Graph Fourier Transform: A Stable Approximation

João Domingos and José M. F. Moura, *Life Fellow, IEEE*

***Abstract*—In graph signal processing (GSP), data dependencies are represented by a graph whose nodes label the data and the edges capture dependencies among nodes. The graph is represented by a weighted adjacency matrix A that, in GSP, generalizes the Discrete Signal Processing (DSP) shift operator z^{-1}. The (right) eigenvectors of the shift A (graph spectral components) diagonalize A and lead to a graph Fourier basis F that provides a graph spectral representation of the graph signal. The inverse of the (matrix of the) graph Fourier basis F is the Graph Fourier transform (GFT), F^{-1}. Often, including in real world examples, this diagonalization is numerically unstable. This paper develops an approach to compute an accurate approximation to F and F^{-1}, while insuring their numerical stability, by means of solving a non convex optimization problem. To address the non-convexity, we propose an algorithm, the stable graph Fourier basis algorithm (SGFA) that improves exponentially the accuracy of the approximating F per iteration. Likewise, we can apply SGFA to A^H and, hence, approximate the stable left eigenvectors for the graph shift A and directly compute the GFT. We evaluate empirically the quality of SGFA by applying it to graph shifts A drawn from two real world problems, the 2004 US political blogs graph and the Manhattan road map, carrying out a comprehensive study on tradeoffs between different SGFA parameters. We also confirm our conclusions by applying SGFA on very sparse and very dense directed Erdős-Rényi graphs.**

***Index Terms*—Graph signal processing, graph Fourier basis, graph Fourier transform, eigendecomposition, numerical stability, Manhattan road map, political blogs.**

I. Introduction

A TRUISM of current technology is the explosive growth of data—*Big Data*, characterized variously by[1] three, four, five, or seven V's. Of these, we focus on variety that may reflect different data formats, arising from diverse data sources and applications. Examples include tweets, networks of coauthors and citations [1], hyperlinked blogs [2], phone users, customers of service providers, friends in a social network, individuals in populations, traffic flows in urban environments [3], [4], or physical measurements like temperature or humidity in a network of meteorological stations spread over a whole country, among many other examples, e.g., [5], [6], of current interest. Data from these applications contrast with traditional time series, audio or speech signals, images, video, or point clouds where the data samples are indexed by time instants, pixels, voxels, or pixels and time taking values on a regular one-, two-, three- or even four-dimensional grid. This variety of Big Data also often leads to data being referred to as *un*structured, not fitting neatly on a table. This *un*structured data is structured through a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where nodes or vertices in the set $\mathcal{V}$ represent the sources or labels of the data and (directed or undirected) edges in the set $\mathcal{E}$ capture dependencies or relations among the data of different nodes. For example, traffic counts in a metropolitan area are directly dependent if corresponding to close by locations of the same street. In this paper, the graph $\mathcal{G}$ is general and characterized by an adjacency matrix A [7]. As explained in Section II, in Graph Signal Processing (GSP) [8], the adjacency matrix A plays the role of the shift z^{-1} in discrete signal processing (DSP) [9]–[11]. We refer to A as the graph shift. In GSP, the eigenvalues $\{\lambda_i\}$ and (right) eigenvectors $\{f_i\}$, $1 \leq i \leq n$, of A are the graph frequencies and graph spectral components [8], [12], respectively, extending to signals defined on a graph $\mathcal{G}$ the common concepts of frequency and harmonic components for time signals. The matrix F of the eigenvectors or graph spectral components of A will be referred to as the graph Fourier basis. If A is diagonalizable, the inverse of F is the graph Fourier transform[2] (GFT), $F^{-1} = W^H$. The matrix F and its inverse $F^{-1} = W^H$ are unique up to choice of basis in the graph signal space [14], [15]. In DSP, the GFT is the discrete Fourier transform (DFT).

For generic directed graphs (digraphs), A is not symmetric, the eigenvalues λ_i may be complex valued, its eigenvectors or graph Fourier components f_i are not orthogonal and the graph Fourier basis F is not unitary $(F^{-1} \neq F^H)$ or fail to be a complete basis [8], [13], and the columns f_i of F are in general not of unit norm. Authors have taken different approaches to avoid these issues and preserve F to be unitary. For example, 1) [16], [17] redefine the shift matrix to make it a norm preserving operator; 2) [18], [19] consider the case of a normal shift ($AA^H = A^H A$ and A is unitarily diagonalizable, but with eigenvalues not necessarily real valued); 3) more often, the literature considers the graph to be undirected, so, the shift is symmetric, or take the shift to be the graph Laplacian L [12], either of which is diagonalizable by a unitary operator[3]; and 4)

Manuscript received August 1, 2019; revised January 12, 2020 and April 30, 2020; accepted July 5, 2020. Date of publication July 20, 2020; date of current version August 18, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gwo Giun Lee. This work was supported by NSF under Grants CCF 1513936 and CPS 1837607. *(Corresponding author: José M. F. Moura.)*

João Domingos is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15217 USA, and also with Instituto Sistemas e Robótica, Instituto Superior Técnico, Universidade de Lisboa, 1649-004 Lisboa, Portugal (e-mail: joaodomingos42@hotmail.com).

José M. F. Moura is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15217 USA (e-mail: moura@andrew.cmu.edu).

Digital Object Identifier 10.1109/TSP.2020.3009645

[1] Volume, variety, velocity, veracity, value, variability, and visualization.

[2] If A is not diagonalizable, see [13] on how to define the GFT.

[3] The Laplacian L is a second order operator like a second order derivative or difference, while A is first order like first order derivative or difference. Adopting L as shift is then equivalent to working with a time shift z^{-2} in

[20] considers directed graphs but redefines the GFT to keep it orthogonal and avoid Jordan decompositions. However, it may be hard to develop a graph filtering theory for [20].

Our goal is to consider spectral analysis for data supported by generic graphs—graphs not restricted by any special structure—to address the difficulties associated with finding a numerically stable graph Fourier transform and Fourier basis F. These problems may stem from 1) the lack of orthogonality between the f_i (eigenvectors of A and columns of F), some of which may be close to being parallel, 2) possible large scale differences between the f_i's, and 3) for defective A, the set of eigenvectors $\{f_i\}$ not being a complete basis and A not being diagonalizable. We develop an optimization approach to approximately diagonalize the matrix A by a numerically stable, non unitary, Fourier basis F. The algorithm provides a tunable tradeoff between the accuracy of the diagonalization (measured for example by the error[4] $\|AF - F\Lambda\|_{\mathcal{F}}$, where Λ is the diagonal matrix of eigenvalues of A) and the stability of F (measured by the minimum singular value $\sigma_{\min}(F)$).

In fact, attempting to diagonalize shift matrices (or adjacency matrices) of generic graphs by standard methods like the routine `eig` in MATLAB leads often to highly numerically unstable Fourier basis F, i.e., the matrix F has a very large condition number[5] κ and a very small minimum singular value $\sigma_{\min}(F) \ll 10^{-12}$. Numerically unstable Fourier basis F is highly undesirable since then the GFT, F^{-1}, is badly scaled, becoming difficult to carry out graph spectral analysis, graph spectral decompositions, graph Fourier transforms, and related concepts in GSP. To confirm that such matrices occur frequently, we study the class of directed Erdős-Rényi random graphs generated with probability of connection $p \in [0\ \ 1]$. For not very large values of p, Erdős-Rényi random graphs are highly sparse, a characteristic found in many real world graphs. For completeness, we include numerical studies for two types of Erdős-Rényi random graphs: a) models without self-loops, obtained when the probability of connection between a node and itself is zero, and 2) models with possible self-loops where a node may connect with itself with probability p. We consider 100 values of p discretizing $[0\ \ 1]$. For each of these 100 values of p, we generated 10^3 directed Erdős-Rényi graphs for both models (without and with possible self-loops). We then used MATLAB `eig` to find the Fourier basis F diagonalizing their adjacency matrix A and computed for each such F the corresponding $\sigma_{\min}(F)$. Fig. 1 plots the empirical probability that $\sigma_{\min}(F) \leq 10^{-12}$ for forty values of p tested in the ranges[6] $[0\ \ .09]$ and $[.93\ \ 1]$, corresponding to very low and very high probability of connection p. The value of 10^{-12} upper bounding

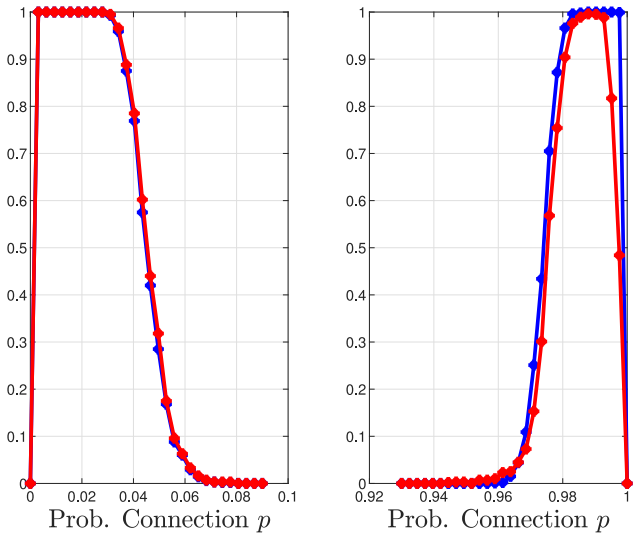


Fig. 1. Empirical probability that $\sigma_{\min}(F) \leq 10^{-12}$ vs. probability of connection p for two types of Erdős-Rényi models with $n = 100$ nodes: in red, models without self-loops and, in blue, models with possible self-loops. Left and right tails for 1000 Monte Carlo trials.

$\sigma_{\min}(F)$ reflects the very bad conditioning of the Fourier basis F computed by MATLAB `eig`. In both models, the extreme points $p = 0$ and $p = 1$ yield a numerically stable Fourier basis F with probability one since the generated (isolated nodes and complete) graphs are trivially undirected. As can be seen, for both with (blue plot) and without (red plot) self-loops and for both very sparse and very dense random *directed* Erdős-Rényi graph models, the Fourier bases F computed by MATLAB `eig` tend to be numerically unstable with high probability. When we allow for self-loops (red plot), we note for high values of p that the (red) plot has a slightly smoother transition to the extreme (completely connected) case $p = 1$. Beyond random networks, this is also the case with real world networks, like we observed in other work, when we analyzed [8] the political blogs network [2] shown in Fig. 2(b) and when we studied [3], [4] the Manhattan road map shown in Fig. 2(c). Using MATLAB `eig` leads to F with $\sigma_{\min}(F) \approx 10^{-33}$ for the political blogs and $\sigma_{\min}(F) \approx 10^{-19}$ for the Manhattan road map. We expand on these in Sections V and VI, respectively.

Remark 1: As we mentioned, the paper gives a method to find an approximate stable approximation to the Fourier basis F when the eigenvector matrix is unstable either because 1) no full rank F exists; or 2) full rank F may exist, but the computed F is numerically unstable. A Fourier basis F may not exist because the shift A is defective. This may occur when the graph shift has a repeated eigenvalue λ with (numerically computed) algebraic multiplicity greater than one.[7] As studied in [13], this is the case with the political blogs network and the Manhattan road map, where, in both cases, as computed with MATLAB `eig`, $\lambda = 0$ has algebraic multiplicity of several hundred (see below in Fig. 12(a) a plot of the eigenvalues for the Manhattan road

traditional DSP or linear systems, or to restrict signals to have even symmetry like autocorrelations and filters to be polynomials (or rational functions) in powers of z^{-2}. In contrast, working with A as shift corresponds in DSP to the time shift z^{-1} and filters to be polynomials (or rational functions) in z^{-1}.

[4]The subindex $\mathcal{F}$ stands for Frobenius norm, $\|A\|_{\mathcal{F}} = \sqrt{\sum_{i,j} A_{i,j}^2}$.

[5]Empirically, we observed that the maximum singular value $\sigma_{\max}(F)$ is not very large. It is the $\sigma_{\min}(F)$ that is very small. We focus this discussion on $\sigma_{\min}(F)$.

[6]Results for other values of p are not shown since the empirical probability that $\sigma_{\min}(F) \leq 10^{-12}$ is zero.

[7]A matrix is diagonalizable iff for every eigenvalue the algebraic and geometric multiplicities are equal.

