Fourier transform to signal processing on graphs

Sampad Kumar Pradhan, Yubaketan Setha, Rakesh Kumar Sadangi,Jitendra Mishra

Department of Electronics and Communication Engineering, NM Institute of Engineering and Technology,Bhubaneswar , Odisha
Department of Electronics and Communication Engineering, Raajdhani Engineering College,Bhubaneswar,Odisha
Department of Electronics and Communication Engineering,Aryan Institute of Engineering and Technology Bhubnaeswar , Odisha
Department of Electronics and Communication Engineering,Capital Engineering College,Bhubaneswar,Odisha

ABSTRACT: The legacy of Joseph Fourier in science is vast, especially thanks to the essential tool that the Fourier transform is. The flexibility of this analysis, its computational efficiency and the physical interpretation it offers makes it a cornerstone in many scientific domains. With the explosion of digital data, both in quantity and diversity, the generalization of the tools based on Fourier transform is mandatory. In data science, new problems arose for the processing of irregular data such as social networks, biological networks or other data on networks. Graph signal processing is a promising approach to deal with those. The present text is an overview of the state of the art in graph signal processing, focusing on how to define a Fourier transform for data on graphs, how to interpret it and how to use it to process such data. It closes showing some examples of use. Along the way, the review reveals how Fourier's work remains modern and universal, and how his concepts, coming from physics and blended with mathematics, computer science, and signal processing, play a key role in answering the modern challenges in data science.

Keywords:
Graph signal processing
Fourier transform
Wavelets
Data science
Machine learning

I. INTRODUCTION

Dealing with data and observations has always been an important aspect of discovery in science. The idea that science is related to data was brilliantly summarised by Fourier in his own work in physics and mathematics [1][1]: "Les causes primordiales ne nous sont point connues ; mais elles sont assujetties à des lois simples et constantes, que l'on peut dé-couvrir par l'observation, et dont l'étude est l'objet de la philosophie naturelle." We argue that the same approach, mixing observations, modelling and analysis, infuses data science. Revisiting J. Tukey's vivid text from 1962 [2] about "The Future of Data Analysis", D. Donoho in [3] posits that "data science is the science of learning from data". One can add that data science can also rely on key concepts coming from physics, especially Fourier analysis, to better unveil information (e.g., in terms of frequencies) and transform or process the data.

Up to recent time, data processing was dealing with standard or regular domains: time series, images in 2-d space, videos in 2 d plus time. Now data reside more and more in irregular domains, and an important case is that of data indexed on networks (themselves coded as graphs), as it allows us to encompass both relational aspects and attributes of a given data set. The purpose of this review article is to shed light on the current challenges raised by the processing of such data on graphs. Then, it shows how this emerging domain, also called graph signal processing, by extension of classical signals, prompted the derivation of adapted and efficient tools inspired by the generalization of Fourier analysis.

1.1. What is so special about graphs?

Graphs are sets of vertices and edges. By connecting the vertices together, edges are giving a structure to the graph. These links and their weights represent the similarity, affinity or distance between vertices. From this notion of distance, the graph can be seen as a discrete space. A graph can represent the discretization of a regular domain. Examples include Euclidean spaces such as the regular grid in 2 d or the path graph in 1 d, or manifolds, like the discrete sphere or a torus. However, a graph can correspond to a more general structure such as a non-Euclidean and highly irregular space. A first step out of regular domains is made by the irregular sampling of a manifold and leads to a nearest neighbour graph where edges have different weights. Progressing further in the graph landscape, some graphs may have highly connected vertices such as influencers in a social network, which have no clear equivalent in a manifold structure. The number of neighbours may vary from vertex to vertex, yielding a local topology that changes with the localization on the graph. A graph may also be highly connected and while having a large number of vertices, it can have a small diameter.[2] In that case, any vertex may be reached within a few hops, and every vertex is a (more or less close) neighbour of every other ones. Last example, graphs may consist of several clusters of highly connected nodes, while clusters are weakly connected together. This makes graphs particularly exotic and fascinating structures that have been extensively studied in data analysis for the past 20 years [4–6].

Now, in order to generalize signal processing to graphs, the first requirement is to generalize the Fourier transform to this domain, as it is the central tool of signal processing. That is precisely what has been ~~done in~~ a series of independent pioneering works [7–12], where all the authors addressed this issue of data processing on graphs. Although these semi-nal works tackled the problem from different angles that led to distinct algebraic theories with their own properties and limitations, they all together paved the way to the Graph Signal Processing (GSP) era, and made possible to coherently decompose, process and analyze a collection of arbitrarily structured data points. Interestingly, at the same time and quite independently, related outcomes flourished under the umbrella of the communities of network science and graph-based machine learning, drawing clear bonds with GSP. When one wants to study a signal on this space, unconventional effects appear that we will describe in the following. For example, a Fourier mode is not always uniformly oscillating on a graph. It may even be localized at some place inside the graph. The uncertainty principle may be much weaker, allowing functions to be localized both on the graph and in the frequency domain. We will also see that some phenomena remain intuitive, such as the heat diffusion that spreads along the edges of the graph, or the notion of smoothness for a function. Analyzing data on a graph brings many surprises, and forces us to abandon some concepts that used to be taken for granted. It revisits old principles, making them shine in a different manner.

1.2. Objective and organization

We will start reviewing the basic ingredients and important aspects of graph signal processing, highlighting the intuitions behind. We will see how and why the Fourier transform, with its concepts of frequency and harmonic analysis, extends to the graph domain. This will also cover filtering, vertex-frequency or wavelet transforms, and other classical signal processing operations that are thus adapted to data indexed on graphs. We will strive to delve into the analogy, to illustrate the theory of GSP, as well as to give an overview of the possibilities it brings in data science. This article does not obliterate the recent review about GSP [13], or the existing collective books on GSP [14–16], which are insightful on the current state of the art of GSP. Our objective is complementary: in addition to cover the basics and some examples, we hope to exhibit the legacy of Fourier's work in the development of GSP, and in data science at large.

Section 2 presents the general way to define a graph Fourier transform, and how to interpret Fourier modes on graphs and associate a frequency. We then review in Section 3 basic operations of graph signal processing such as filtering, denois-ing, translating or sampling. Alternate spaces of representation are described in Section 4. Finally, Section 5 develops some considerations about the place of GSP (and of the concept of graph Fourier transform) within the domain of Data Science for graph-based data. Section 6 concludes.

II. GRAPH FOURIER ANALYSIS: FREQUENCIES AND MODES

Working on heat diffusion, Fourier introduced the eponymous mathematical decomposition, in order to solve specific differential equations, writing p. xii of [1][3]: "Les équations du mouvement de la chaleur, comme celles qui expriment les vibrations des corps sonores, ou les dernières oscillations des liquides, appartiennent à une des branches de la science du calcul les plus récemment découvertes, et qu'il importait de perfectionner. Après avoir établi ces équations différentielles, il fallait en obtenir les intégrales [...]." The legacy of Fourier through his Fourier transform is precisely this: how to deal with and how to solve dynamical systems. As discussed in the Introduction of [17], the now central place of Fourier analysis for data analysis comes from i) its mathematical soundness to address a large class of linear operators, ii) its physical relevance for decomposing a signal into waves related to physically relevant frequencies, and iii) the existence of remarkably efficient algorithms (Fast Fourier transform) to compute it numerically on regular discrete domains. Realizing recently that data indexed on irregular domains – such as graphs or manifolds – can be studied through a generalization of Fourier analysis led to the emergence of graph signal processing [7].

2.1. Graph Fourier modes and transform

The Fourier transform (FT) is classically a transformation to represent a signal or a function in a frequency domain. From Fourier's perspective, the purpose of introducing this transformation was to solve the heat equation. The FT was then introduced as the decomposition onto the eigenmodes of a basic linear operator that describes physical phenomena (heat diffusion, and also other types of diffusive transports, wave propagations, oscillations) on the domain. Fourier was expressing diffusion via the Laplace operator , related to the second order derivative (more exactly the divergence of the gradient) of a function. For a one-dimensional (1-d) function (or signal) of the time t, the usual Fourier modes describe oscillations

$= {}_{dt}2 \qquad = -$

$\underline{d^2}{}_{(e}{-i2\pi\nu t}$

$e^{-i2\pi\nu t}$ that satisfy $e^{-i2\pi\nu\ t}$ $)$ $(2\pi\nu)^2\ e^{-i2\pi\nu t}$. These Fourier modes are eigenvectors ofwith eigenvalues associated with the (square of the) frequency ν.

