophys. Res.*, vol. 101(B4), pp. 8741–8743, 1996.
51. N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, "GSPBOX: A toolbox for signal processing on graphs," *ArXiv e-prints*, Aug. 2014.