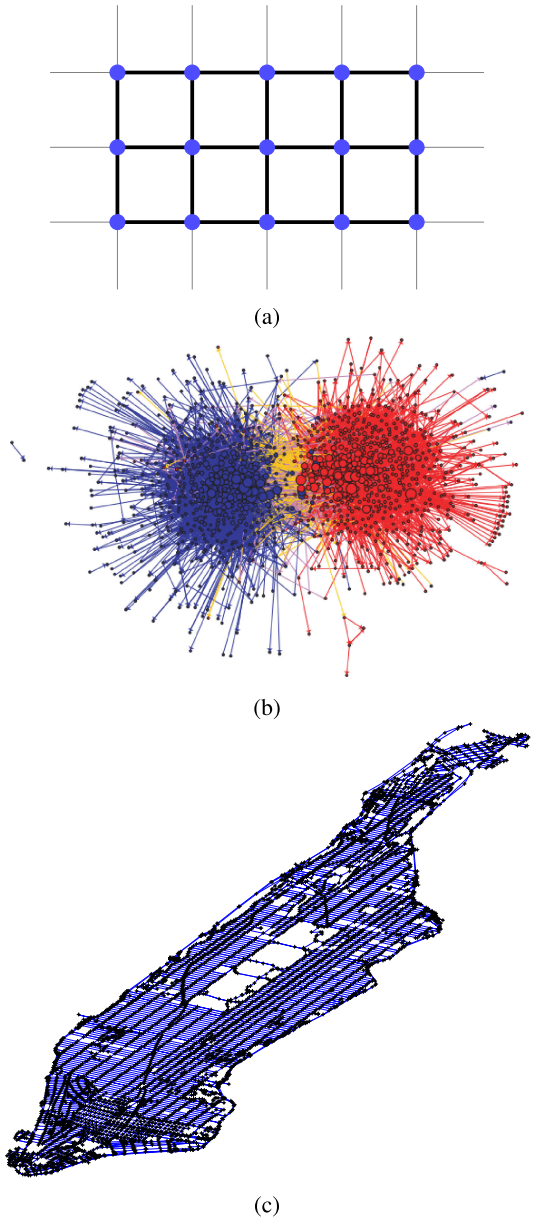


Fig. 2. Different graphs in the context of GSP. (a) A 2D rectangular lattice indexing pixels in a digital image; (b) graph of political blogs; and (c) Manhattan road map.

map). Numerically computing an eigenvalue with large algebraic multiplicity is a well known difficult problem. It may give rise to a cluster of values with the true eigenvalue possibly at its center [21], [22] or the true eigenvalue may be best approximated by a pseudoeigenvalue [23], [24]. Rounding errors may make it not possible to determine if an eigenvalue is actually a repeated eigenvalue or a cluster of (not necessarily close together) eigenvalues. In other words, determining true versus spurious eigenvalues in the numerically computed spectrum of the shift matrix may be ambiguous. Further, as observed by [25], with repeated eigenvalues, the direct computation of the eigendecomposition of the shift A is in general numerically unstable (even more so, if it involves the Jordan decomposition [13], [26] where a small perturbation of the input matrix may drastically change the computed Jordan form). The paper will not dwell further on this; we simply state that our algorithm can be applied to compute a stable F and stable F^{-1} that accurately approximate the diagonalization of A in either the no full rank F case 1) or the full rank F case 2) above. ■

The examples above of random graphs and real world applications point to the difficulty encountered often of computing the Fourier basis F and its inverse, the GFT, F^{-1}. They confirm the interest and the need to develop a numerically efficient method to find an approximate stable Fourier basis F, our goal with this work. This paper develops SGFA, the stable graph Fourier basis algorithm, to compute an *accurate stable* approximation to F for generic digraph shifts A (not necessarily symmetric). SGFA is a two stage procedure that, iteratively, attempts 1) to diagonalize a triangular decomposition of A, and 2) to optimize the numerical stability of the resulting F. There are trade-offs between accuracy (degree of diagonalizability of the decomposition as measured by how close the numerically stable approximation of F is to the true F) and stability (well conditioning of the resulting approximation of F).

We can compute the GFT, F^{-1}, by numerically inverting the graph Fourier basis F obtained with SGFA. But this may affect its numerical accuracy, since we first compute the (right) eigenvectors and then invert the Fourier basis F. To avoid this, we compute directly the GFT by applying SGFA to A^H, since the right eigenvectors of A^H are the left eigenvectors[8] of A. In other words, diagonalizing A^H with SGFA computes directly a stable approximation to the GFT, F^{-1}, without inverting the SGFA computed F. Since we do not know the true F and true GFT, F^{-1}, applying SGFA to both A and A^H provides the opportunity to study the quality of the numerically computed graph Fourier basis F by comparing its numerically computed inverse with the SGFA directly computed GFT, and, vice-versa, comparing the directly computed graph Fourier basis with the (numerically computed) inverse of the matrix of (left) eigenvectors.

To analyze tradeoffs between accuracy and stability and to confirm the quality of the diagonalization and of the approximations to F, we carry out a number of studies: 1) We evaluate how close $F \cdot F^{-1}$ and $F^{-1} \cdot F$ are to the identity matrix I, where F^{-1} is the numerically computed inverse of F. If SGFA computes accurately a stable F, then F^{-1} is accurately computed, and $F \cdot F^{-1}$ and $F^{-1} \cdot F$ should be close to I. 2) We test how close the columns of F are to eigenvectors of A by comparing $A \cdot F$ to $F \cdot \Lambda$, where Λ is the diagonal matrix of the graph frequencies (eigenvalues of A). This is carried out both in terms of how close the magnitudes of $A \cdot f_i$ and $\lambda_i f_i$ are, where f_i is the eigenvector of A corresponding to the eigenvalue λ_i, and in terms of how aligned $A \cdot f_i$ and f_i are (how small the angle between the two vectors is). 3) We compare how close the inverse of the SGFA computed graph Fourier basis F is to the (Hermitian of the) matrix of directly computed left eigenvectors of A and vice-versa. We carry out extensive empirical studies with two real world graphs, the political blogs network [2] and

[8]Let $A = F\Lambda F^{-1} = W^{-H}\Lambda W^H$, then $W^H A = \Lambda W^H$, $A^H = W\Lambda^\star W^{-1}$, and the columns of W are the left eigenvectors of A and W is the Fourier basis of A^H.

the Manhattan Road Map [3], [4]. The random models of Fig. 1 are also considered in Section VII.

We summarize the remaining of the paper. In Section II, we briefly review GSP concepts and present two interesting graphs arising in applications. Section III introduces the problem of computing numerically the graph Fourier basis F for generic graphs A. Section IV introduces, motivates, and analyzes the Stable Graph Fourier basis Approximation (SGFA) algorithm. We prove that the accuracy of our approximation to F improves (at least) exponentially, per iteration. In Sections V and VI, we apply SGFA to compute the graph Fourier basis F and the GFT for the 2004 US political blogs graph [2] and for the Manhattan Road Map [3], [4] and carry out a comprehensive empirical study to illustrate accuracy, stability, and tradeoffs of the SGFA. The random models of Fig. 1 are studied in Section VII. Section VIII displays examples of low, medium, and high frequency of the graph spectral components of the Manhattan road map as computed by SGFA presented in the paper. Finally, Section IX concludes the paper.

II. Graph Signal Processing

Graph Signal Processing (GSP) [8], [12], [27], [28], see also overview [29], considers the problem of analyzing and processing signals indexed by the nodes of a graph $\mathcal{G} = (\mathcal{V}, A)$. We will follow [8], [27], [28] that applies to general graphs $\mathcal{G}$, directed or undirected, while [12] considers undirected graphs. The set of nodes is $\mathcal{V} = \{v_0, \ldots, v_{n-1}\}$ and the $n \times n$ graph weighted adjacency matrix is A. Entry (i, j) of A represents a directed edge between nodes v_i and v_j with weight $A_{i,j}$. The graph signal s is a complex valued vector $s = [s_0 \cdots s_{n-1}]^T \in \mathcal{C}^n$ such that the ith-component s_i of s corresponds to the element of s indexed by node i in graph $\mathcal{G}$, i.e., s maps nodes in graph $\mathcal{G}$ to complex numbers in $\mathcal{C}$. The application at hand dictates the structure of $\mathcal{G}$, as highlighted in the three graphs of Fig. 2. In Fig. 2(a), $\mathcal{G}$ represents the graph associated with an image where each node corresponds to a pixel, and, for a common image model, each pixel value (pixel color or intensity) is related to the values of its four adjacent pixels. This relation for the graph in Fig. 2(a) is symmetric, hence all edges are undirected and have the same weight, with possible exceptions of boundary nodes that may have directed edges and/or different edge weights, depending on boundary conditions [15]. Fig. 2(b) [2] represents a directed network, where each node corresponds to a political blog and the edges represent hyperlinks between blogs. The colors of the nodes in Fig. 2(b) are the graph signal and represent the political orientation of the blog (red for conservative, blue for liberal). Orange edges go from liberal to conservative, and purple ones from conservative to liberal; blue and red edges hyperlink blogs of the same color. The size of each blog reflects the number of other blogs that link to it, as according to [2]. Fig. 2(c) models the Manhattan Road Map where each node (in black) corresponds to a two dimensional coordinate location (latitude, longitude) [4] and the directed edges (in blue) represent one- or two-way streets between locations, as verified by Google Maps [3]. In GSP [8], [27], [28], the adjacency matrix A plays the role of the shift operator z^{-1} in DSP. Its eigendecomposition is

$$A = F\Lambda F^{-1}, \tag{1}$$

assuming A is not defective. We let

$$F = [f_0 \cdots f_{n-1}] \tag{2}$$

$$\Lambda = \operatorname{diag}[\lambda_0, \ldots, \lambda_{n-1}] \tag{3}$$

$$F^{-1} = W^H = \begin{bmatrix} w_0^H \\ \vdots \\ w_{n-1}^H \end{bmatrix}. \tag{4}$$

From (1), (2), and (3), we obtain

$$AF = F\Lambda$$

$$Af_i = \lambda_i f_i, \;\; i = 0, \ldots, n-1.$$

Likewise, from (3), (1), and (4), we get

$$F^{-1}A = \Lambda F^{-1} \;\; \left(\text{or } W^H A = \Lambda W^H\right)$$

$$w_j^H A = \lambda_j w_j^H, \;\; j = 0, \ldots, n-1.$$

Matrix Λ is the diagonal matrix of the eigenvalues $\lambda_0, \cdots, \lambda_{n-1}$ of A, F is the matrix of (right) eigenvectors $f_0, \ldots, f_{n-1}$ of A, and $F^{-1} = W^H$ is the matrix of (left) eigenvectors[9] $w_0, \ldots, w_{n-1}$ of A.

The eigenvectors $f_0, \ldots, f_{n-1}$ (columns of F) are the graph frequencies or graph spectral components (corresponding to the harmonic components of time signals), and we will refer to them as the Fourier basis, and the eigenvalues $\lambda_0, \cdots, \lambda_{n-1}$ are the graph frequencies.[10] The matrix $F^{-1} = W^H$ is the graph Fourier transform (GFT).

For a graph signal s, its graph Fourier transform $\widehat{s}$ is

$$\begin{aligned} \widehat{s} &= F^{-1} \cdot s = W^H \cdot s \\ &= \left[w_0^H s \cdots w_{n-1}^H s\right]^T \\ &= \left[\langle w_0, s\rangle \cdots \langle w_{n-1}, s\rangle\right]^T, \end{aligned} \tag{5}$$

where $\langle w_j, s\rangle$ is the complex inner product of the left eigenvectors w_j and s. Equation (5) analyzes the graph signal s in terms of its graph Fourier coefficients $\widehat{s}_0, \cdots, \widehat{s}_{n-1}$, the entries of $\widehat{s}$.

Similarly, the graph signal s is obtained from its graph Fourier transform $\widehat{s}$ through the inverse GFT, F,

$$s = F \cdot \widehat{s} \tag{6}$$

$$= \widehat{s}_0 f_0 + \cdots + \widehat{s}_{n-1} f_{n-1}. \tag{7}$$

Equation (6) shows that $\widehat{s}$ is the representation of the graph signal s on the graph Fourier basis F, while (7) synthesizes the graph signal s as a linear combination of the graph Fourier components $f_0, \ldots, f_{n-1}$, with the coefficients of this linear combination being the graph spectral coefficients $\widehat{s}_j$ of s.

If we take the Hermitian of both sides of Equation (1), we obtain the eigendecomposition of the Hermitian of A

$$\begin{aligned} A^H &= F^{-H}\Lambda^\star F^H \\ &= W\Lambda^\star F^H, \end{aligned} \tag{8}$$

[9]By convention, the left eigenvectors of A are the columns of the matrix W and not the rows of W^H.

[10]With time signals, say continuous time, it is common to call frequencies $f = \frac{1}{j2\pi}\ln\lambda = \frac{1}{j2\pi}\ln e^{j2\pi f}$. Here, the graph frequencies are the eigenvalues λ themselves.

where Λ^* is the conjugate of the matrix Λ of the eigenvalues of A. From (8), it follows

$$A^H W = W\Lambda^\star.$$

This shows that the right eigenvectors of A^H are the (left) eigenvectors of A (written as column vectors). If A is real valued (which we will assume in the sequel, unless otherwise stated), $A^H = A^T$.

Graph signal processing has experienced significant research activity in the last few years, a very incomplete list of references or topics include on sampling [18], [30]–[34], on extending concepts of randomness and stationarity [19], [35], [36], on recovering the underlying graphs from data [37]–[41], on extensions to multirate graph signal processing and wavelets [42], [43], on denoising [44], [45] and other signal reconstruction problems [46], as well as many others. Not much work however has been pursued on computing the graph Fourier transform and related topics for actual real world networks, the focus of this paper.