For finite discrete domains (such as that of time series, images...), the same approach leads to discrete versions of FT: the discrete Fourier transform (DFT) for the discrete 1-d line with periodic boundary conditions, or the DCT-II [18] for the discrete 1-d line with Neumann boundary conditions (cancelling the derivative at the boundaries). To connect these Fourier transforms to the case of graphs, it suffices to remark that in both discrete cases, the domain is in fact a graph (a chain or a cycle in 1 d, a grid or a torus in 2 d) and that, under adequate normalisation, the Laplacian L plays the role of − . Then, discrete transforms naturally stem from diagonalizing L. For a(n) (undirected) graph G = (V , E) where V is the set of N nodes and E the set of edges, the (combinatorial) Laplacian is defined as L = D − A, where A is the adjacency matrix ($A_{i\,j}$ = 0 if nodes i and j are not connected, and takes positive values whenever (i, j) ∊ E; this value is the weight on the

edge, which reduces to 1 for unweighted graphs) and D is the diagonal matrix of degrees ($D_{ii} = d_i = {}_j\ A_{i\,j}$, also called strength for weighted graphs). The idea of graph signal processing [7] is to generalize the notions of oscillation, frequency and FT to any graph, even if irregular, using the spectral domain of the linear operator L [19].

The notion of regular oscillations on an irregular domain such as a graph can seem paradoxical. To circumvent this difficulty, the generalization is framed in terms of smoothness and low variations of the modes, and superseding the notion of frequency in that way takes us back to the Laplacian. When it comes to study a function, one key element is to analyse its variations, as measured by the gradient operator ∇ . On a graph, it can be defined as follows. Let f be a graph signal, that is, a function defined on the vertices of a graph G. The gradient of this function between two connected vertices i and j is defined as:

$$\nabla f\,(i, j) = \overline{}\ {}^{A}i\,j\,(\,f\,(\,j) - f\,(i)) \qquad (1)$$

This gradient maps the vertex domain V to the edge domain E. It is connected to the Laplacian operator, whose expression can be rewritten for undirected graphs as:

$$\cdot$$

$$L{=}D{-}A{=}\nabla\ \nabla \qquad (2)$$

This expression[4] highlights how L conveys information about the variations of a function. In fact, for undirected graphs, as L is symmetric, it is more convenient to study than the ∇ : L is a diagonalizable operator with real and non-negative eigenvalues and its eigenmodes form a set of orthonormal vectors [19].

Henceforth, any graph signal, i.e. a function f defined on the vertices V of the graph G, can be projected on the eigenvectors of the Laplacian. Let λ_k and u_k be respectively the k-th eigenvalue and eigenvector of the Laplacian L (with the eigenvalues ordered increasingly). By analogy, the u_k ’s are called “graph Fourier modes”. This set of modes is an orthogonal

basis for the Hilbert space of the graph^{N}f . Therefore, we call a “Graph Fourier transform” f λ
signals f ∊ R (GFT) ˆ of at k ’
the change of basis written as:

$$\hat{}\quad k \quad = \quad k \qquad (3)$$
$$f \quad (\lambda\ ) \quad f\,(i)u\ (i)$$

$i \in V$

Since the $\{u_k\}_{k=0,\ldots,N-1}$ form an orthogonal basis, an inverse Fourier transform exists and reads:

$$f\,(i) \quad {}^{N-1}\ f\,(\lambda\ )u \quad (i)$$
$$= \hat{}\quad k\ \ k \qquad (4)$$

k=0

This direct–inverse GFT pair constitutes the starting point of Fourier analysis: a convenient decomposition of signals, which can help to solve equations with linear operators on graphs in the dual spectral domain.

As an example, let us go back to the diffusion equation, which can be expressed as follows on a graph [6]:

$$\frac{\partial f}{\partial t} = -L\ f \text{ with the initial condition } f\,(i, t = 0) = f_0(i) \qquad (5)$$

The solution to this equation is directly obtained in the Fourier domain, since applying the GFT of Eq. (3) to both terms

∂t $= -$ ˆ ˆ = ˆ

(λ

f ,t)

of the equality∂ (λ). Using functional calculus [8,22], this solution yields: ˆ k L f (λ , t) and so f (λ , t) $e^{t\lambda}$k f can

k k − 0 k

equivalently be written as f (i, t) = $e^{-t\ L}$ f_0(i). Fig. 1 illustrates this diffusion on different graphs. As time increases, not only the diffusion process spreads the signal over the graph, but also it attenuates more rapidly the components corresponding to larger eigenvalues. To understand this and to explain diffusion as a smoothing operation, modes need to be interpreted in terms of frequencies and oscillations or more generally in term of variation rate.

2.2. Frequencies and oscillations on graphs

Classically, the notion of frequency is so deeply rooted in the physical interpretation of the Fourier transformation that FT is often introduced as a tool for measuring oscillations. As for graphs, such notion is not intrinsic and it is critical to relate graph Fourier modes to a concept of oscillation, if not of frequency. The previously introduced Laplace operator turns out again to be central as a measure of the smoothness of a graph signal f . Indeed, the following scalar product (also called Dirichlet energy)

$$(\ f\ , L\ f\ ) = f\ \ L\ f = A_{i\,j}\ (\ f\ (\ j) - f\ (i))^2 = \nabla\ \ f\ _2^{\ 2} \qquad (6)$$

$$(i,\ j) \in E$$

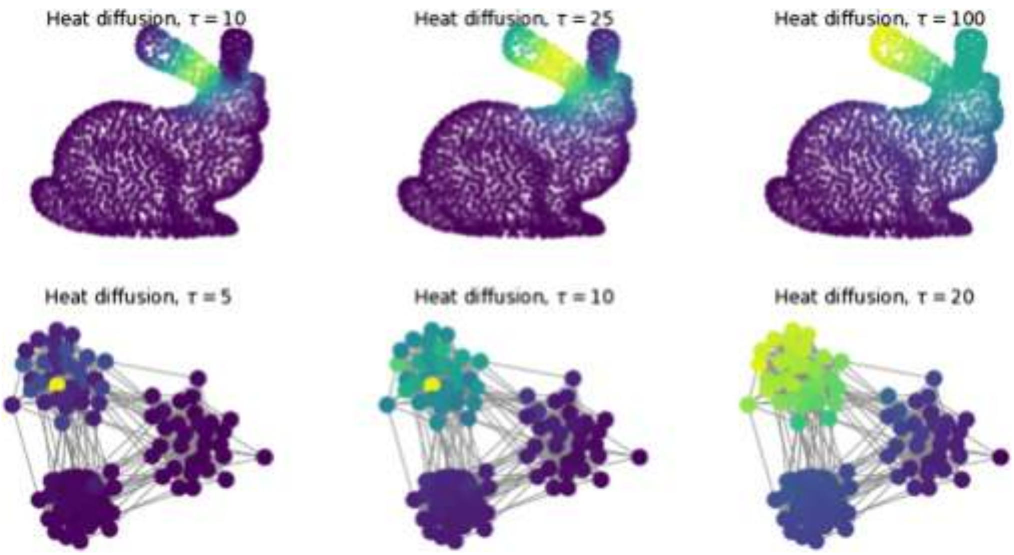

Fig. 1. Illustration of the heat diffusion over a 2-d manifold (top), and over a graph with communities (bottom), at different time τ . In both graphs, the heat spreads from node to node, following the edges. Top: the initial hot spot is a node located on the ear of the bunny. The Bunny graph is a discretization of a 2-d surface, with nodes

connected to their nearest neighbours in 3 d. Bottom: The diffusion starts inside a community and quickly spreads within it.

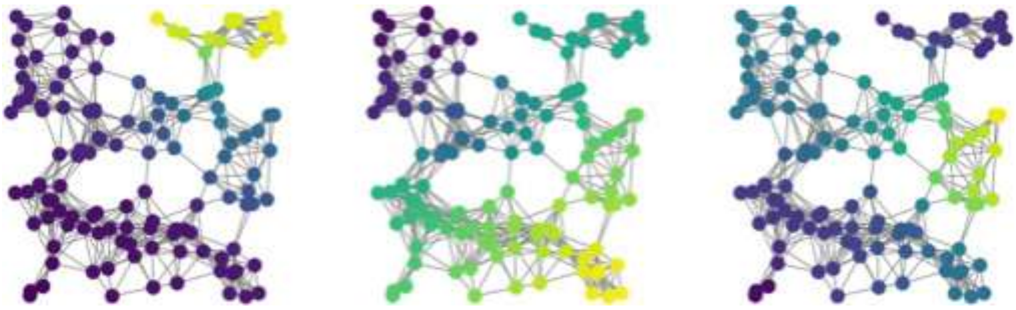

Fig. 2. Some graph Fourier modes on a random sensor graph. From left to right: first non-constant eigenvector (Fiedler vector) u_1 , second and third eigenvectors (u_2 and u_3). Colourmap: positive values in yellow, negative ones in blue.

computes the sum of the variations of a graph signal f , where $A_{i\,j}$ is the weight of the link (i, j). Hence, for each graph Fourier mode u_k , the (non-negative) eigenvalue λ_k quantifies its variation and smoothness, because: $(u_k , Lu_k)/(u_k , u_k) = \lambda_k$. This motivates us to identify this value with the mode frequency, opening the door to harmonic analysis on graphs: the larger λ_k is, the less smooth the mode is and the faster it oscillates.