III. Problem Formulation

The paper considers an algorithm to compute an accurate and stable diagonalization (1) of A. This is important when either the matrix F of eigenvectors is theoretically not full rank (A is defective), see footnote 7, or it is numerically unstable (very small minimum singular value). Although the diagonal structure of (1) might not be achieved, it can be approximated to arbitrary precision as stated next.

Theorem 1 ([47], Theorem 2.4.7.2): Any n-dimensional square matrix A is similar to an upper triangular matrix

$$\forall A,\ \exists\ F(\epsilon), T(\epsilon):\ \ A = F(\epsilon)T(\epsilon)F(\epsilon)^{-1},$$

where $\epsilon > 0$ denotes arbitrary precision, the eigenvalues of A are on the main diagonal of $T(\epsilon)$, and $T(\epsilon)$ is upper triangular

$$T_{i,i}(\epsilon) = \lambda_i(A), \quad |T_{i,j}(\epsilon)| \leq \epsilon \text{ for } j > i. \tag{9}$$

By (9), the triangular matrix $T(\epsilon)$ has infinitesimal small energy on the upper diagonal entries.

The proof of Theorem 1 is constructive—[47] provides closed form expressions for both $F(\epsilon)$ and $T(\epsilon)$ through the complex Schur decomposition of A. Although their matrix $F(\epsilon)$ is provably invertible, a simple argument shows that the minimum singular value $\sigma_{\min}$ of $F(\epsilon)$ is upper bounded by ϵ^{n-1}; so, in practice, it will not be numerically stable for a sufficiently low ϵ or large enough n. In words, they trade the diagonalization degree of T with the numerical stability of F. To prove this, we import their construction of $T(\epsilon)$ and $F(\epsilon)$ for an $\epsilon < 1$. It suffices to analyse $\sigma_{\min}(F(\epsilon))$. Without loss of generality, we need only consider the particular instance $\epsilon < 1$, since, for any scalar $\phi \geq \epsilon$, we get that if $|T_{i,j}| \leq \epsilon$ then $|T_{i,j}| \leq \epsilon \leq \phi$. Matrices $T(\epsilon)$ and $F(\epsilon)$ are [47]

$$(U,T):\ \ UU^H = I,\ \ AU = TU,\ \ T_{i,j} = 0 \text{ for } j < i \tag{10}$$

$$t := \max_{j>i}\ |T_{i,j}|$$

$$D_\theta := \begin{bmatrix} 1 & & & \\ & \theta & & \\ & & \ddots & \\ & & & \theta^{n-1} \end{bmatrix}$$

$$(F,T)(\epsilon) = \begin{cases} \left(UD_\epsilon,\ \ D_\epsilon^{-1}TD_\epsilon\right) & \text{if } t \leq 1 \\ \left(UD_{1/t}D_\epsilon,\ \ D_\epsilon^{-1}D_{1/t}^{-1}TD_{1/t}D_\epsilon\right) & \text{if } t > 1, \end{cases} \tag{11}$$

where U is the orthogonal matrix of Schur vectors for A. As stated in Theorem 1, the matrix Λ of eigenvalues of A is the matrix of diagonal entries of the triangular matrix $T(\epsilon)$.

Let $t > 1$ in (10); the alternative is similarly addressed.

$$\begin{aligned} \sigma_{\min}^2\left(F(\epsilon)\right) &= \sigma_{\min}^2\left(UD_{1/t}D_\epsilon\right) \\ &= \min_{\|x\|=1} \left\|UD_{1/t}D_\epsilon x\right\|_2^2 \\ &\leq \|U\|_2^2 \min_{\|x\|=1} \left\|D_{1/t}D_\epsilon x\right\|_2^2 \\ &\leq \left\|D_{1/t}\right\|_2^2 \min_{\|x\|=1} \|D_\epsilon x\|^2 \\ &\leq \sigma_{\min}^2\left(D_\epsilon\right) \\ &= \epsilon^{2(n-1)}. \end{aligned} \tag{12}$$

The last equation shows as desired that $\sigma_{\min}(F(\epsilon)) \leq \epsilon^{(n-1)}$. Inspired by this result, we propose to approximate the diagonal structure of A in (1), while maintaining the numerical stability of the Fourier Basis as measured by the minimum singular value $\sigma_{\min}(F)$, by solving the following problem

$$\begin{aligned} \underset{F,T\in\mathbb{C}^{n\times n}}{\text{minimize}} \quad & \|AF - F\Lambda\|_{\mathcal{F}} \\ \text{subject to} \quad & AF = FT, \quad \sigma_{\min}(F) \geq \alpha, \\ & T_{i,j} = 0 \quad i > j, \quad T_{i,i} = \lambda_i, \end{aligned} \tag{13}$$

where $\|.\|_{\mathcal{F}}$ denotes the Frobenius norm, $\alpha \leq 1$ is an arbitrary strictly positive constant, and Λ is obtained from the diagonal entries of T, see (9). To impose the diagonal structure of (1), we can choose other measures of accuracy like $\sqrt{\sum_{j>i}|T_{i,j}|^2} = \|T-\Lambda\|_{\mathcal{F}}$ instead of $\|F(T-\Lambda)\|_{\mathcal{F}}$. The algorithm of Section IV is invariant to both formulations, and we can show that the same type of theoretical reasoning applies (Result 1 and Theorem 2).

For generic stability margin α, problem (13) is challenging since 1) we have a bilinear interaction between matrices (F,T); and 2) $\sigma_{\min}$ is a non concave function.

Remark 2: Even though by Theorem 1 any matrix is arbitrarily close to being similar to a diagonal matrix, the solution of the problem (13) is not simply to rescale the triangular T in Theorem 1 till $\sigma_{\min}(F) \geq \alpha$. In fact, the objective in problem (13) is not invariant to scaling. If we scale F, we obtain a feasible solution, but the objective function will become worst. In fact, assume we find a pair (F,T) such that $\sigma_{\min}(F) > 0$ and all constraints of problem (13) hold, but $\sigma_{\min}(F) \ngeq \alpha$. In this case, if, to solve problem (13), we simply scaled F by $\gamma/\sigma_{\min}(F)$, even if $\gamma F/\sigma_{\min}(F)$ is still feasible for any $\gamma \geq \alpha$, the objective

value is also scaled by $\gamma/\sigma_{\min}(F) > 1$ and so it gets worse. In summary, the solution is not simply to find an F that satisfies all constraints except $\sigma_{\min}(F) \geq \alpha$ and then rescale this F.

IV. Stable Accurate Approximation to the GFT

In this section, we present a local solution to problem (13), the stable graph Fourier algorithm (SGFA).

A. Stable Graph Fourier Algorithm (SGFA)

Although non-convex, problem (13) can be globally solved for a small α, as we describe in the next result.

Result 1: For any $\epsilon > 0$, if $\alpha > 0$ is sufficiently small, then problem (13) can be optimally solved up to ϵ, by using in (11)

$$T\left(\frac{\epsilon}{\|F\|_{\mathcal{F}}\sqrt{0.5n(n-1)}}\right) \text{ and } F\left(\frac{\epsilon}{\|F\|_{\mathcal{F}}\sqrt{0.5n(n-1)}}\right).$$

Proof: Result 1 follows since the objective in problem (13) is upper bounded by the off diagonal energy of matrix T. To be concrete, we show that the objective in problem (13) is upper bounded by ϵ. To see this, take $T(\phi)$ and $F(\phi)$ from (11), and for $\phi = \frac{\epsilon}{\|F\|_{\mathcal{F}}\sqrt{0.5n(n-1)}}$, we get successively

$$\begin{aligned}\|AF - F\Lambda\|_{\mathcal{F}} &= \|F(T-\Lambda)\|_{\mathcal{F}} \\ &\leq \|F\|_{\mathcal{F}}\|T-\Lambda\|_{\mathcal{F}} \\ &\leq \|F\|_{\mathcal{F}}\sqrt{0.5n(n-1)\phi^2} \\ &= \epsilon.\end{aligned}$$

The third inequality follows from the second because the $.5n(n-1)$ upper elements of the triangular matrix T are bounded by ϕ and from the assumed expression for ϕ. The proof would be done if ϵ was a free variable. Note however that, by (12), there exists a lower bound on ϵ namely $\epsilon \geq \alpha^{1/(n-1)}||F||_{\mathcal{F}}\sqrt{0.5n(n-1)}$. Since α is assumed arbitrarily small, the bound $\alpha^{1/(n-1)}||F||_{\mathcal{F}}\sqrt{0.5n(n-1)}$ can be made arbitrarily small and the optimality result follows. ■

Our heuristic for problem (13) is based on Result 1: if we find a point (F,T) such that the off diagonal energy of T is arbitrarily small and matrix F is numerically stable, then (F,T) is (close to being) globally optimal for problem (13). We then propose the next simple iterative scheme. Start with a feasible initial point (F_0,T_0) and proceed by updating both variables as follows: 1) contract the upper diagonal energy of T; and 2) compute a Fourier Basis F, compliant with T, that has maximal $\sigma_{\min}$. Contraction is obtained by simply multiplying the off diagonals elements of T_k by a factor $\beta < 1$.

Maximizing the minimum singular value of an arbitrary matrix is challenging, since its $\sigma_{\min}$ is a non-concave function. Instead, we consider the following general concave bound [47]

$$\forall\,(F_k,F_{k+1}){:}\,\sigma_{\min}(F_{k+1}) \geq \sigma_{\min}(F_k) - \|F_{k+1}-F_k\|_{\mathcal{F}}, \quad (14)$$

where F_k is the (constant matrix of) Fourier Basis from iteration k. Hence, the Fourier Basis F_k is updated as follows

$$\begin{aligned} F_{k+1} \in \ & \arg\max_{F} && \sigma_{\min}(F_k) - \|F_k - F\|_{\mathcal{F}} \\ & \text{subject to} && AF = FT_{k+1}, \end{aligned}$$

which is equivalent to

$$\begin{aligned} F_{k+1} \in \ & \arg\min_{F} && \|F_k - F\|_{\mathcal{F}} \\ & \text{subject to} && AF = FT_{k+1}. \end{aligned} \quad (15)$$

Remark that problem (15) is always feasible and, hence, the iterative scheme is well posed for any iteration number k. Regardless of matrix A, one possible starting point (F_0,T_0) comes from the complex Schur decomposition of A:

$$A = F_0T_0F_0^H, \ \ F_0^HF_0 = I_n, \ \ \{T_0\}_{ij} = 0 \ \ i > j. \quad (16)$$

The pair (F_0,T_0) is feasible for problem (13), since $\sigma_{\min}(F_0) = 1 \geq \alpha$ and the diagonal elements of T_0 correspond to eigenvalues of A. SGFA is in Algorithm 1 where the numerical tolerance α defines the stopping criteria.

Algorithm 1: Stable Graph Fourier Algorithm-SGFA.

1: Input Parameters: contraction factor $0 < \beta < 1$ and stability tolerance $0 < \alpha \leq 1$.
2: Compute the complex triangular decomposition of matrix A, i.e., compute F_0, T_0 such that (16) holds.
3: Update T_k by contracting the upper diagonal elements:

$$\{T_{k+1}\}_{i,j} = \begin{cases} \beta\,\{T_k\}_{i,j}\,, & \text{for } j > i \\ \{T_k\}_{i,j} & \text{otherwise} \end{cases}.$$

4: Update F_k by solving optimization problem:

$$\begin{aligned} F_{k+1} \in \ & \arg\min_{F\in\mathbb{C}^{n\times n}} && \|F - F_k\|_{\mathcal{F}} \\ & \text{subject to} && AF = FT_{k+1}, \end{aligned} \quad (17)$$

through an iterative solver.
5: Update (T_k,F_k) until $\sigma_{\min}(F_k) < \alpha$.

The next theorem proves that SGFA exponentially decreases the objective of problem (13), by exploring the orthogonal projection nature of the update F_{k+1}.

Theorem 2: If SGFA runs for N iterations, the objective of problem (13) decays, at least, exponentially fast in N,

$$\|AF_N - F_N\Lambda\|_{\mathcal{F}} \leq \beta^N\,\|T_0-\Lambda\|_{\mathcal{F}}\,\|F_0\|_{\mathcal{F}}\,. \quad (18)$$

Proof: Bound (18) follows from $\|F_{k+1}\|_{\mathcal{F}} \leq \|F_k\|_{\mathcal{F}}$. To show this, note that F_{k+1} corresponds to the orthogonal projection of F_k on the linear subspace Φ_{k+1} defined by

$$\Phi_{k+1} := \{F : AF = FT_{k+1}\}\,.$$

Hence, F_k can be uniquely decomposed as

$$F_k = F_{k+1} + W_{k+1}$$

where $F_{k+1} \in \Phi_{k+1}$ and W_{k+1} belongs to the orthogonal complement of Φ_{k+1}, i.e., $W_{k+1} \in \Phi_{k+1}^{\perp}$. By the orthogonality between W_{k+1} and F_{k+1}, we can conclude that

$$\begin{aligned} 0 &\leq \|F_{k+1}-F_k\|_{\mathcal{F}}^2 \\ &= \|F_{k+1}\|_{\mathcal{F}}^2 + \|F_k\|_{\mathcal{F}}^2 - \text{trace}\left(F_{k+1}^HF_k\right) - \text{trace}\left(F_k^HF_{k+1}\right) \\ &= \|F_{k+1}\|_{\mathcal{F}}^2 + \|F_k\|_{\mathcal{F}}^2 - 2\,\text{trace}\left(F_{k+1}^HF_{k+1}\right) \\ &= \|F_k\|_{\mathcal{F}}^2 - \|F_{k+1}\|_{\mathcal{F}}^2\,. \end{aligned} \quad (19)$$

The first equality follows by expanding the right-hand-side of the first inequality and recalling that the inner product of two matrices is given by the trace, and the third equality follows by recognizing that by orthogonality $\text{trace}(F_{k+1}^H W_{k+1}) = 0$. We now show that the exponential bound (18) is a consequence of (19). By step 4 of the SGFA, Algorithm 1, $AF_k = F_k T_k$. Then

$$\begin{aligned}\|AF_k - F_k\Lambda\|_{\mathcal{F}} &= \|F_k\,(T_k - \Lambda)\|_{\mathcal{F}} \\ &\leq \|T_k - \Lambda\|_{\mathcal{F}}\,\|F_k\|_{\mathcal{F}} \\ &= \beta^k\,\|T_0 - \Lambda\|_{\mathcal{F}}\,\|F_k\|_{\mathcal{F}} \\ &\leq \beta^k\,\|T_0 - \Lambda\|_{\mathcal{F}}\,\|F_0\|_{\mathcal{F}}\,. \end{aligned} \tag{20}$$

The last inequality follows from step 3 of the SGFA, algorithm 1, and (20). ■

Example 1 (Algorithm 1: Tradeoffs): We illustrate with an analytical example the tradeoffs between the algorithm parameters β and α, and the error $\|AF_k - F_k\Lambda\|_{\mathcal{F}}$ as a function of the iteration number k. Consider the nilpotent graph shift

$$A = \begin{bmatrix} 0 & 1 & & \\ & 0 & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{bmatrix} \in \mathbf{R}^{n\times n}$$

that has eigenvalue zero with algebraic multiplicity n and geometric multiplicity 1. This graph shift is a $n \times n$ Jordan block associated with $\lambda = 0$ and, so, it is not diagonalizable. We apply algorithm 1 and derive a closed form expression for the approximate Fourier basis F_k of A. Since A is upper triangular, the initialization step of algorithm 1 is $A\,F_0 = T_0\,F_0$ with $T_0 = A$, $F_0 = I_n$. The iterates $\{T_k, F_k\}_{k=1}^{+\infty}$ are

$$T_k = \begin{bmatrix} 0 & \beta^k & & \\ & 0 & \ddots & \\ & & \ddots & \beta^k \\ & & & 0 \end{bmatrix}$$

$$F_k = \arg\min_F \{\,\|F - F_{k-1}\|_{\mathcal{F}},\;\; A\,F = F\,T_k\} \tag{21}$$

$$= \gamma_k\,\text{diag}\left[1, \beta^k, \ldots, \beta^{k(n-1)}\right] \tag{22}$$

$$\gamma_k :\, = \prod_{i=1}^{k} \frac{1 + \beta^{2i-1} + \cdots + \beta^{2i\,(n-1)-(n-1)}}{1 + \beta^{2i} + \cdots + \beta^{2i\,(n-1)}} \tag{23}$$

Result (22) follows from the structure of A and the quadratic nature of (21),

$$\begin{aligned} &\underset{F}{\text{minimize}} && \|F - F_{k-1}\|_{\mathcal{F}} \\ &\text{subject to} && A\,F = F\,T_k \\ &&\Leftrightarrow& \\ &\underset{F}{\text{minimize}} && \sum_{i,j=1}^{n} \left(F_{i,j} - \{F_{k-1}\}_{i,j}\right)^2 \\ &\text{subject to} && \{F_{i,j}\}_{i,j=2,\ldots,n} = \beta^k\;\{F_{i,j}\}_{i,j=1,\ldots,n-1} \end{aligned} \tag{24}$$

Since F_0 is diagonal, it follows that F_k will remain diagonal because in problem (24) the objective is minimized by selecting $F_{i,j} = 0$ for $i \neq j$ (which is feasible). Given that $F_{i,i} = \beta^{k\,(i-1)}\,F_{1,1}$ for $i = 2, \ldots, n$, the convex objective (24) only needs to be optimized with respect to the scalar variable $F_{1,1}$. The optimal $F_{1,1}$ given by

$$F_{1,1} = \frac{\{F_{k-1}\}_{1,1} + \beta^k\{F_{k-1}\}_{2,2} + \cdots + \beta^{k(n-1)}\{F_{k-1}\}_{n,n}}{1 + \beta^{2k} + \cdots + \beta^{2k(n-1)}}. \tag{25}$$

Expression (22) follows by setting $F = F_k$ in (25) and unrolling the resulting recursion. The factor γ_k in (23) will converge to an $\ell \geq 1$ since $\beta < 1$. To show this, first note that

$$\gamma_k \leq \prod_{i=1}^{k}\{1 + \beta^{2i-1} + \cdots + \beta^{2i\,(n-1)-(n-1)}\} := \widehat{\gamma}_k$$

since the denominator of (23) is upper bounded by one. To show that γ_k converges, we show instead that $\widehat{\gamma}_k$ converges. Note

$$\log\{\widehat{\gamma}_k\} = \sum_{i=1}^{k} \log\{1 + h_i\}$$

with $h_i := \beta^{2i-1} + \cdots + \beta^{2i\,(n-1)-(n-1)}$. Let us compare the two series $\sum_{i=1}^{k} \log\{1 + h_i\}$ and $\sum_{i=1}^{k} h_i$ by the limit comparison test,

$$\lim_{i\to+\infty} \frac{\log\{1 + h_i\}}{h_i} = 1,$$

since $h_i \to 0$. Hence $\log\{\widehat{\gamma}_k\}$ converges if and only if $\sum_{i=1}^{k} h_i$ converges. This last series trivially converges since we are summing $n - 1$ geometric series with ratio lower than unit. So $\log\{\widehat{\gamma}_k\}$ converges and γ_k converges to a limit ℓ. The limit ℓ is greater than or equal to one, since the k terms being multiplied in (23) are greater than or equal to one. Using (22), one can directly relate the approximation error $\|AF_k - F_k\Lambda\|_{\mathcal{F}}$ and the minimum singular value $\sigma_{\min}(F_k)$,

$$\|AF_k - F_k\Lambda\|_{\mathcal{F}}^2 \,= \gamma_k^2\{\beta^{2k} + \cdots + \beta^{2k(n-1)}\} \tag{26}$$

$$\sigma_{\min}^2\,(F_k) \,= \gamma_k^2\,\beta^{2k(n-1)}. \tag{27}$$

Equations (26) and (27) show for this simple example important facts: 1) they give explicit expressions for both accuracy $\|AF_k - F_k\Lambda\|_{\mathcal{F}}$ and stability $\sigma_{\min}^2(F_k)$; 2) they show that the singular value $\sigma_{\min}(F_k)$ also decays exponentially fast with parameter β, but at much faster rate than the approximation error $\|AF_k - F_k\Lambda\|_{\mathcal{F}}$ ($\beta^{k(n-1)}$ vs β^k); they show that $\sigma_{\max}(F_k) = 1$, i.e., the condition number $\sigma_{\max}(F_k)/\sigma_{\min}(F_k)$ behaves as $1/\sigma_{\min}(F_k)$. The last two comments will be verified empirically for much broader examples. 3) Because $\gamma_k \to \ell \geq 1$, $\lim_{k\to\infty}\sigma_{\min}(F_k) = \lim_{k\to\infty}\beta^{k(n-1)}\,\gamma_k = 0$, showing that F_k asymptotically becomes rank deficient. they impose a fundamental limit on the quality of our approximation, i.e., for any fixed stopping criteria α the approximation error $\|AF_k - F_k\Lambda\|_{\mathcal{F}}$ cannot be made arbitrarily small. Indeed, for $\|AF_k - F_k\Lambda\|_{\mathcal{F}}$ to be arbitrarily small one would need to a) take infinitely many iterations k, or b) choose an infinitesimal small contraction factor β. In either case, the minimum singular

value $\sigma_{\min}(F_k)$ would also decay to zero, and we would get $\sigma_{\min}(F_k) < \alpha$ after a) sufficiently many iterations k, or b) in the first iteration $k = 1$ when β is sufficiently small. So, this simple example shows that SGFA comes with a trade-off: in general, it is impossible to find a Fourier basis that is both arbitrarily stable and arbitrarily accurate. One must trade the two metrics to get an appropriate approximation that meets a stability criteria defined by α. □

Sections V–VII will confirm empirically the observation in example 1 that there exists a tradeoff between contraction factor β and the minimum singular value $\sigma_{\min}(F_k)$. The sections verify empirically that, when β is small, a large contraction is performed in T_k and $\sigma_{\min}(F_k)$ tends to decrease rapidly. This decay in $\sigma_{\min}(F_k)$ is approximately exponential for low values of β, replicating the behavior observed in the previous example.

B. Updating the Fourier Basis

To update the Fourier Basis F_k and solve problem (17), we consider the (iterative) LSQR solver implemented in MATLAB [48]. The method is based on the Golub-Kahan bidiagonalization process. It is algebraically equivalent to the standard method of conjugate gradient, but it has better numerical properties.

To use LSQR, we first reformulate problem (17) and step 4 of algorithm 1 as follows,

$$\begin{aligned} \underset{f\in\mathbb{C}^{n^2}}{\text{minimize}} \quad & \|f\|_2 \\ \text{subject to} \quad & \mathcal{A}f = b, \end{aligned} \tag{28}$$

where

$$\mathcal{A} = I_n \otimes A - T_{k+1}^T \otimes I_n, \;\; b = \text{vec}\left(F_k T_{k+1} - A F_k\right).$$

The symbol $\otimes$ denotes the Kronecker product. Formulation (28) simply interprets matrix F as a vector $f = \text{vec}(F)$ in $\mathbb{C}^{n^2}$, rewrites the underlying matrix equality in vector form and shifts the solution by $\text{vec}(F_k)$ to get the desired format for the MATLAB solver [48].

LSQR has no need to store in memory matrix $\mathcal{A} \in \mathbb{C}^{n^2\times n^2}$, by defining a proxy to efficiently compute $\mathcal{A}f$ and $\mathcal{A}^H f$. In our context, this is possible since

$$\begin{aligned} \mathcal{A}f &= \text{vec}\left(AF - FT_{k+1}\right) \\ \mathcal{A}^H f &= \text{vec}\left(A^T F - F T_{k+1}^H\right). \end{aligned} \tag{29}$$

We now consider the computational effort of SGFA and algorithm 3. We start by estimating this effort for step 4 of algorithm 1 using solver LSQR and as refomulated by (28) and (29). To compute $\mathcal{A}f$ and $\mathcal{A}^H f$, we just store, multiply, and sum $n \times n$ matrices. To solve (28), by [48], LSQR will perform $10\,n^2$ multiplications plus computing $\mathcal{A}f$ and $\mathcal{A}^H f$ per iteration. By (29), computing $\mathcal{A}f$ and $\mathcal{A}^H f$ involves computing AF and FT_{k+1} (or $A^T F$ and FT_{k+1}^H). We recall that in practice A is highly sparse, F is dense, and T_{k+1} will vary from an upper triangular to essentially a diagonal matrix (whose diagonal entries are fixed being equal to the eigenvalues of A). So, computing FT_{k+1} and FT_{k+1}^H will require from $O(n^3)$ (when T_{k+1} is triangular) to $O(n^2)$ (when T_{k+1} is diagonal). We now consider the computational effort of calculating AF and $A^T F$.

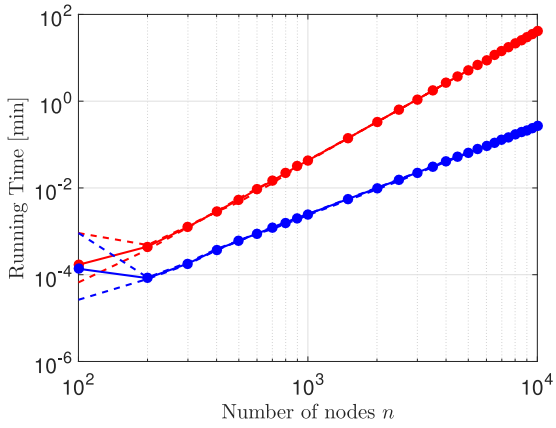


Fig. 3. Running time of a single iteration of LSQR for (28): mean (dots) and minimum/maximum (dotted lines) times for 10 random initializations. Matrix T_{k+1} is upper triangular for the red plot and diagonal for the blue plot.

In practice, graphs are highly sparse with node average degree[11] $d \ll n$. Assuming that the average number of nonzero entries of each row of A is $d \approx \ln n \ll n$, then the product $\mathcal{A}\,f$ (or $\mathcal{A}^H\,f$) has computational complexity $O(dn^2) \sim O(n^2 \ln n)$, since matrix F is typically dense. Therefore, the overall computational cost of one iteration of LSQR varies from $O(n^3)$ to $O(n^2 \ln n)$.