This argumentation does not necessarily imply that modes are pure oscillations. For regular domains (e.g., 1-d discrete line or ring, or 2-d grids or torus), the notion of oscillation is well defined and corresponds to cosine and sine functions. In the case of graphs, the domain can be so irregular that the intuitive idea of an oscillation may simply not hold. Let us discuss the graph Fourier modes in order of increasing eigenvalue.

– No oscillation: classically, the zero frequency means no oscillation and this is the case for graphs. There is always a null eigenvalue for L. Its multiplicity is equal to the number of connected components of G, and the associated eigenvectors are constant on each connected component. Each connected component can be treated separately, and in the following, we assume that there is only one component. Then, u_0 is the only constant mode in the GFT, with eigenvalue (i.e. frequency) $\lambda_0 = 0$, as in the DFT case.

– One oscillation over the whole graph (see Fig. 2): as all other eigenmodes have to be orthogonal to u_0 , they must, at least, behave as a crude oscillation in one way: their values are positive on some nodes and negative on others. This is the case for the next smoothest possible Fourier mode, with the smallest non-zero possible frequency (eigenvalue). Also named the Fiedler vector, this component verifies an insightful property: as it must change sign and because this is costly in term of variations, it switches where the graph is the less connected. It turns out that, on a graph made of two weakly connected communities, the Fiedler vector will have a different sign on each of the communities. It is precisely this property that led to the celebrated spectral clustering method in machine learning [23].

– Several oscillations: one can build orthogonal oscillations on the graph with increasing frequencies by considering the successive eigenvectors that satisfy the minimax principle,

$$u_k = \underset{v \in / \;_{k-1}}{\mathrm{Argmin} v} \frac{(v, L\, v)}{\quad 2} \qquad (7)$$

where $_{k-1}$ is the subspace spanned by the first k − 1 eigenvectors. These constraints of smoothness and orthogonality lead to cosine and sine functions in the standard domain. On the graph domain, there is no analytic formulas for the eigenvectors, and in practice they may have complex shapes. Still, they fulfil the constraints: they alternate sign more often than the modes with smaller frequency (eigenvalue), while varying as smoothly as possible from vertex to vertex.

This supports the interpretation of a mode's eigenvalue as its frequency. Returning to the GFT formula, Eq. (3) means that any graph signal can be decomposed into a sum of graph Fourier modes, each of them being associated with a frequency that measures how fast it oscillates. This lays the foundations of harmonic analysis on graphs, where Eq. (6) properly defines the frequency of a Fourier mode, the graph Fourier modes are the eigenmodes of the Laplacian – a physically relevant linear operator – and analytically, there exists a direct decomposition (3), with an exact reconstruction formula (4).

2.3. Uncertainty principle, localization, and sparsity

On a regular domain, homogeneous and isotropic, the heat diffusion process does not depend on the localization or direction within the domain. This is reflected by the shape of the Laplacian eigenvectors. They are evenly spread over the domain. The meaning of "spreading" here is related to the fact that modes have non-zero values everywhere on the domain (except on some isolated points). Intuitively, this spreading can be measured using the L^2 -norm f $_2$ of f on compact subdomains and by comparing them. If the norms remain close for all subdomains of fixed size, the function is well spread. The entropy H of a function is another way to measure spreading. For a function f on a discrete, finite domain of N samples, the entropy is a non-negative real value given by:

$$N$$
$$\underline{|\, f\,(n)|^2} \qquad \underline{f\,(n)|^2}$$
$$H(\,f\,) \qquad \log$$
$$= - \;_n \quad f\;_2 \qquad |\, f\;_2$$
$$=1$$

Alternatively, one can use the L^1- $_1$, which is connected to the notion of sparsity, with sp(f) = f $_1$/
norm, · f $_2$. A large
sp value indicates a large spreading over the domain [24,25]. The sparsity measures the concentration
of f of a function

on the domain or equivalently, its lack of spreading: it describes how localized a function is on part(s) of the domain, with zero values outside.

This spreading property of Fourier modes has been extensively studied and it is the core of the uncertainty principle. It states that the concentration (or sparsity) of a signal and the concentration of its Fourier transform are related and cannot be both small at the same time. Different versions of the uncertainty principle have been established for different spreading

f
measures [25]. For instance, the entropy of a function f and its Fourier transform ˆ (in the regular discrete 1D domain) are
related through:

$$H(\,f\,) \qquad f\,) \qquad N$$
$$+ \; H(\hat{\;} \quad \geq \log \qquad (8)$$

From this relationship, it follows that, if f is made of a single non-zero value, i.e. H(f) = 0, then its Fourier transform has to be broader than a single peak in the frequency domain.

On irregular domains such as graphs, as seen in Sec. 2.2, homogeneity is lost and the Fourier modes have different behaviours at different locations in the graph. To understand what happens to spreading and to the uncertainty principle, let us take an extreme case. Let G be a graph made of two disconnected subgraphs G_1 and G_2, where G_1 is a single vertex. The two subgraphs form two independent spaces and the set of Fourier modes of G is the union of the Fourier modes of these two subgraphs. On this single vertex graph G_1 , the signal is reduced to a single value and the Fourier basis is only one eigenvector. Therefore, a signal concentrated on G_1 , with zeros on G_2 , is orthogonal to any Fourier mode of G_2 and will have a single non-zero value in the frequency domain, associated with the single mode of G_1 . Considering the two subgraphs separately, the uncertainty principle of Eq. (8) is valid on the subgraph G_1 , as N = 1. However, considering G, Eq. (8) does not hold anymore. Similarly, the Fourier modes have at least two different behaviours and localisations depending on their associated space (from G_1 or G_2). For this union of graphs, the lower bound of the uncertainty principle is given by the graph G_1, a particular region of G. This is also shown by the generalization of Eq. (8) (see [25,26]):

$$H(\;f \qquad 2 \;\; \log$$
$$) \quad + \; H(\hat{\;} \;\; \geq - \,\mu \qquad \mu \;\; = i, j \quad j \qquad (9)$$
$$f\,) \qquad , \qquad (\delta \quad , u\;)$$

The quantity μ is the mutual coherence of the canonical basis $\{\delta_i\}_i$ on the graph and the graph Fourier basis $\{u_j\}_j$, with $(\cdot, \cdot)$ being the scalar product. In our extreme case, $\mu = 1$, given by the scalar product of δ_i localized on G_1 and the eigenvector of the graph Laplacian of G_1. Note that in the case of a ring graph, which is the standard discrete setting, we recover Eq. (8).

Hence, with this basic example we see that the uncertainty principle depends on the local structure of the graph and its inhomogeneity. Furthermore, adding a small number of connections between G_1 and G_2 can be seen as a small perturbation of the graph Laplacian of G and hence of its eigenvectors. As a result, the spreading of the eigenvectors will not increase much when connecting the two subgraphs.

To go beyond this global uncertainty principle, one needs to introduce uncertainty principles taking into account the local structure of a graph. Then, it becomes possible to obtain tighter bounds that depend on the localisation on the graph, and on the structure of the neighbourhood. This is the work proposed in [27].

2.4. Generalizations: operators on graphs

The proposed analogy to derive the GFT can be modified if one focuses on a specific operator on the graph instead of relying on the Laplacian (and physically on the way a signal diffuses as heat would do).

For instance, graph Fourier modes were introduced in [12] as the generalized eigenvectors of the adjacency matrix A, using its decomposition in Jordan form (A may not be diagonalizable in the general case of directed graphs). The emphasis is put on the shift invariance property of the Fourier modes. The core of the analogy comes from the direct 1-d ring graph. Let us write A_d its adjacency matrix, which happens to correspond exactly to the translation of f along the direction of the network: $(\forall i \in V)\ f(i+1) = (A_d f)(i)$. In this case, the derivative is also $\nabla_d = A_d - \mathbf{I}_d$, so the gradient and the adjacency matrix share the same set of generalized eigenvectors. Moreover, in the directed ring case, A_d is not symmetric but it is normal, i.e. $A_d A_d = A_d A_d$, so that it is diagonalizable and the set of eigenvectors are the DFT modes [13]. It is also the case of the gradient and, as a consequence, of $L = \nabla_d \nabla_d$. Now this framework can be used for the undirected ring graph with adjacency matrix $A_u = A_d + A_d$; then, one has $L = 2\mathbf{I}_d - A_u$ and L and A_u possess the same eigenvectors. Note also that, up to a constant, the Laplacians of the directed and undirected rings are equal. Keeping the viewpoint of physics, the formula $L = \nabla \nabla$ makes the two approaches (through L and through A) very similar. On an undirected ring graph,

$A_u = (\nabla_d + \mathbf{I}_d) + (\nabla_d + \mathbf{I}_d)$, it commutes with L, and the eigenvectors for A_u and L are the same.

The analogy of [12] is relevant as it emphasises the role of a possible shift operator, here A on the graph as it generalizes the usual discrete signal processing. A general approach would be to consider other choices of shift operators and suitable measures of variation or smoothness to define variations. For instance, Sections II of [28] or [29] describe the different pos-sible choices of shift operators and their associated measure of variation, and hence of frequency. A variant is to normalise

L into the so-called normalised Laplacian: $L_n = D^{-1/2} L D^{-1/2}$, well known for instance for graph spectral clustering [23]. In the case of directed graphs, [21] advocates why one could use the random walk operator as the shift operator, and the directed Laplacian as proposed in [20] to measure variations. Other recent works [30,31] deal with the directed case by considering more elaborate measures of variations and building the graph Fourier modes accordingly.

Nonetheless, as the connection with physical process (e.g., the heat equation) is not straightforward in all these works, from now on, we will restrict our discussion to harmonic analysis defined on the Laplacian of undirected graphs, a choice that is closer to the initial work and motivation of Fourier.

III. GRAPH SIGNAL PROCESSING

With graph Fourier transform and harmonic (or frequency) analysis thus defined, we have all appropriate building blocks to develop a theory of signal processing on graphs. In this section, we discuss some central tools in signal processing and how they are adapted to graph signals.

3.1. Graph filters

The first operation that finds a simple equivalence for graph signals is filtering. That is, the extraction of a part of a signal that corresponds to a subset of frequencies for ideal filters, or more generally the weighting of the Fourier coefficients $f(\lambda_k)$ obtained with the GFT of Eq. (3). As advocated in Section 2.2, the frequency of the graph Fourier modes is associated with its eigenvalue as an eigenmode of the Laplacian. It follows that a linear filter H can be designed by specifying a filtering function h that weights independently the Fourier coefficients; the resulting filtering of a function f reads:

$$H(f)(i) = \sum_{k=0}^{N-1} h(\lambda_k)\hat{f}(\lambda_k)u_k(i) \qquad (10)$$

An example of low-pass filtering is shown in Fig. 3 (i). The solution to the diffusion equation (5) would be another instance of low-pass filtering with the kernel function $h(\lambda_k) = e^{-t\lambda_k}$, with parameter $t \in R^+$ controlling the cut-off frequency.

Some points deserve attention. The graph spectral domain is a discrete and finite set spanned by the (non-negative) eigenvalues of the graph Laplacian. It is an irregularly sampled half-line that contrasts with the regularly spaced frequencies of the DFT. Hence, there are several manners to define a graph filter:

i) define $h(\lambda)$ over the non-negative real line. This allows us to define filters that can be applied to graphs of different shapes and sizes (and therefore different eigenvalues). Optionally, the function can be a function of λ/λ_{max} where λ_{max} is the maximum eigenvalue of L (according to Gershgorin's circle theorem, λ_{max} is bounded by 2 $\max_i d_i$). Another possible normalisation is to use the normalized Laplacian L_n as it automatically sets the maximum frequency for every graph: $\lambda_{max}(L_n) = 2$ [19]. Whatever the chosen normalisation, it provides a sort of "universal" design of filters and allows for easily reproducing classical forms of filters in signal processing. The idea of defining continuous filter has been first introduced for the design of graph wavelets [8]. It takes advantage of the particular line shape of the spectral domain, a convenient structure for scale changes;

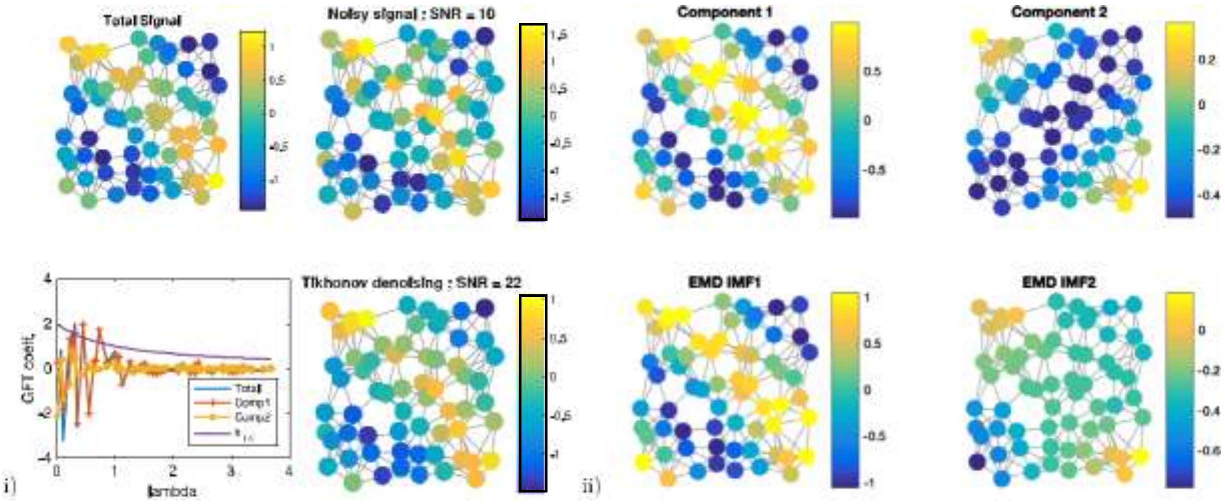

Fig. 3. A graph signal, composed of two components (on the far right) is processed: i) with Tikhonov denoising as in Sec. 3.2 to increase its SNR; and ii) with graph empirical mode decomposition as from Sec. 4.3 [34] so as to separate its two components, which would not be directly separable either in the frequency domain (see the GFT on bottom left) or in the vertex domain.

ii) set only the values at the specific eigenvalues λ_k of the studied graph. This approach is sometimes termed "frequency aware" design (e.g., [32]). In particular, this form is used in [12] and introduces the DSP framework on graphs. It has the advantage of being simple (the filter is a vector of size N) and it leverages linear algebra to design filters [33]. The main drawback of this method is that it is graph-dependent and does not allow us to transpose a filter from one graph structure to another.

A complementary approach for designing filters takes its root in classical DSP, where the building block is the shift operator. An important class of filters is then obtained as polynomials, and more generally as rational fractions, of this shift operator. Then, the same polynomial coefficients serve to express the filter operator, its frequency response or the resulting filtered signal:

$$\begin{array}{llllll} & & (\quad f & & & \\ M \quad h\ L^{m} & M \quad h\ \lambda^{m} &)(i) \quad N-1 \quad M & {}^{m}\, f(\lambda\ )u\ (i) & & \\ H = \ = {}_{m}\text{or } h(\lambda_k) = & = \ m\ {}_{k}\ {}^{\text{or } H} & = \quad = & h_m(\lambda_k)\lambda_k\ \hat{}\ {}_{k}\ {}_{k} & & (11) \\ \quad 0 & \quad 0 & k=0 \quad 0 & & & \\ m & m & \quad m & & & \end{array}$$

Here, $h(\lambda)$ is defined for all λ, yet only the values at the λ_k 's are relevant if a "frequency-aware" design is chosen. We do not elaborate on the construction for graph filters, but we refer the reader to [32], where finite-response (polynomials) and infinite-response (rational fractions) graph filters have been studied.

3.2. Denoising of graph signals

One important purpose of filters is to perform signal denoising, that is to say, to remove (or to attenuate) part of the signal that is not deemed relevant for the analysis. Graph filters offer a way to do this. In order to fix ideas, let us consider the simple case where the signal is the sum of two terms $s = f_0 + n$, where f_0 corresponds to the useful information and n is the spurious noise. In general, one necessary condition to ensure rational denoising of this mixture is to be able to use some prior on the two components, to increase the signal-to-noise ratio. For example, if we know that the relevant signal is smooth, namely that it is localised on some low-frequency domain, a correctly designed low-pass filter aims at reducing (if not zeroing) the high-frequency content related to the noise in the mixture, while preserving the informative low-pass component, as illustrated in Fig. 3 (i). A design to find an optimal balance between these two competitive actions is to go back to Eq. (6) that measures the smoothness of a graph signal, and we rewrite denoising as an inverse problem with Tikhonov regularisation [7]:

$$\begin{array}{ll} f^{*} \quad \arg\min \quad f \quad s\,2 \quad \gamma\,(\,f\,,L\,f\,) & \qquad (12) \\ \quad = f \qquad \| \;-\; \|_2 \,+ & \end{array}$$

The parameter γ controls the trade-off between data fidelity (first term of the sum) and smoothness (second term of the sum). Using GFT leads to the solution to this compromise $\hat{f}^{*}(\lambda_k) = \hat{s}(\lambda_k)/(1 + \gamma\,\lambda_k)$, and the corresponding low-pass graph filter with response function $h(\lambda_k) = 1/(1 + \gamma\,\lambda_k)$ readily follows.