To have a better estimate of the computational requirements of LSQR, we plot in Fig. 3 the running time of a single iteration of LSQR (single iteration of step 4 of algorithm 1) when graph shift A is a directed Erdős-Rényi graph with probability of connection[12] $p = \frac{1}{n}$, and matrices (F_k) are complex matrices with real and complex parts sampled from an i.i.d. standard Gaussian distribution and matrices (T_k) are either triangular i.i.d. standard Gaussian (red or top curve of Fig. 3) or diagonal (blue or bottom curve of Fig. 3). We plot these two curves since, as explained previously, matrix T_{k+1} will vary from an upper triangular to an essentially diagonal matrix as the iteration number k increases. Hence, the curves of Fig. 3 represent bounds on the expected computational time of a single iteration of LSQR (single iteration of step 4 of algorithm 1): at the beginning (initial iterations) the red (top) curve tracks the time it takes for an LSQR iteration, while for later iterations, as LSQR becomes faster, it is the blue (bottom) curve that tracks this time.

We now consider the time it takes one iteration of SGFA—step 3 (contracting T_{k+1}), multiple iterations of step 4 (LSQR), say K, and step 5 of algorithm 1. We empirically verified that the complexity of an iteration of SGFA is essentially dominated by the LSQR steps. Then, the overall computational complexity of a single iteration of SGFA,[13] algorithm 1, varies from

[11] For example, for undirected Erdős-Rényi graphs with probability of connection p, the average degree is pn. With sharp threshold of connectedness $\frac{\ln n}{n}$, the average degree of a sparse connected Erdős-Rényi graph is of the order $\ln n$. Recent work [50] shows that $\frac{\ln n}{n}$ is also a threshold for directed Erdős-Rényi graphs, now with probability p of being strongly connected.

[12] By using MATLAB `eig` function, we always verify that $\sigma_{\min}(F) < 10^{-12}$, where F denotes the eigenvector matrix of the graph shift A.

[13] In algorithm 1 the complexity of contracting upper triangular matrix T_k by β – step 3 – is not considered since it is dominated by the other updates – steps 4 and 5.

$O(Kn^2 \ln n + n^3)$ to $O((K+1)\, n^3)$. The extra n^3 term comes from the computation of the stopping criterion of SGFA, step 5 of algorithm 1, that requires computation of the singular value of F_k and will be ignored in the sequel.[14]

We use Fig. 3 to estimate the running time of SGFA. For example, for a graph with $n = 1500$ nodes, Fig. 3 shows that a single iteration of LSQR takes from .0055 min to .14 min, while for a much larger graph, $n = 10\,000$, it takes from .27 min to 41.56 min. If we assume that we run $K = 100$ iterations of LSQR (inner loop) to solve optimization (17), we conclude that one iteration of SGFA (with $K = 100$ iterations of LSQR) takes for the graph with $n = 1500$ nodes on average from 5 min to 14 min, while for the larger graph with $n = 10\,000$ nodes it takes from 27 min to roughly 70 h. All numerical experiments were carried out on a personal computer with processor Intel Core i7-2600, CPU 3.4 GHz, and 16 GB of RAM.[15]

V. Political Blogs

We now study empirically the SGFA, Algorithm 1, by applying it to the shift matrix for the graph of the political blogs network.

A. Dataset

We consider the network of Fig. 2(b)—(a) graph of hyperlinks between $n = 1490$ weblogs, over the period of two months preceding the U.S. Presidential Election of 2004 [2]. The underlying adjacency matrix A is directed and weighted: if node v_i has k URL references to node v_j then $A_{i,j} = k$. The maximum number of URL references between two nodes is two. Matrix A is made publicly available by Mark Newman,[16] together with some additional general information: $\sigma_{\min}(A) = 0$ (default precision), there are 19025 non zero entries, and $\text{rank}(A) = 784 < 1490$; in this case $\lambda = 0$ is a repeated eigenvalue. By computing F with MATLAB `eig`, we find that $\sigma_{\min}(F) \approx 1.1 \times 10^{-33}$. We now apply SGFA, algorithm 1, to find an approximation to the diagonalization of A.

B. Approximating a Stable Graph Fourier Transform

In this section, we evaluate the performance of SGFA, algorithm 1, for different values of the threshold α for $\sigma_{\min}(F_k)$ and contracting factor β, namely, values of (α, β) on a two dimensional 6×14 grid. As explained in Section III, the objective is to get an accurate approximation of the Fourier Basis F (inverse GFT), while maintaining the numerical stability of F. We discuss accuracy and stability.

1) Accuracy: Fig. 4 plots the objective of problem (13), $\|AF - F\Lambda\|_{\mathcal{F}}$, for the range of values of (α, β) indicated on the horizontal and vertical axes of the figure, respectively. The vertical color code bar on the right of the figure gives the values of $\|AF - F\Lambda\|_{\mathcal{F}}$. For the brown-red blocks on the upper left corner of the image ($.01 \le \beta \le .32$ and $\alpha \le 5 \times 10^{-4}$), SGFA, algorithm 1, only ran for a single iteration, since after the first iteration the stopping condition $\sigma_{\min}(F_1) < \alpha$ is already met; hence, the best estimate of the Fourier Basis is the starting F_0 obtained by performing the complex Schur decomposition of A in (16). For any value of the contracting factor β, when the threshold α decreases, the accuracy of the approximation improves as can be observed from the gradient of the colors in each row of Fig. 4. This is expected since, if the threshold α decreases, our stability criterion is relaxed and SGFA, algorithm 1, runs for more iterations. SGFA may stop before noticeable reduction of the objective. For example, for $\beta = .01$ and $\alpha \ge .001$ the algorithm stops with the approximation error still significant, see top left corner of the figure that gives $\|FA - \Lambda F\|_{\mathcal{F}} \ge 100$. To decrease the error objective to for example $\|FA - \Lambda F\|_{\mathcal{F}} \le 20$, we can increase $\beta \ge .43$ with $\alpha \ge .01$. To reduce even more significantly the objective $\|FA - \Lambda F\|_{\mathcal{F}} \le 10$, we can keep $\beta \ge .43$ but reduce $\alpha \approx 10^{-5} - 10^{-6}$.

Theorem 2 proves that there is an exponential dependency between accuracy and the number of iterations of SGFA, algorithm 1. We consider this in Fig. 5 that plots in log scale

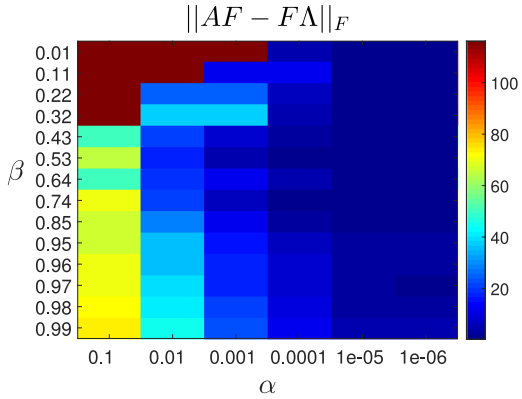


Fig. 4. Political blogs network: Accuracy of SGFA, algorithm 1, 2-D map as a function of algorithm parameters (α, β).

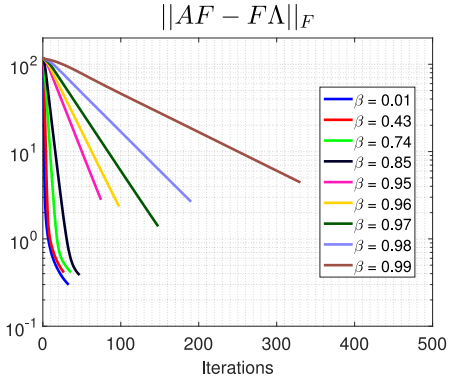


Fig. 5. Political blogs network: Log scale accuracy of SGFA, algorithm 1, with threshold kept fixed, $\alpha = 10^{-6}$.

[14] Singular values are usually computed by LAPACK [50] that 1) reduces the square matrix A to a bi-diagonal form B, and 2) finds the singular values of B, which are equal to those of A. Step 1 uses Householder reductions with complexity $O(n^3)$, while several alternatives exist for 2 with cost $O(n^2)$. We ignore this time in the discussion.

[15] Such large graphs will be handled by more powerful computational resources. Also, the number of iterations of LSQR can be adjusted to improve the scalability of algorithm 1.

[16] Visited March 2019. [Online]. Available: https://www.cise.ufl.edu/research/sparse/matrices/Newman/polblogs

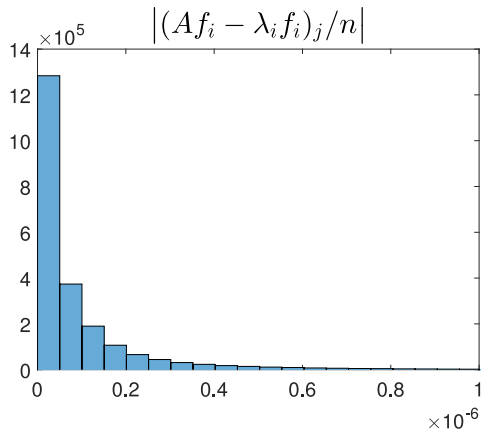


Fig. 6. Political blogs network—Histogram of component wise errors: Other measure of the accuracy of SGFA, algorithm 1, with $(\alpha, \beta) = (10^{-6}, 0.74)$. The tail of the histogram was removed for visualization purposes. From the 1490^2 data points, only 0.006% had an error larger then 10^{-6}, with maximum error 5.03×10^{-6}.

$\|AF - F\Lambda\|_{\mathcal{F}}$ versus the number of iterations for the indicated values of the contracting factor β, while maintaining fixed the threshold at $\alpha = 10^{-6}$. Note the linear decay tendency of the several plots in Fig. 5. It is interesting to note that for $\alpha \leq 10^{-5}$ the approximation error tends to be less sensitive to moderate values of $\beta \leq 0.85$, as seen from the corresponding plots of Fig. 5 (the four left lines close to the vertical axis) that all terminate at a similar endpoint $\|AF - F\Lambda\|_{\mathcal{F}} \approx 0.4$. This can also be concluded by observing that there are no visible significant color differences on the right of Fig. 4.

While Fig. 5 gives the sum of the errors over all $1490^2 \approx 2.25 \times 10^6$ entries, we consider now the accuracy over each normalized component of the eigenvector error given by

$$\left| \frac{(Af_i - \lambda_i f_i)_j}{n} \right|, \quad (i, j) \in \{1, \ldots, n\}^2. \tag{30}$$

Fig. 6 plots the histogram of the error (30) after computing F with SGFA, algorithm 1, with $(\alpha, \beta) = (10^{-6}, 0.74)$. As can be seen, the component wise error given by (30) and shown on the horizontal axis of the figure is on the order of 10^{-6} and, in fact, 75% of all $(Af_i - \lambda_i f_i)_j$ have an absolute value lower than 10^{-7} (bars on the left of the figure).

2) Stability: Figs. 7 and 8 plot measures of numerical stability for the approximated Fourier Basis F—the minimum singular value $\sigma_{\min}(F)$, the condition number $\kappa(F)$, and the inverse error $\epsilon_{\text{inv}}(F)$. The condition number $\kappa(.)$ in Fig. 8 is

$$\kappa(F) = \frac{\|F\|_2}{\sigma_{\min}(F)}. \tag{31}$$

If the condition number $\kappa(F)$ is large the underlying matrix is close to being singular, and, hence, (31) provides another classical measure of numerical stability for F.

The inverse error $\epsilon_{\text{inv}}(F)$ shown in Fig. 8(a) is defined by

$$\epsilon_{\text{inv}}(F) = \max\left\{ \left\| FF^{-1} - I \right\|_{\mathcal{F}}, \left\| F^{-1}F - I \right\|_{\mathcal{F}} \right\}. \tag{32}$$

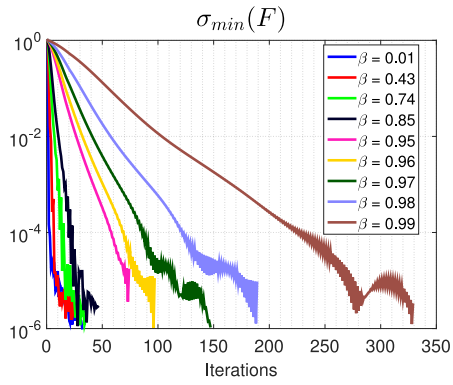


Fig. 7. Political blogs network: Stability of SGFA, algorithm 1, as a function of number of iterations K with stopping criteria $\alpha = 10^{-6}$: $\sigma_{\min}(F)$.