In fact, any classical filter function can be imported as a filter in GSP. More generally, the combination of filters with different shapes in filter banks has led to major progresses in the field for denoising or efficient decoding of the information in graph signals. It also paved the way to introduce powerful new representations of graph signals: wavelet and Gabor transformations that we will revisit under the light of GSP in Section 4.

3.3. Convolution, localization, and translation

The convolution operator is another important operation that is deeply rooted in harmonic analysis and signal processing. Defining its equivalent for graph signal processing is an intricate task as, in general, graphs are irregular structures for which shift invariance is meaningless.[5] However, a classical property of the Fourier transform is that the convolution of two functions in the initial domain becomes a multiplication in the Fourier domain. Transposing this duality property straightforwardly to the GSP domain, the authors of [8,35] proposed the following definition. The convolution $f * g$ of two functions f and g defined on a graph is defined at each vertex i as:

$$\begin{array}{ll} (\,f \quad g)(i)\ {}^{N-1}\ f\,(\lambda\ )g(\lambda\ )u \qquad (i) & \\ \quad * \qquad {}_{=}\hat{} \qquad k \quad \hat{} \quad k \quad k & \qquad (13) \\ k=0 & \end{array}$$

In other words, the convolution is defined as the operation that multiplies two functions in the spectral domain. It verifies the usual properties of standard convolution [35]: it is commutative and admits the Dirac δ function (defined as $\delta(0) = 1$ and $\delta(i) = 0$ for all i 0) as neutral element, i.e. $\delta * f = f$.

More interestingly, this definition entails an important consequence of the convolution with respect to the localisation of graph signals when convoluted with localised kernels. Indeed, one can localise a graph signal f towards a specific node v by convolving it with a distribution δ_v, which is 0 everywhere except at v. To see this, let us remark that $\hat{\delta}_v(\lambda_k) = u_k(v)$ and apply the definition (13):

$$(f * \delta_v)(i) = \sum_{k=0}^{N-1} \hat{f}(\lambda_k) u_k(i) u_k(v) \qquad (14)$$

It is shown in [8,35] that if $\hat{f}(\lambda_0)$ is significantly large at $\lambda_0 = 0$ and if the function $\hat{f}$ can be accurately approximated by a low-order polynomial on the spectral domain, then $(f * \delta_v)$ is localised on the vertex domain, centred around vertex v and decays when moving away from it.

Notice that formula (14) is not the unique way to define a translation on a graph. Although it is relevant for localisation, it suffers from other limitations and it is not so appropriate to consider it as a translation or a shift operator. Other approaches try to recover some group structure [36], or some explicit shift in the vertex domain [37]. Still, in many situations, the localisation property of the convolutions defined by Eq. (14) is sufficient. This is the case for wavelet and vertex-frequency transforms (see Sec. 4), as it would be for kernel methods on graphs (see also Sec. 5.2).

3.4. Sampling

The periodic sampling paradigm of the classical Shannon–Nyquist theorem for bandlimited signals does not apply to arbitrary graphs (besides, what does periodic sampling mean on an arbitrary graph?), so new sampling theorems have been designed to tackle the problem of sampling – and reconstructing – graph signals. First, one extends the notion of bandlim-itedness to graph signals: signals that are linear combinations of the first few low-frequency graph Fourier modes. Writing $U_k = (u_1| \ldots |u_k) \in \mathbb{R}^{n\times k}$, we have the formal definition: a graph signal $z \in \mathbb{R}^n$ is k-bandlimited if $z \in \mathrm{Span}(U_k)$, i.e. $\exists\, \alpha \in \mathbb{R}^k$ such that $z = U_k \alpha$.

By construction, these graph signals “vary slowly” along any paths of the graph. Moreover, this particular low-rank assumption enables generalized versions of the Nyquist–Shannon sampling theorems, which take into account the explicit structure of the underlying graph.

In order to write down the important questions pertaining to this problem, let us introduce some notations. Sampling entails selecting a subset $S = (s_1, \ldots, s_m)$ of m nodes of the graph. With each possible sampling set we associate a measurement matrix $M = (\delta_{s1} |\delta_{s2} | \ldots |\delta_{sm}) \in \mathbb{R}^{m\times n}$ where $\delta_{si}(j) = 1$ if $j = s_i$, and 0 otherwise. Now, consider a k-bandlimited signal $z \in \mathrm{Span}(U_k)$. The measurement of z on S reads:

$$y = M z + n \in \mathbb{R}^m \qquad (15)$$

where n models measurement noise. The sampling question boils down to: how should we sample S such that one can robustly recover any bandlimited z given its measurement y? There are three important components of this question:

(i) how many samples m do we allow ourselves (m = k being the strict theoretical minimum)? (ii) how much does it cost to sample S? (iii) how do we in practice recover z from y and how much does that inversion cost? Depending on how one decides to answer the previous questions, a number of generalized sampling theorems adapted to graph signals have been proposed, e.g., [28,38–44]. We refer the interested reader to [45] for a recent review of the existing schemes.

IV. ALTERNATE REPRESENTATIONS OF GRAPH SIGNALS

As we have seen, the (graph) Fourier transform defines a dual representation space that in some cases, eases the defi-nition or the implementation of processing tasks on (graph) signals. Yet, the spectral domain holds its own limits and it is not necessarily the best space for representing the data. For instance, FT maps local features in the direct space to global characteristics delocalised over the entire spectral domain. Then, an intermediate representation of signals is to mix the domains and to express data onto a basis (or a frame) indexed both by the time (or space for images) and the frequency. This domain has been particularly active for time signals in the past 40 years, with seminal works in time-frequency anal-ysis (e.g., [46,47]) and with the introduction of the wavelet transform [48,49] and all its variants. This exploration beyond Fourier has recently been extended to graph signals and we review now some of its most important results.

4.1. Vertex-frequency decompositions

Thanks to the localisation operator offered by the convolution in Sec. 3.3, and the frequency analysis explained in Sec. 2, one can develop vertex-frequency representations for graph signals, as in [35], where the authors develop the equivalence of a windowed Fourier analysis to signals on graphs.

Given a window function $g \in L^2(R)$ (e.g., a Gaussian function, or any smooth and localised function), the (classical)
windowed Fourier transform is the decomposition in:

$S f(t, \nu) = f(u) g(u - t)\, e^{-i2\pi\nu u}\, du$ (16)

R

that is the decomposition on atoms $g(u - t)\, e^{i2\pi\nu u}$, modulated at frequency ν and translated (and so localised) around t. Because g is localised, the resulting decomposition extracts localised information near instant t at frequency ν. Now, using the generalized convolution operator of Eq. (13), this transform generalizes on graphs as:

N−1

n

$S f(i, \lambda_k) =$ $f(n) u_k(n)\, \hat{g}(\lambda\ ) u\ (i) u\ (n)$ (17)

∈ =0

V

As shown in [35], this equation reduces to $S f(i, \lambda_k) = (f, g_{i,\lambda k})$ where the function $g_{i,\lambda k}$ is the kernel g, which is frequency modulated at λ_k and localised by convolution around vertex i (thanks to Eq. (14)).

4.2. Wavelets on graphs

Wavelet transforms is a decomposition analogous to the previous one, except that the frequency modulation is replaced by a scaling of the atoms. Given the impact of Wavelet transform in data processing [49], it is not a surprise that the generalization of continuous wavelet transforms on graphs was one of the seminal works in graph signal processing [8].

In a nutshell, a classical wavelet transform is obtained by choosing atoms of a frame (or of a basis) as being band-pass functions (with zero-mean) that are localised at a specific instant by translation and shifted in frequency by scaling (instead of modulating as it would be in the windowed Fourier transform). Given the initial band-pass kernel $\psi \in L^2(R)$, called the mother wavelet, one defines the wavelet transform of f at scale $s \in R^+$ and instant t as the scalar product with the scaled (by s) and translated (at t) wavelets as:

= s * s = ˆ * ˆ

W $\frac{1}{}$ $\overline{u - t}$

f (t, s) ψ f (u) du ψ (sv) f (v) $e^{i2\pi\nu t}$ dν (18)

R R

If ψ satisfies the admissibility condition R | ˆ | = ∞ , then the transform is invertible that (ψ $^*(\nu)$ $^2/\nu$) dν c_ψ < and the function wavelet coefficients multiplied by the scaled and translated wavelets: f (u)
can be expressed by combining back the
$1/c_\psi$ +∞ W f (t, s)$\psi_{(s,t)}$ (u) dt ds/s. More details can be found in [48,49]. =

0 R

For functions indexed on graphs, the spectral form of Eq. (18) leads to a straightforward generalization using GFT and the localisation operator as initially discussed in [8]. For f a graph signal on V , one defines:

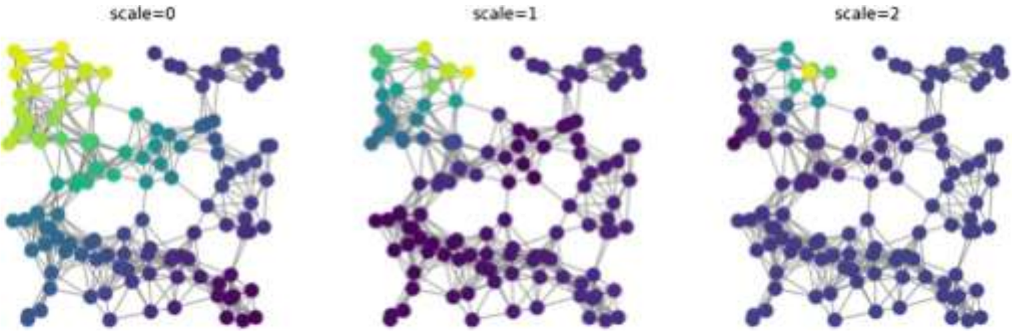

Fig. 4. A graph wavelet at different scales. The wavelet spread decreases with the scale, as for a wavelet on a regular domain.

$$W_f(i, s) = \sum_{k=0}^{N-1} \psi(s\lambda_k)\hat{f}(\lambda_k)u_k(i) \qquad (19)$$

the wavelet transform of f at vertex i and scale s > 0. Examples of graph wavelets are shown in Fig. 4.

Having at hand such a transform opens the way to multiscale processing of graph signals, and many exploratory works flourished in the recent years. Many other multiscale transforms have been studied in the graph setting: filterbanks [9], Laplacian pyramids [50], Haar-like wavelets [51,52], wavelets via random forest [53], and this includes the original discrete wavelet transform built from the diffusion wavelets [54] before using the GFT analogy. We refer also to [29] for a review of various multiscale transforms on graphs.

4.3. Extension: extraction of dictionaries or of modes on graphs

A more general point of view is to adopt a data-driven approach to decompose a signal. Then, the atoms or the modes on which to represent a signal are not specified in advance. A general possibility is to estimate one (or several) dictionary(-ies) that are particularly well adapted at representing a collection of graph signals, while striving to meet certain particular constraint, such a sparsity, for instance. This is the line of thought followed in [55,56]. For compression, learning dictionaries outperform spectral wavelet transform, and this is true on graphs as it is for the usual setting. Both parametric (e.g., with a polynomial expansion in the Laplacian, as in Eq. (11)) and non-parametric filters (where the atoms are coded by their individual coefficients at each frequency λ_k) have been considered and have interesting properties.

An alternative representation is to try to decompose the signal as a sum of oscillating modes at separate local frequen-cies; this is one of the diverse roads to local frequency analysis [17]. For 1-d signals, this can be achieved in a data-driven manner using Empirical Mode Decomposition (EMD) [57]. For graph signals, we have developed EMD in [34]. The general idea of EMD, and of graph EMD, is to extract the fastest oscillation present in the signal (thanks to the so-called sifting process), and then iterate on the residual this same processing so as to achieve an exact additive decomposition. Once adapted to graphs [34], this procedure offers a data-driven approach to represent a graph signal, where the oscillations are not constrained by the graph Fourier modes. A visual example is shown in Fig. 3 (ii); here the graph signal is the sum of two components having overlapping GFTs, and the transform is still able to separate them.

V. FOURIER AS A DATA SCIENTIST

To end this article on the basics of GSP, our objective is to illustrate the ubiquitous legacy of Fourier's work in Data Science through some applications of GSP, and the use of GSP as part of Machine Learning (ML) and Data Science. More originally, the connections between GSP and ML will be illustrated through the use of operators describing linear dynamics on graphs. This leads us back to Fourier because, an aspect of his original work was to find solutions for specific operators, the ones dealing with linear dynamics.

5.1. Applications of graph signal processing

As titled in the recent overview [13], GSP has already found many applications. From the two foremost applications that are often considered, one is the generalization of image processing to irregular domains (e.g., missing samples, or on some mesh as in Fig. 1) [15,58]. An important case is the processing of 3-d point clouds because a proximity graph (k-nearest neighbours or -radious neighbourhood) is a natural space on which a signal of colour and/or depth can be considered.

A second application is in network science at large, that is, when confronted to data where the relations between elements or data points is as important (or even more important) than the data values (or signal) on these points. Since the late 1990s, many sensor networks, communication networks, transportation networks, biological networks (e.g., functional connections in the brain or protein and/or gene interaction networks) and even social or economical networks were studied from the angle of their structure [4,5] and/or their dynamics [6]. The introduction of graph signal processing was a way to re-introduce the importance of signal on these networks, and of joint processing of the structure and the signal. For sensor networks, this has been useful for reconstructing sensor readings from compressed measurements [59], or anomaly detection in such networks [60]. An exciting domain of application is that of brain networks, where one seeks to use both the structure of the network (be it a structural network or one coming from functional connectivity) and the signals of activity of the brain (e.g., see Chapter 31 in [14]). As discussed in Part 4 of [14], adding the GSP point of view opens also new roads in social network analysis.

Another aspect where networks and signal processing meet is when one considers joint time-vertex dynamics, as in [61]. There, the authors leverage GSP for time series on graphs, showing how to build a higher-dimensional FT where the graph is one dimension and time the second dimension. As the time dimension can equivalently be considered as a path graph, this amounts to defining GFT and more generally GSP operations into a tensor product of graphs, so that the theories reviewed here are directly usable.

5.2. Operators for data processing on graphs

The spectral domain associated with GFT is the proper domain to deal with dynamics involving linear operators on graphs. This allows one to easily consider such operators for data processing on graphs.

Let us consider diffusion on graphs. The way heat diffuses is the same everywhere on a regular domain. On an irregular domain like a graph, where vertices may be more connected in some places of the graph and less in other ones, the homogeneity is broken and, as a consequence, heat diffusion changes. A signature is that the graph Fourier modes are not anymore well spread and can be localised in particular places within the graph. On the one hand, it is disturbing as it disrupts our intuitive idea of a Fourier mode. On the other hand, it is interesting as it allows us to characterize the inhomogeneities in the graphs. Let us consider the heat kernel exp(−t L) or any similar low-pass filter defined in the graph spectral domain; applying it on a graph with the localization operator and measuring its spreading allows us to probe the inhomogeneity of the space [27]. It brings us back to the uncertainty principle presented in Sec. 2.3. The same spreading on the spectral domain leads to different spreading on the graph domain, depending on the local graph structure. The multiscale community detection of [62] works under the same intuition: information about the community a node is in “diffuses” naturally more easily in its community that in the next ones.

A second example is to consider the behaviour of operators as a means to characterize graphs (while we were character-izing nodes in the first example). An idea for that can be found for instance in [63], where a distance between two graphs G_1 and G_2 (resp. with Laplacians L_1 and L_2) is introduced as:

$$d_{gd}(G_1,G_2) = \max_{t \in \mathbb{R}_+} \| e^{-tL_1} - e^{-tL_2} \|_F \quad (20)$$

It can be generalized by considering the multiple δ_v ’s as initial conditions of Eq. (5) [64] before combining them in a new measure of distance between graphs. A nice point is that the choice of the heat kernel generalizes well to other operators. The work in [65] does exactly that: in order to classify graphs, it considers not only the behaviour of the heat kernel exp(−t L) at all times t, which are, by nature, low pass in term of graph frequencies, but also the simple wave equation:

$$\frac{\partial^2 f}{\partial t^2} = -L f \text{ (plus initial conditions)} \quad (21)$$

Keeping fully the operators solving these equations (e^{-tL} for diffusion, or e^{-itL} for waves) is too much, and [65] keeps only their trace on a large range of t; still, convincing results in graph classification are obtained, using efficient numerical approximations for large graphs.

5.3. GSP and machine learning

In machine learning (ML) also, the legacy of Fourier is apparent. A classical goal of machine learning is to find structure and classes in data, based on how similar the data points are, and it is fairly classical to consider the points are the vertices of a graph and the edges are based on some thresholding of the similarities. Among graph-based methods in ML, the heat (or diffusion) kernel has long been recognised as a good approach to use kernel methods on graph data [66]. Spectral clustering [23] is also a method that emphasises the role of the graph coding of a data to cluster complex data.

While GSP is foremost dealing jointly with graphs and signals, it has been shown that it is possible to specifically probe the structure of networks, coded as graphs, using concepts from GSP. For instance, spectral clustering can be shown to be using filters in the GFT domain [67] and, combining that with graph sampling techniques, efficient algorithms have been developed [44,68]. Multiscale representations offered by spectral graph wavelets lead to the development of new multiscale community detection for graphs without signals [62]. Semi-supervised learning also has found a GSP-based interpretation as [69] shows that it amounts to a form of Wiener filtering on graphs.