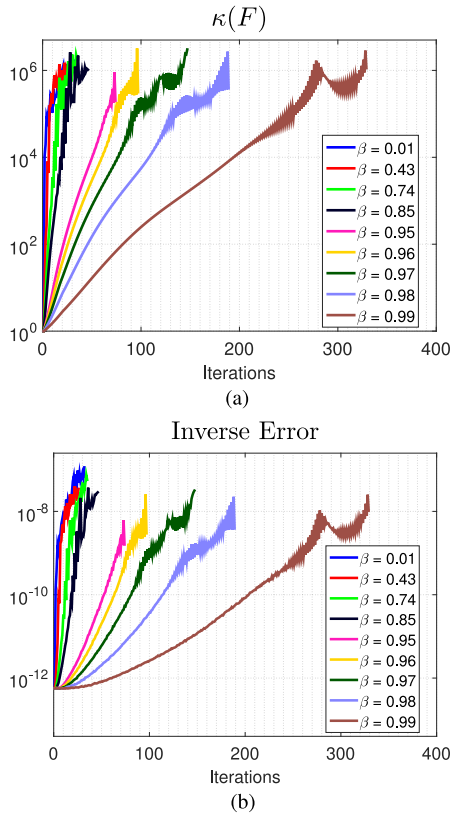


Fig. 8. Political blogs network: Other measures of stability of F as a function of the number of iterations K with the same stopping criteria $\alpha = 10^{-6}$: (a) condition number $\kappa(F)$; (b) inverse error $\epsilon_{\text{inv}}(F)$ that quantifies the error of inverting matrix F.

It measures the quality of the numerically computed inverse F^{-1}. This is important in graph signal processing since, as it is well known [8] and shown by (5), F^{-1} is the GFT.

Contrasting the five lines on the right with the four left lines graphed in the three plots shown in Figs. 7 and 8, we see that the three stability measures of F, $\sigma_{\min}(F)$, $\kappa(F)$, and $\epsilon_{\text{inv}}(F)$ all tend to converge at a much slower rate for $\beta \geq 0.85$ than with low-medium values of β, while still achieving, after a sufficiently large number of iterations, the same overall numerical stability.

In practice, this suggests that one should avoid a high value of β: SGFA, algorithm 1, will take longer to achieve an equally stable but less accurate Fourier basis (as shown in Fig. 5).

Although the stopping criteria of SGFA, algorithm 1, is completely defined by $\sigma_{\min}(F_k) < \alpha$, we observe that in practice all three stability measures tend to scale linearly with the number of iterations K just as the accuracy measure $\|AF - FA\|_{\mathcal{F}}$ scales linearly with K, see Fig. 4. This is more true with small values of β. Further, from Fig. 8(a), we observe that the limiting values of the minimum singular value and of the condition number of F tend to scale linearly with α or $1/\alpha$, respectively, i.e., $\sigma_{\min}(F) \approx \alpha \approx 10^{-6}$ (Fig. 7 or $\kappa(F) \approx 1/\alpha \approx 10^{6}$ (Fig. 8(a))). This suggests that, empirically, one can interpret the constant stopping criteria α either in terms of $\sigma_{\min}$ or the condition number κ, i.e., although SGFA, algorithm 1, was motivated by the convex upper bound of $\sigma_{\min}$ given in Equation (14), Fig. 8(a) suggests that the termination criteria of algorithm 1 could compare κ against $1/\alpha$ instead of $\sigma_{\min}$ against α.

As a final comment, by numerically computing the inverse F^{-1} of the approximate matrix F obtained with SGFA, algorithm 1, Fig. 8(b) shows that the inverse error $\epsilon_{\text{inv}}(F)$ measuring how far FF^{-1} is away from the identity I as given by (32) is in the range $[10^{-7}, 10^{-8}]$ for a tolerance of $\alpha = 10^{-6}$. Since the total number of entries of F is $n^2 = 1,490^2 \approx 2.2 \times 10^6$ this means that the error in any individual entry of FF^{-1} is, on average, on the order of 10^{-13} to 10^{-14}.

3) Approximating Left vs Right Eigenvectors—Political Blogs Network: All previous sections considered the problem of computing accurate and stable approximation of right eigenvectors F for an arbitrary graph shift A. Note, however, that the same exact reasoning applies for (the conjugate transposed of the) left eigenvectors $W^H = F^{-1}$, as defined in Equation (4), since the left eigenvectors of A correspond to the right eigenvectors of A^H, i.e.,

$$AF = F\Lambda \Leftrightarrow F^H A^H = \Lambda^* F^H \Leftrightarrow A^H W = W\Lambda^*. \quad (33)$$

So, one could, equivalently, apply SGFA, algorithm 1, with input A to obtain F and then invert it to compute F^{-1}, or apply SGFA, algorithm 1, directly to A^H and obtain W and then invert W^H to get F, and choose the better approximation, in terms of accuracy and stability. If SGFA, algorithm 1, finds the global solution of (13), Equation (33) implies that we will exactly have $F^{-1} = W^H$, computed either way, given that the matrices are stable enough for numerical invertibility, i.e., parameter α in SGFA, algorithm 1, is sufficiently high. Fig. 9 plots accuracy and discrepancy results between F and W, considering SGFA, algorithm 1, with input A, $A^H = A^T$ and $\Lambda = \Lambda^*$ (A is real). The subscript F_A (W_{A^T}) indicates that the right (left) eigenvectors of A (A^T) were computed with SGFA, algorithm 1, with input A/A^T, respectively. Each point of Fig. 9 corresponds to a run of SGFA, algorithm 1, for A (blue curve) and A^T (red curve), for a specific stability parameter $\alpha \in \{10^{-1}, 10^{-2}, \ldots, 10^{-6}\}$ (α increases from left to right). For very small α, both approximations, F_A and W_{A^T} are accurate (low objective of problem (13) corresponds to points on the left of the horizontal axis) but the discrepancy between F_A and W_{A^T} is higher (higher values on the vertical scale). Increasing α (moving from left to right), discrepancy decreases (lower values on the vertical axis),

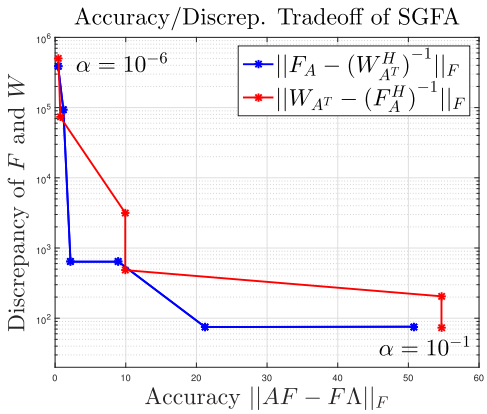


Fig. 9. Political blogs network: Approximating left and right eigenvectors, F_A and W_{A^T} (respectively), using SGFA, algorithm 1, with $\beta = 0.43$ and considering different α levels.

but the accuracy worsens (moving to the right on the horizontal axis). For a value of $\alpha = 10^{-3}$, we find a good trade-off between these two factors: accurate approximation (error ≈ 9.5) and small discrepancy (error $\approx 0.37n$).

VI. Manhattan Road Network

We now carry out a similar study by applying SGFA, algorithm 1, to the Manhattan road map.

A. Dataset

We start by briefly describing the Manhattan road network of Fig. 2(c). This Network consists of 5464 nodes that represent latitude and longitude coordinates [4], connected by 11,568 directed or undirected edges that represent one or two way streets as verified by Google Maps [3]. The underlying graph $\mathcal{G} = (\mathcal{V}, A)$ is directed and unweighted: $A_{i,j} = 1$ if and only if there is a directed edge from node i to node j. Otherwise $A_{i,j} = 0$. As it is natural (no dead-end streets in a city), graph $\mathcal{G}$ is strongly connected (there exists, at least, one path from each node to any other node). Matrix A has an eigenvalue $\lambda = 0$ with high multiplicity as seen in Fig. 12(a). Computing F with MATLAB `eig`, we find that $\sigma_{\min}(F) \approx 6.2 \times 10^{-19}$.

B. Approximating a Stable Fourier Basis

Taking into account the numerical study of Section V-B, we applied SGFA, algorithm 1, and choose a (fixed) stopping criterion $\alpha = 10^{-6}$ and a grid of six values for the contraction factor $.2 \leq \beta \leq .9$. We did not consider values $\beta > .9$ since as observed before this takes SGFA, algorithm 1, longer to converge with no noticeable performance improvement.

1) Accuracy: Fig. 10 displays the accuracy $\|AF - F\Lambda\|_{\mathcal{F}}$ (log scale) versus number of iterations (linear scale). It shows that accuracy increases exponentially with β (theorem 2); again, there is no reason to consider a high value for the contraction factor β since, for $\beta \geq 0.7$, SGFA, algorithm 1, will run for more iterations while achieving a worst overall approximation.

2) Stability: Figs. 11(a), 11(b), and 11(c) plot the stability measures of Section V, now for the Manhattan road network. We

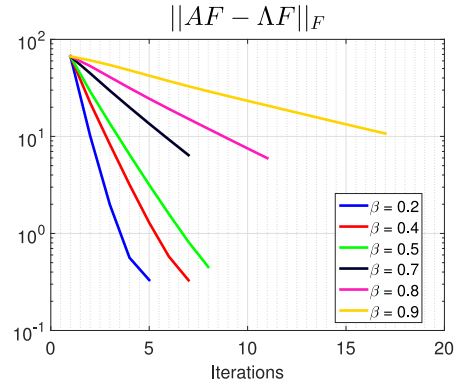


Fig. 10. Manhattan road network: Accuracy of SGFA, algorithm 1.

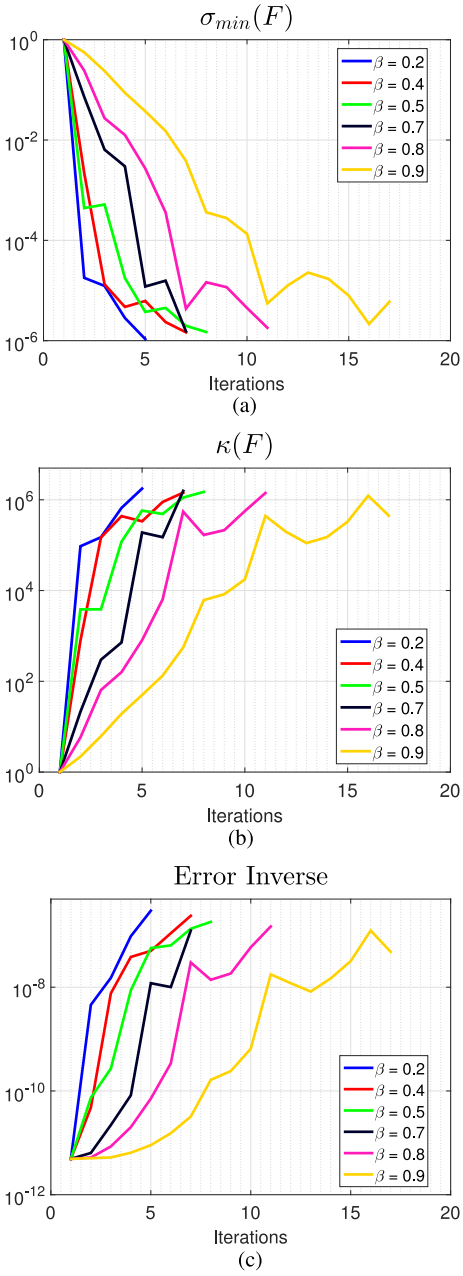


Fig. 11. Manhattan road network: Stability of SGFA, algorithm 1.

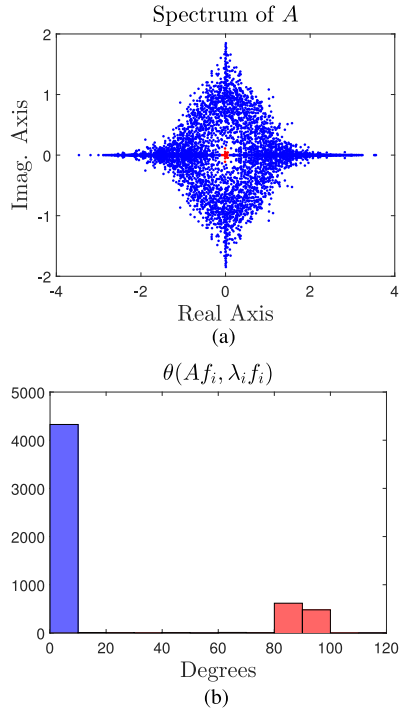


Fig. 12. Manhattan road network: Angle (34) stability measure. In (a) the spectrum of A and in (b) the histogram of the angles $\theta(Af_i, \lambda_i f_i)$. Blue indicates that $\theta(Af_i, \lambda_i f_i) < 10^\circ$ and red represents the complement $\theta(Af_i, \lambda_i f_i) \geq 10^\circ$.

observe a phase transition behavior with β where the stability measures converge at a much slower rate for $\beta > 0.7$. Again, we verify that the condition number $\kappa(F)$ tends to scale linearly with $1/\alpha$, i.e., $\kappa(F) \approx 10^6$ as seen in Fig. 11(b). As observed with the political blogs network, the same conclusion applies here: one should avoid very high values of β since there is no significant payoff in terms of numerical stability.