Another fruitful combination of ML and GSP is for the task of graph inference, where one tries to estimate a graph from observed signals in the nodes [70]. Generally, it can be framed as an optimization problem for graph learning, and many works tackled this trying to learn the graph Laplacian in order for the data to be smooth on this graph [71], or for the graph to satisfy structural constraints [72]. A more physical interpretation of the same task is to infer the graph while observing signals diffused in time, i.e. signals that result from the application of a graph filter at each time step. This has been considered independently in [73,74] and these works give interesting results in that case. Still, they assume knowledge of the eigenfunctions of L and the work [75] revisits the question with a more physical insight. The authors show in [75], when a sparse prior for the data can be used, how to learn the graph while assuming that locally the observed signals result from a dynamics of heat diffusion, as in (5). Given the omnipresence of the diffusive process, even on graphs [6], this appears as a sound approach.

The final example consists of convolution operators on graphs, as seen in Sec. 3.3. Current research on alternative formu-lations of convolution on graphs is today driven by attempts to generalize convolutional Deep Neural Networks (DNN), that obtain currently state-of-the-art results for data on regular domains, such as images [76], to data defined on graphs. For graph data, with objectives to classify either nodes or graphs, frameworks of DNN for graphs were evolved from GSP, and reviewed as “geometric deep learning” in [77], or smartly captured in a unified formalism in [78], while [79] draws a recent picture of current graph neural methods. The idea is to say that one layer of a neural network on graph consists of a non-linearity applied to the output of a graph spectral filter (as from Sec. 3.1) that can be implemented directly in the spectral domain (or be considered in the node domain for a better efficiency), and whose coefficients have to be learnt (usually by backpropagation). Then, pooling operations are obtained as local averages on the graph (e.g., [80]), or by graph coarsening (where it is possible to have some spectral guarantees, e.g., [81]). The current picture is that we have many architectures of deep neural networks on graphs, some rooted in GSP and Fourier transform on graphs, and the interpretation of GFT appears as a good way to explore what is a convolution on graphs, and so how to generalize DNN on irregular domains.

VI. CONCLUSION

This article has proposed an overview of graph signal processing, which is a recent and important addition in data science to deal with data on graphs. The emphasis has been on the generalization of the Fourier transform to data on graphs as a keystone to generalize data processing. Also, it has been a journey where we tried to show that the way Joseph Fourier produced science is acute, even so in modern time. Indeed, the modernity of Fourier is that his work combines physical intuition (or insight about what is studied), mathematical precision, and good computational (or algorithmic) properties [47]. For signals on graphs, we have tried to explain how the proposed GFT is sound for data, how it is mathematically accurate, and why it is useful for data processing. One last aspect is missing somehow, as we do not have yet algorithms as efficient as the Fast Fourier transform, despite progresses thanks to, e.g., iterative algorithms for polynomial approximations of functions of L [82], or to nice greedy approximation schemes of the Laplacian [83]. Improving on this aspect is a first challenge awaiting GSP, the second challenge being to come to fruitful applications for data on complex and irregular domains, graphs being such a case. Like the seminal article [7], the text is also an invitation to read further on the developments in signal processing on graphs.

REFERENCES

[1] J.B.J. Fourier, Théorie analytique de la chaleur, Chez Firmin Didot, père et fils, 1822.

[2] J.W. Tukey, The future of data analysis, Ann. Math. Stat. 33 (1) (1962) 1–67.

[3] D. Donoho, 50 years of data science, J. Comput. Graph. Stat. 26 (4) (2017) 745–766.

[4] E.D. Kolaczyk, Statistical Analysis of Network Data: Methods and Models, Springer, 2009.

[5] M.E.J. Newman, Networks: An Introduction, Oxford University Press, 2010.

[6] A. Barrat, M. Barthlemy, A. Vespignani, Dynamical Processes on Complex Networks, Cambridge University Press, 2008.

[7] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, 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. 30 (3) (2013) 83–98.

[8] D.K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory, Appl. Comput. Harmon. Anal. 30 (2) (2011) 129–150.

[9] S.K. Narang, A. Ortega, Perfect reconstruction two-channel wavelet filter banks for graph structured data, IEEE Trans. Signal Process. 60 (6) (2012) 2786–2799.

[10] S.K. Narang, A. Ortega, Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs, IEEE Trans. Signal Process. 61 (19) (2013) 4673–4685.

[11] A. Agaskar, Y.M. Lu, A spectral graph uncertainty principle, IEEE Trans. Inf. Theory 59 (7) (2013) 4338–4356.

[12] A. Sandryhaila, J.M.F. Moura, Discrete signal processing on graphs, IEEE Trans. Signal Process. 61 (7) (2013) 1644–1656.

[13] A. Ortega, P. Frossard, J. Kovaceviˇc,´ J.M.F. Moura, P. Vandergheynst, Graph signal processing: overview, challenges, and applications, Proc. IEEE 106 (5) (May 2018) 808–828.

[14] P.M. Djuric,´ C. Richard (Eds.), Cooperative and Graph Signal Processing, Elsevier, 2018.

[15] O. Lezoray, L. Grady (Eds.), Image Processing and Analysis with Graphs. Theory and Practice, CRC Press, 2012.

[16] L. Stankovic,´ E. Sejdi (Eds.), Vertex-Frequency Analysis of Graph Signals, Springer, 2019.

[17] P. Flandrin, Time-Frequency/Time-Scale Analysis, Academic Press, 1999.

[18] G. Strang, The discrete cosine transform, SIAM Rev. 41 (1) (1999) 135–147.

[19] F.R.K. Chung, F.C. Graham, Spectral Graph Theory, Regional Conference Series in Mathematics, vol. 92, American Mathematical Society, Providence, RI, USA, 1997.

[20] F. Chung, Laplacians and the Cheeger inequality for directed graphs, Ann. Comb. 9 (1) (2005) 1–19.

[21] H. Sevi, G. Rilling, P. Borgnat, Harmonic analysis on directed graphs and applications: from Fourier analysis to wavelets, arXiv preprint, arXiv:1811. 11636v2 [Math.FA], 2018.

[22] L. Grady, J.R. Polimeni, Discrete Calculus: Applied Analysis on Graphs for Computational Science, Springer, 2010.

[23] U. Von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.

[24] D.L. Donoho, P.B. Stark, Uncertainty principles and signal recovery, SIAM J. Appl. Math. 49 (3) (1989) 906–931.

[25] B. Ricaud, B. Torrésani, A survey of uncertainty principles and some signal processing applications, Adv. Comput. Math. 40 (3) (2014) 629–650.

[26] B. Ricaud, B. Torrésani, Refined support and entropic uncertainty inequalities, IEEE Trans. Inf. Theory 59 (7) (2013) 4272–4279.

[27] N. Perraudin, B. Ricaud, D.I. Shuman, P. Vandergheynst, Global and local uncertainty principles for signals on graphs, APSIPA Trans. Signal Inf. Process. 7 (2018).

[28] A. Anis, A. Gadde, A. Ortega, Efficient sampling set selection for bandlimited graph signals using graph spectral proxies, IEEE Trans. Signal Process. 64 (14) (2016) 3775–3789.

[29] N. Tremblay, P. Gonçalves, P. Borgnat, Design of graph filters and filterbanks, in: P.M. Djuric,´ C. Richard (Eds.), Cooperative and Graph Signal Processing, Elsevier, 2018, pp. 299–324.

[30] S. Sardellitti, S. Barbarossa, P. Di Lorenzo, On the graph Fourier transform for directed graphs, IEEE J. Sel. Top. Signal Process. 11 (6) (2017) 796–811.

[31] R. Shafipour, A. Khodabakhsh, G. Mateos, E. Nikolova, A digraph Fourier transform with spread frequency components, arXiv preprint, arXiv:1705.10821, 2017.

[32] E. Isufi, A. Loukas, A. Simonetto, G. Leus, Autoregressive moving average graph filtering, IEEE Trans. Signal Process. 65 (2) (2017) 274–288.

[33] J. Liu, E. Isufi, G. Leus, Autoregressive moving average graph filter design, in: Proc. 6th Joint WIC/IEEE Symposium on Information Theory and Signal Processing in the Benelux, Louvain-la-Neuve, Belgium, 19–20 May 2016.

[34] N. Tremblay, P. Borgnat, P. Flandrin, Graph empirical mode decomposition, in: Proc. European Signal Processing Conference (EUSIPCO 2014), Lisbon, Portugal, 1–5 September 2014, pp. 2350–2354.

[35] D.I. Shuman, B. Ricaud, P. Vandergheynst, Vertex-frequency analysis on graphs, Appl. Comput. Harmon. Anal. 40 (2) (2016) 260–291.

[36] B. Girault, P. Goncalves, E. Fleury, Translation on graphs: an isometric shift operator, IEEE Signal Process. Lett. 22 (12) (2015).

[37] N. Grelier, B. Pasdeloup, J. Vialatte, V. Gripon, Neighborhood-preserving translations on graphs, in: 2016 Proc. IEEE Global Conference on Signal and Information Processing, Greater Washington, D.C., USA, 7–9 December 2016, pp. 410–414.