The accuracy measure $\|AF - F\Lambda\|_{\mathcal{F}}$ studies how close the magnitude of Af_i is to $\lambda_i f_i$. But both the magnitude and orientation of f_i are important for achieving an accurate and stable approximated Fourier Basis F. To further test how close the columns f_i of F are to being eigenvectors,[17] we compute the angle $\theta(Af_i, \lambda_i f_i)$ between the two vectors Af_i and $\lambda_i f_i$. If the angle is small, the two vectors are aligned as they should be. This represents yet another important stability measure for SGFA, algorithm 1. Generally, Af_i and $\lambda_i f_i$ are complex vectors, hence we compute $\theta(\cdot, \cdot)$ by relying on the real vector space $\mathbf{R}^{2n}$ that is isometric to $\mathbb{C}^n$ [51]

$$\forall x, y \in \mathbb{C}^n \setminus \{0\}: \quad \theta(x, y) = \arccos\left(\frac{\mathrm{Re}(x^H y)}{\|x\|_2 \ \|y\|_2}\right). \quad (34)$$

Fig. 12(a) shows a scatter plot of the eigenvalues (spectrum) of A. A majority of the eigenvalues form a (blue) cloud away from the origin, while the remaining form a dense (red) cluster very close to the origin. Fig. 12(b) is the histogram of the

[17]Note that we do not normalize f_i to norm 1 because normalizing f_i will affect the numerical stability of F, as shown in Section III for $F(\epsilon)$. This was precisely the problem with Theorem 1 that motivated formulation (13).

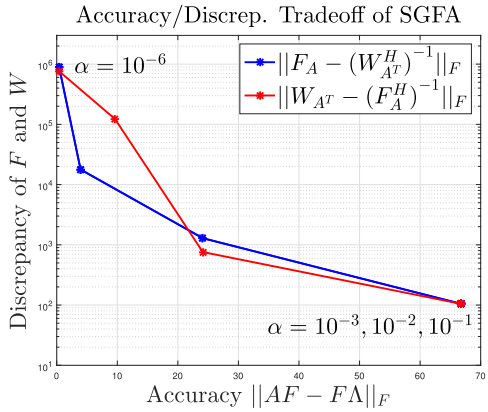


Fig. 13. Manhattan road network: Approximating left and right eigenvectors, F_A and W_{A^T} (respectively), using SGFA, algorithm 1 with $\beta = 0.43$ and considering different α levels.

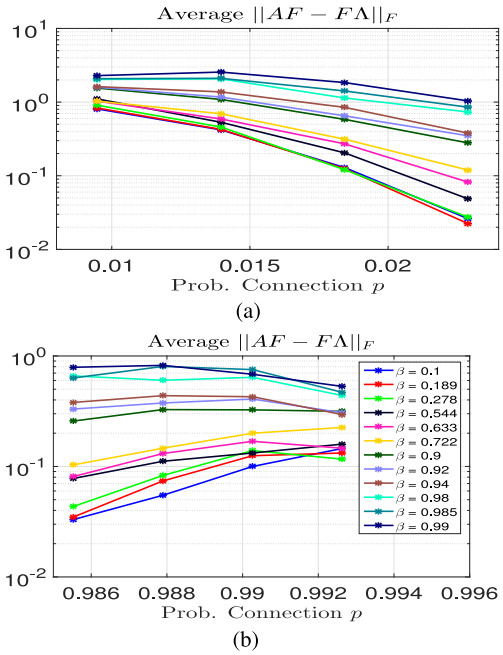


Fig. 14. Accuracy of SGFA for directed Erdős-Rényi graphs with probability of connection p. In (a) low values of p and in (b) high values of p. Mean for 100 Monte Carlo trials.

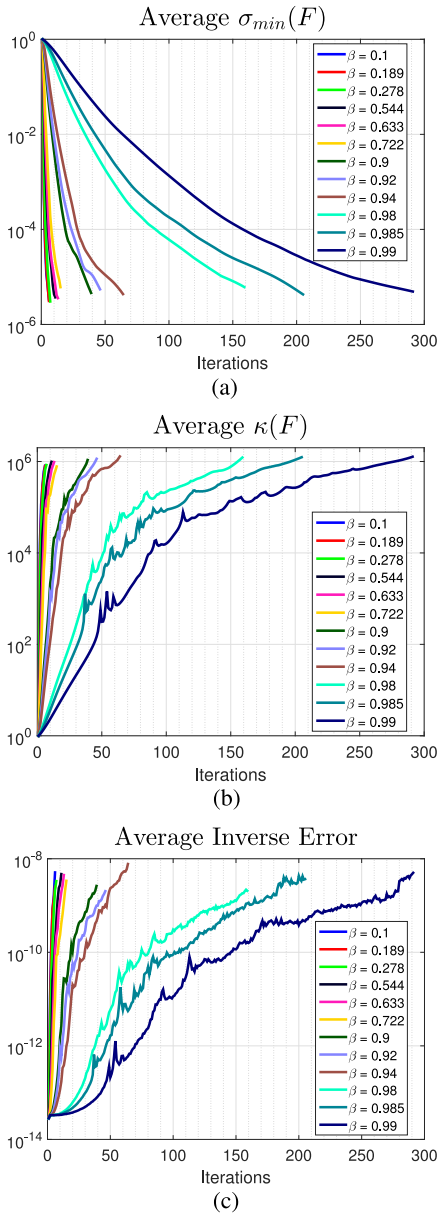


Fig. 15. Stability of SGFA for directed Erdős-Rényi graphs with $p = 0.0184$. Mean for 100 Monte Carlo trials.

angles $\theta(Af_i, \lambda_i f_i)$ between Af_i and $\lambda_i f_i$ computed by (34). The histogram has three main bars, a large (blue) bar on the left collecting about 80% of the angles $\theta(Af_i, \lambda_i f_i) < 10^\circ$, and two small (red) bars on the right with the remaining 20% of the angles $\theta(Af_i, \lambda_i f_i) \approx 90^\circ$. The large (blue) bar close to the origin corresponds to the 80% of the eigenvectors f_i associated with the eigenvalues away from zero (blue cloud) in Fig. 12(b), demonstrating that the vectors Af_i and $\lambda_i f_i$ are (almost) colinear as they should be. The two small (red) bars on the right of the histogram close to 90° correspond to the 20% of the eigenvectors $\lambda_i f_i$ associated with the eigenvalues close to zero (dense red cluster) in Fig. 12(b), apparently showing that the vectors Af_i and $\lambda_i f_i$ are practically orthogonal. This is simply an artifact

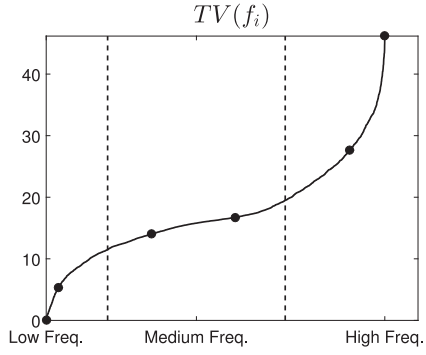
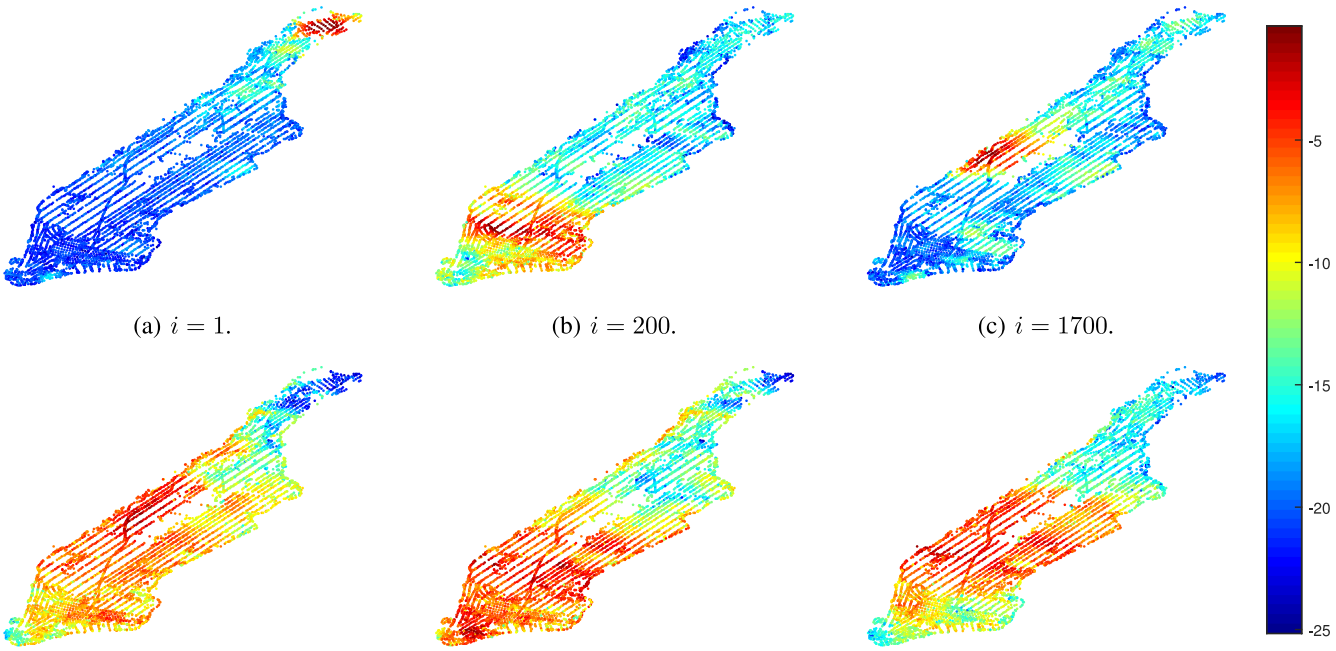


Fig. 16. Manhattan road map: total variation (35) of the Fourier basis F computed in Section VI-B with $\beta = 0.5$.



(a) $i = 1$. (b) $i = 200$. (c) $i = 1700$.

(d) $i = 3050$. (e) $i = 4900$. (f) $i = 5464$.

Fig. 17. Manhattan road map: Heat Map of the log scale magnitude of several Fourier basis vectors f_i. Index i is ordered according to the total variation in Fig. 16—figures (a)–(f) correspond to the graph frequency components marked by the black dots (from left to right) in Fig. 16.

of how the angle is computed through (34). Because for these eigenvectors f_i the associated eigenvalues $\lambda_i \approx 0$, then $\lambda_i f_i$ is approximately the zero vector, and once a vector is the zero vector the inner product computed in (34) is also zero leading to interpreting the angles as $\approx 90^\circ$.

3) Approximating Left vs Right Eigenvectors—Manhattan Road Network: Fig. 13 plots the discrepancy versus accuracy now for the Manhattan road network graph shift. For $\alpha = 10^{-1}, 10^{-2}, 10^{-3}$, SGFA, algorithm 1, ran for a single iteration with both A and A^T (this explains why there are only four points in the figure). Our conclusions replicate those for Fig. 9 for the political blogs network: increasing α decreases accuracy but improves discrepancy. For $\alpha = 10^{-4}$, SGFA, algorithm 1, achieves mean accuracy and mean discrepancy of, respectively,

$$\frac{\|AF - F\Lambda\|_{\mathcal{F}} + \left\|A^T W - W\Lambda\right\|_{\mathcal{F}}}{2} \approx 24$$

$$\frac{\left\|F_A - \left(W_{A^T}^H\right)^{-1}\right\|_{\mathcal{F}} + \left\|W_{A^T} - \left(F_A^H\right)^{-1}\right\|_{\mathcal{F}}}{2} \approx 0.19n.$$

VII. Directed Erdős-Rényi graphs

We consider synthetic random graphs; these allow graph properties to be better controlled and to investigate the impact of varying its properties. In concrete, we consider *directed* Erdős-Rényi graphs (with possible self loops) of fixed dimension $n = 100$ and varying probability of connection p. As seen in Fig. 1 (blue curve), these random models will have an unstable Fourier basis F for small and large values of p, i.e., for either very sparse or very dense graph shifts A.

1) Accuracy: Fig. 14 plots the average accuracy of algorithm 1 for directed Erdős-Rényi graphs with varying probability of connection p. The color legend of Fig. 14(b) also applies to 14(a). For each value of p, the best accuracy results are achieved by considering the smallest values of β, i.e., $\beta \in \{0.1, 0.189, 0.278\}$. This behavior is consistent with that observed in both previous sections where a high value of β was discouraged.

For sparse models, Fig. 14(a) shows that accuracy tends to increase with p regardless of the value of β. Say for $\beta = 0.189$, the average approximation error $\|AF - F\Lambda\|_{\mathcal{F}}$ is improved by a factor of 37 by moving from $p = 0.0095$ to $p = 0.0229$. Similar observations can be taken for dense directed random graphs, see Fig. 14(b). Like for sparse graphs, the average behavior of dense models depends on the value of β and improves, now as p is reduced. To be concrete, for $\beta = 0.189$, the average approximation error $\|AF - F\Lambda\|_{\mathcal{F}}$ is improved by a factor of 4 by moving from $p = 0.9926$ to $p = 0.9855$. For $\beta > 0.722$, the mean accuracy plots of Fig. 14(b) are approximately constant.

2) Stability: Fig. 15 plots stability measures for an average iteration of algorithm 1 for probability of connection $p = 0.229$ (right most point in Fig. 14(a)). The number of iterations of SGFA is not fixed, since it depends on the input matrix A and contraction factor β. Hence, we consider 100 directed Erdős-Rényi graphs and compute the median number of iterations $\text{med}(A, \beta)$ of algorithm 1. Since the sample size is 100 there exists 50 epochs for which the algorithm ran for more than $\text{med}(A, \beta)$ iterations. Fig. 15 plots the average among these 50 epochs for the first $\text{med}(A, \beta)$ iterations, i.e., for each random graph A and factor β, we average 50 time series of dimension

med(A, β). We observe that, on average, all three stability measures scale linearly with the number of iterations K, and this exponential behavior tends to be smoother and faster for low values of β, say $\beta \leq 0.9$ for Fig. 15(b) and 15(c). Again, Fig. 15(b) suggests that the termination criteria of algorithm 1 could compare $\kappa(F_k)$ against $1/\alpha$ instead of $\sigma_{\min}(F_k)$ against α, with F_k the approximated Fourier basis at iteration k.

VIII. Frequency Ordering

We now illustrate the usefulness of SGFA by analyzing and ordering the graph frequencies and graph spectral components of a (large) real world network, namely, the Manhattan road map in Fig. 2(c). Graph Signal Processing orders the graph frequencies from low to high through the (increasing) total variation TV [27], [28] of the corresponding graph spectral components. The TV of the graph spectral component f is [27], [28]:

$$\text{TV}(f) = \|f - A_{\text{norm}} f\|_1 , \;\; A_{\text{norm}} = \frac{A}{|\lambda_{\max}|}, \qquad (35)$$

where $\lambda_{\max}$ denotes the eigenvalue of A with largest magnitude. Fig. 16 plots the total variation of the columns of the Fourier basis F reordered by their TV. Fig. 16 labels regions as low, medium, and high frequencies where the boundaries are simply indicative. Fig. 17 plots the magnitude of several Fourier basis vectors (columns of F) in the two dimensional plane corresponding to the Manhattan road map. Fig. 17(a)–(f) are displayed in order of increasing variation as computed by (35), i.e., corresponding from low to high graph frequencies (dots from left to right in Fig. 16). The three graph spectral components f_i, $i = 1, 200, 1700$, shown in Fig. 17(a)–(c), tend to be sparse vectors with slowly varying magnitude. In contrast, the three spectral graph components f_i, $i = 3050, 4900, 5646$, displayed in Fig. 17(d)–(f), are much less smooth and exhibit (visually) much pronounced variation.

IX. Conclusion

This paper addresses an open problem in Graph Signal Processing, namely, it presents SGFA, an approach to compute the graph spectral components for graph signals supported by generic (directed) sparse graphs. With generic graphs, the corresponding graph shift A may have complex graph frequencies (eigenvalues), non orthogonal graph spectral components (eigenvectors), and, as commonly observed in practical real world applications, possibly repeated eigenvalues. Having an efficient procedure to approximately diagonalize these shifts A enables pursuing the graph spectral analysis of graph signals. We formulated the problem of computing accurate stable approximations of the Fourier basis F (matrix of the graph spectral components) of a generic directed graph shift A as a constrained optimization—optimize accuracy, as evaluated by the degree of invariance of the columns of F, while maintaining numerical stability, as measured by the minimum singular value of F (and corresponding condition number, since the largest singular value of F is empirically verified not to be very large). This optimization is non-convex, so, we propose an efficient algorithm—SGFA—that decreases, at least exponentially, the value of the objective per each iteration. SGFA attempts to diagonalize a triangular decomposition of A, while guaranteeing the numerical stability of the resulting local solution through a threshold α that controls the minimum singular value of our approximation. We applied SGFA to two real-world graphs, namely, the 2004 U.S. Presidential Election blogs network [2] and the Manhattan road map [3], and also to directed Erdős-Rényi graphs with either very small or very large probability of connection. The paper shows that SGFA generates efficiently good accurate approximations of F, while insuring its stability, as demonstrated experimentally with respect to several metrics of accuracy and stability (not just the optimized ones). Finally, we illustrate the application of SGFA by computing, ordering, and displaying the graph frequencies (eigenvalues) and graph spectral components (eigenvectors) for the Manhattan road map. Future work includes further analysis of structured random models that may better approximate real world graphs, for example, further analysis of Erdős-Rényi graphs with community structure.

References

[1] M. E. J. Newman, "Coauthorship networks and patterns of scientific collaboration," in *Proc. Nat. Acad. Sci.*, vol. 101, pp. 5200–5205, 2004.

[2] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 U.S. election: Divided they blog," in *Proc. 3rd Int. Workshop Link Discovery*, New York, NY, USA, 2005, pp. 36–43.

[3] J. A. Deri and J. M. F. Moura, "Taxi data in New York City: A network perspective," in *Proc. 49th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2015, pp. 1829–1833.

[4] B. C. B. Geoportal, "'NYC Geodatabase'," [Online]. Available: https://www.baruch.cuny.edu/confluence/display/geoportal/NYC+Geodatabase

[5] M. Jackson, *Social and Economic Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2008.

[6] M. Newman, *Networks: An Introduction*. London, U.K.: Oxford Univ. Press, 2010.

[7] F. R. K. Chung, *Spectral Graph Theory*. AMS, 1996.

[8] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[9] W. M. Siebert, *Circuits, Signals, and Systems*. Cambridge, MA, USA: MIT Press, 1986.

[10] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1999.

[11] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*. New York, NY, USA: McGraw-Hill, 1998.

[12] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, pp. 83–98, May 2013.

[13] J. A. Deri and J. M. F. Moura, "Spectral projector-based graph Fourier transforms," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 785–795, Sep. 2017.

[14] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory: Foundation and 1-D time," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3572–3585, Aug. 2008.

[15] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory: 1-D space," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3586–3599, Aug. 2008.

[16] B. Girault, P. Gonçalves, and É. Fleury, "Translation on graphs: An isometric shift operator," *IEEE Signal Process. Lett.*, vol. 22, no. 12, pp. 2416–2420, Dec. 2015.

[17] A. Gavili and X. P. Zhang, "On the shift operator, graph frequency, and optimal filtering in graph signal processing," *IEEE Trans. Signal Process.*, vol. 65, no. 23, pp. 6303–6318, Dec. 2017.

[18] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Sampling of graph signals with successive local aggregations," *IEEE Trans. Signal Process.*, vol. 64, no. 7, pp. 1832–1843, Apr. 2016.

[19] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *IEEE Trans. Signal Process.*, vol. 65, no. 22, pp. 5911–5926, Nov. 2017.

[20] S. Sardellitti, S. Barbarossa, and P. D. Lorenzo, "On the graph Fourier transform for directed graphs," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 796–811, Sep. 2017.
[21] W. Kahan, "Conserving confluence curbs ill-condition," Dept. Comput. Sci., California Univ. Berkeley, Berkeley, CA, USA, Tech. Rep. AD0766916, 1972.
[22] Z. Zeng, "Regularization and matrix computation in numerical polynomial algebra," in *Approximate Commutative Algebra*, Texts and Monographs in Symbolic Computation, L. Robbiano and J. Abbott, Eds. Berlin, Germany: Springer-Verlag, 2010, pp. 125–162.
[23] Z. Zeng, "Sensitivity and computation of a defective eigenvalue," *SIAM J. Matrix Anal. Appl.*, vol. 37, no. 2, pp. 798–817, Apr. 2016.
[24] L. Trefethen and M. Embree, *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*. Princeton, NJ, USA: Princeton Univ. Press, 2005.
[25] G. H. Golub and J. H. Wilkinson, "Ill-conditioned eigensystems and the computation of the jordan canonical form," *SIAM Rev.*, vol. 18, no. 4, pp. 578–619, 1976.
[26] T. Beelen and P. Van Dooren, "Computational aspects of the Jordan canonical form," in *Reliable Numerical Computation*, M. G. Cox and S. Hammerling, Eds. Oxford, U.K.: Clarendon, Jan. 1990, pp. 57–72.
[27] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014.
[28] A. Sandryhaila and J. M. F. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 80–90, Sep. 2014.
[29] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," in *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.
[30] S. Chen, R. Varma, A. Sandryhaila, and J. Kovacević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, Dec. 2015.
[31] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3775–3789, Jul. 2016.
[32] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4845–4860, Sep. 2016.
[33] Y. Tanaka, "Spectral domain sampling of graph signals," *IEEE Trans. Signal Process.*, vol. 66, no. 14, pp. 3752–3767, Jul. 2018.
[34] J. Shi and J. M. F. Moura, "Graph signal processing: Modulation, convolution, and sampling," 2019, *arXiv:1912.06762*.
[35] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3462–3477, Jul. 2017.
[36] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat, "Characterization and inference of graph diffusion processes from observations of stationary signals," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 4, no. 3, pp. 481–496, Sep. 2018.
[37] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, Dec. 2016.
[38] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology identification from spectral templates," in *Proc. IEEE Statistical Signal Process. Workshop*, 2016, pp. 1–5.
[39] J. Mei and J. M. F. Moura, "Signal processing on graphs: Causal modeling of unstructured data," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 2077–2092, Apr. 2017.
[40] J. Mei and J. M. F. Moura, "Silvar: Single index latent variable models," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2790–2803, Jun. 2018.
[41] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from filtered signals: Graph system and diffusion kernel identification," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 5, no. 2, pp. 360–374, Jun. 2019.
[42] O. Teke and P. P. Vaidyanathan, "Extending classical multirate signal processing theory to graphs–part I: Fundamentals," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 409–422, Jan. 2017.
[43] J. Zeng, G. Cheung, and A. Ortega, "Bipartite approximation for graph wavelet signal decomposition," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5466–5480, Oct. 2017.
[44] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Signal denoising on graphs via graph filtering," in *Proc. IEEE Global Conf. Signal Inf. Process.*, 2014, pp. 872–876.
[45] M. Onuki, S. Ono, M. Yamagishi, and Y. Tanaka, "Graph signal denoising via trilateral filter on graph spectral domain," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 2, no. 2, pp. 137–148, Jun. 2016.
[46] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4609–4624, Sep. 2015.
[47] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. New York, NY, USA: Cambridge Univ. Press, 2012.
[48] C. C. Paige and M. A. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Trans. Math. Softw.*, vol. 8, no. 1, pp. 43–71, Mar. 1982.
[49] A. Graham and D. Pike, "A note on thresholds and connectivity in random directed graphs," *Atlantic Electron. J. Math. [electronic only]*, vol. 3, pp. 4–5, Jan. 2008.
[50] E. Anderson *et al.*, *LAPACK Users' Guide*, 3rd ed. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1999.
[51] K. Scharnhorst, "Angles in complex vector spaces," *Acta Appl. Math.*, vol. 69, pp. 95–103, Sep. 2001.

João Domingos received the M.S. degree in electrical and computer engineering from Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal, in 2018. His master thesis won the Luís Vidigal award for the best master thesis at Instituto Superior Técnico (IST) in the scientific areas of electrical and computer engineering. He is currently working towards his Ph.D. degree in electrical and computer engineering, at Instituto Superior Técnico, Lisbon, Portugal. His research interests include distributed signal processing and optimization.

José M. F. Moura holds the engenheiro electrotécnico degree from Instituto Superior Técnico (IST), Lisbon, Portugal, and the M.Sc., E.E., and D.Sc. degrees in EECS from the Massachusetts Institute of Technology (MIT), Cambridge, MA. He is the Philip L. and Marsha Dowd University Professor at Carnegie Mellon University (CMU). He was on the faculty at IST and has held visiting faculty appointments at MIT and New York University (NYU). He founded and directs a large education and research program between CMU and Portugal, www.cmuportugal.org.

His research interests are on data science and graph signal processing. He has published over 550 papers and holds 16 US patents. Two of his patents (co-inventor A. Kavčić) are used in over four billion disk drives in 60% of all computers sold in the last 15 years worldwide and were, in 2016, the subject of a 750 million dollar settlement between CMU and a chip manufacturer, the largest ever university verdict/settlement in the information technologies area.

Dr. Moura was the 2019 IEEE President and CEO, he was IEEE Technical Activities Vice-President, IEEE Division IX Director, President of the IEEE Signal Processing Society(SPS), and Editor-in-Chief for the IEEE Transactions in Signal Processing.

Dr. Moura received the Technical Achievement Award and the Society Award from the IEEE Signal Processing Society, and the CMU College of Engineering Distinguished Professor of Engineering Award. He is a Fellow of the IEEE, the American Association for the Advancement of Science (AAAS), a corresponding member of the Academy of Sciences of Portugal, Fellow of the US National Academy of Inventors, and a member of the US National Academy of Engineering. He received a Doctor Honoris Causa from the University of Strathclyde and the Great Cross of the Order of Infant Henry from the Republic of Portugal.