[38] S. Chen, R. Varma, A. Sandryhaila, J. Kovacevic, Discrete signal processing on graphs: sampling theory, IEEE Trans. Signal Process. 63 (24) (2015) 6510–6523.

[39] L.F.O. Chamon, A. Ribeiro, Greedy sampling of graph signals, IEEE Trans. Signal Process. 66 (1) (2018) 34–47.

[40] A. Sakiyama, Y. Tanaka, T. Tanaka, A. Ortega, Eigendecomposition-free sampling set selection for graph signals, IEEE Trans. Signal Process. 67 (10) (2019) 2679–2692.

[41] Y.H. Kim, A. Ortega, Toward Optimal rate allocation to sampling sets for bandlimited graph signals, IEEE Signal Process. Lett. 26 (9) (2019) 3775–3789.

[42] M. Tsitsvero, S. Barbarossa, P. Di Lorenzo, Signals on graphs: uncertainty principle and sampling, IEEE Trans. Signal Process. 64 (18) (2016) 4845–4860.

[43] N. Tremblay, P.O. Amblard, S. Barthelmé, Graph sampling with determinantal processes, in: Proc. 25th European Signal Processing Conference (EU-SIPCO 2017), Kos Island, Greece, 28 August–2 September 2017, pp. 1674–1678.

[44] G. Puy, N. Tremblay, R. Gribonval, P. Vandergheynst, Random sampling of bandlimited signals on graphs, Appl. Comput. Harmon. Anal. 44 (2) (2018) 446–475.

[45] P. Di Lorenzo, S. Barbarossa, P. Banelli, Sampling and recovery of graph signals, in: P. Djuric, C. Richard (Eds.), Cooperative and Graph Signal Processing, Elsevier, 2018, pp. 261–282.

[46] L. Cohen, Time-Frequency Analysis, Prentice-Hall, 1995.

[47] P. Flandrin, Explorations in Time-Frequency Analysis, Cambridge University Press, 2018.

[48] I. Daubechies, Ten Lectures on Wavelets, SIAM, 1992.

[49] S. Mallat, A Wavelet Tour of Signal Processing, Academic Press, 1999.

[50] D.I. Shuman, M.J. Faraji, P. Vandergheynst, A multiscale pyramid transform for graph signals, IEEE Trans. Signal Process. 64 (8) (2016) 2119–2134.

[51] N. Tremblay, P. Borgnat, Subgraph-based filterbanks for graph signals, IEEE Trans. Signal Process. 64 (15) (August 2016).

[52] J. Irion, N. Saito, Hierarchical graph Laplacian eigen transforms, JSIAM Lett. 6 (2014) 21–24.

[53] L. Avena, F. Castell, A. Gaudillière, C. Mélot, Intertwining wavelets or multiresolution analysis on graphs through random forests, Appl. Comput. Harmon. Anal. (2018).

[54] R.R. Coifman, M. Maggioni, Diffusion wavelets, Appl. Comput. Harmon. Anal. 21 (1) (2006) 53–94.

[55] D. Thanou, D.I. Shuman, P. Frossard, Learning parametric dictionaries for signals on graphs, IEEE Trans. Signal Process. 62 (15) (2014) 3849–3862.

[56] X. Zhang, X. Dong, P. Frossard, Learning of structured graph dictionaries, in: Proc. 37th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012), Kyoto, Japan, 25–30 March 2012, pp. 3373–3376.

[57] N. Huang, Z. Shen, S. Long, M. Wu, H. Shih, Q. Zheng, N.C. Yen, C.C. Tung, H. Liu, The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, Proc. R. Soc. Lond., Ser. A, Math. Phys. Eng. Sci. 454 (1971) (1998) 903–995.

[58] G. Cheung, E. Magli, Y. Tanaka, M.K. Ng, Graph spectral image processing, Proc. IEEE 106 (5) (2018) 907–930.

[59] X. Zhu, M. Rabbat, Graph spectral compressed sensing for sensor networks, in: Proc. 37th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012), Kyoto, Japan, 25–30 March 2012, pp. 2865–286.

[60] H.E. Egilmez, A. Ortega, Spectral anomaly detection using graph-based filtering for wireless sensor networks, in: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014), Florence, Italy, 4–9 May 2014, pp. 1085–1089.

[61] F. Grassi, A. Loukas, N. Perraudin, B. Ricaud, A time-vertex signal processing framework: scalable processing and meaningful representations for time-series on graphs, IEEE Trans. Signal Process. 66 (3) (2018) 817–829.

[62] N. Tremblay, P. Borgnat, Graph wavelets for multiscale community mining, IEEE Trans. Signal Process. 62 (20) (2014) 5227–5239.

[63] D.K. Hammond, Y. Gur, C.R. Johnson, Graph diffusion distance: a difference measure for weighted graphs based on the graph Laplacian exponential kernel, in: Proc. 1st IEEE Global Conference on Signal and Information Processing (GlobalSIP 2013), Austin, TX, USA, 3–5 December 2013, pp. 419–422.

[64] D. Bao, K. You, L. Lin, Network distance based on Laplacian flows on graphs, CoRR, arXiv:1810.02906, 2018.

[65] A. Tsitsulin, D. Mottin, P. Karras, A. Bronstein, E. Müller, NetLSD: hearing the shape of a graph, in: Proc. 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, 19–23 August 2018, pp. 2347–2356.

[66] R. Kondor, J.D. Lafferty, Diffusion kernels on graphs and other discrete input spaces, in: Proc. 19th International Conference on Machine Learning (ICML-2002), Sidney, 8–12 July 2002, pp. 315–322.

[67] N. Tremblay, G. Puy, R. Gribonval, P. Vandergheynst, Compressive spectral clustering, in: Proc. 33rd International Conference on Machine Learning (ICML 2016), New York, 19–24 June 2016, pp. 1002–1011.

[68] N. Tremblay, A. Loukas, Approximating spectral clustering via sampling: a review, CoRR, arXiv:1901.10204, 2019.

[69] B. Girault, P. Gonçalves, E. Fleury, A.S. Mor, Semi-supervised learning for graph to signal mapping: a graph signal Wiener filter interpretation, in: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014), Florence, Italy, 4–9 May 2014, pp. 1115–1119.

[70] G. Mateos, S. Segarra, A.G. Marques, A. Ribeiro, Connecting the dots: identifying network structure via graph signal processing, IEEE Signal Process. Mag. 36 (3) (2019) 16–43.

[71] X. Dong, D. Thanou, P. Frossard, P. Vandergheynst, Learning Laplacian matrix in smooth graph signal representations, IEEE Trans. Signal Process. 64 (23) (2016) 6160–6173.

[72] H.E. Egilmez, E. Pavez, A. Ortega, Graph learning from data under Laplacian and structural constraints, J. Sel. Top. Signal Process. 11 (6) (2017) 825–841.

[73] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, M.G. Rabbat, Characterization and inference of graph diffusion processes from observations of stationary signals, IEEE Trans. Signal Inf. Process. Netw. 4 (3) (2018) 481–496.

[74] S. Segarra, A.G. Marques, G. Mateos, A. Ribeiro, Network topology inference from spectral templates, IEEE Trans. Signal Inf. Process. Netw. 3 (3) (2017) 467–483.

[75] D. Thanou, X. Dong, D. Kressner, P. Frossard, Learning heat diffusion graphs, IEEE Trans. Signal Inf. Process. Netw. 3 (3) (2017) 484–499.

[76] Y. LeCun, Y. Bengio, G.E. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[77] M.M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, P. Vandergheynst, Geometric deep learning: going beyond Euclidean data, IEEE Signal Process. Mag. 34 (4) (2017) 18–42.

[78] M. Bontonou, C.E.R. Kós Lassance, J.-C. Vialatte, V. Gripon, A unified deep learning formalism for processing graph signals, CoRR, arXiv:1905.00496, 2019.

[79] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, M. Sun, Graph neural networks: a review of methods and applications, CoRR, arXiv:1812.08434, 2018.

[80] F. Gama, A.G. Marques, G. Leus, A. Ribeiro, Convolutional neural network architectures for signals supported on graphs, IEEE Trans. Signal Process. 67 (4) (2019) 1034–1049.

[81] A. Loukas, P. Vandergheynst, Spectrally approximating large graphs with smaller graphs, arXiv, CoRR, arXiv:1802.07510, 2018.

[82] D.I. Shuman, P. Vandergheynst, P. Frossard, Chebyshev polynomial approximation for distributed signal processing, in: Proc. 7th IEEE International Conference on Distributed Computing in Sensor Systems (IEEE DCOSS ’11), Barcelona, Spain, 27–29 June 2011, pp. 1–8.

[83] L. Le Magoarou, R. Gribonval, N. Tremblay, Approximate fast graph Fourier transforms via multilayer sparse approximations, IEEE Trans. Signal Inf. Process. Netw. 4 (2) (2018) 407–420.