Universitat Politècnica de Catalunya

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

Bachelor's Degree in Engineering Physics

A graph signal processing solution for defective directed graphs

Final Bachelor's Thesis

Author:
Júlia Barrufet Ribes

Director:
Dr. A. Ortega (USC Viterbi)

Co-director:
Dr. P. Salembier (UPC - ETSETB)

June 29, 2020

Abstract

The main purpose of this thesis is to find a method that allows to systematically adapt GSP techniques so they can be used on most non-diagonalizable graph operators.

In Chapter 1 we begin by presenting the framework in which GSP is developed, giving some basic definitions in the field of graph theory and in relation with graph signals. We also present the concept of a Graph Fourier Tranform (GFT), which will be of great importance in the proposed solution.

Chapter 2 presents the actual motivation of the research: Why the computation of the GFT is problematic for some directed graphs, and the specific cases in which this happen. We will see that the issue can not be assigned to a very specific graph topography, and therefore it is important to develop solutions that can be applied to any directed graph.

In Chapter 3 we introduce our proposed new method, which can be used to form, based on the spectral decomposition of a matrix obtained through its Schur decomposition, a complete basis of vectors that can be used as a replacement of the previously mentioned Graph Fourier Transform. The proposed method, the Graph Schur Transform (GST), aims to offer a valid operator to perform a spectral decomposition of a graph that can be used even in the case of defective matrices.

Finally, in Chapter 4 we study the main properties of the proposed method and compare them with the corresponding properties offered by the Diffusion Wavelets design. In the last section we prove, for a large set of directed graphs, that the GST provides a valid solution for the problem.

Acknowledgements

This thesis is the result of five months of research at the Ming Hsieh Department of Electrical and Computer Engineering at USC Viterbi, developed as the final bachelor thesis of my Engineering Physics Degree at the ETSETB school at UPC.

I want to especially thank my Professor Antonio Ortega, who has supervised this project from the beginning, giving me essential guidelines and support for the development of this research. At every point of this thesis his active advice has been an enormous help. Thanks to the support and advice of Professor Philippe Salembier, for his valuable insights on the structure and progress of the project and for attending all the doubts raised troughout the development of project. To my school at UPC, for giving me the opportunity to dedicate a semester of my degree at USC.

And finally to Anna and Sergi, for their help and support, and to my family and friends, without their constant support it wouldn't have been possible.

Contents

Introduction

Data is constantly being collected around us. All this volume of information is stored by researchers, governments and companies for many different purposes. We not only refer to massive amounts of data collected for applications on the well known Big Data area such as bank, security and communications information, it is also being critical for medical solutions, sensor control or the study of traffic patterns. And something that all this applications have in common is that the raw data needs to go through a specific processing in order to be useful for its purpose. But the abundance and complexity of such amounts of data means that the information now resides on irregular and complex structures that do not lend themselves to standard tools [1].

To face this issue, graphs are being used to represent the geometric structure of data domains in numerous applications, including social, energy, biology, transportation, sensor, and neural networks. For example, in social networks individual users establish connections with each other and interact generating large amounts of data. In this case graph nodes represent users and graph edges correspond to possible connections between users. The data on these graphs can be visualized as a finite collection of samples with one sample at each node in the graph. This setup could be used, for example, to infer the structure of a community by its relations and friendships, perceive alliances between agents through game theoretic models [2], quantify the connectedness of the world or study the relevance of particular users [3].

In order to process all the data generated and structured as a graph signal, it is necessary to adapt basic methods used in signal processing to generalize fundamental operations such as filtering, translation, modulation or dilation. The emerging field of Graph Signal Processing (GSP) merges algebraic and spectral graph concepts with computational harmonic analysis in order to process signals on graph structures [4]. Doing so requires to extend classical signal processing concepts such as Fourier transform and frequency response so they can be used on data residing on graphs. With this purpose, the Graph Fourier Transform (GFT) has been created [5], in order to develop the tools necessary to adapt the classical setting and extend it to graphs. With this adapted tools we can filter graph signals, and also sample and denoise them.

A significant problem that can be faced when using graph signal processing on real-world graphs arises when the matrix corresponding to a graph operator cannot be diagonalized this complicates the study of the graph in the spectral domain (Chapter 2). Finding the spectral decomposition of the matrix representing one of the graph operators is essential to build graph signal processing tools. This problem appears when dealing with defective directed graphs, a case that authors tend to avoid assuming that the matrices they study are always diagonalizable.

The main purpose of this research has been to first identify the structure and properties of the set of graphs causing this issue to then to develop a systematic tool that can be used to obtain an alternative form to the GFT concept, valid for graphs where the graph operator is not diagonalizable. This thesis studies the diagonalizability of directed graphs, describes the proposed method and studies its main properties.

State of the art and related work

In recent years a lot of research has been developed around graphs. Some of the most studied areas on this emerging field are 1) the discovery of efficient models to represent large high-dimensional data [6] [7], 2) the quantification of network characteristics [8], 3) the labeling of large amounts of data [9] 4) and, in the field of GSP, the development of transforms for data indexed by graphs such as regression algorithms [10], wavelet decomposition [11] [12] [13] or filter banks [14] [15]. Specifically, the definition of a Fourier transform for use in uncertainty analysis on graphs in the frequency domain has been deeply studied [16].

We next introduce recent relevant contributions before presenting the work developed in this research.

GFT for irregular directed graphs: The difficulties to use the standard GFT [3] on some directed graphs have been widely studied, with the purpose of finding alternatives that allow GSP tools to be used on these group of graphs. The problem, described in further detail in Chapter 2, is based on the non-diagonalizability of the adjacency matrix of a graph. Many studies use the Jordan matrix decomposition [17] to find an spectral decomposition of the graph. For example, in [18] and [19] a graph Fourier transform for which the spectral components are the Jordan subspaces of the adjacency matrix is presented. More recently, in [20] a method to replace a given adjacency shift $\mathbf{A}$ by a diagonalizable shift $\mathbf{A}_D$ is obtained via the Jordan-Chevalley decomposition, while this method leads to a diagonalizable shift operator, it starts by computing the Jordan form, which is well known to be numerically unstable. And this is a common limitation for all solutions based on the Jordan decomposition. This situation leads to investigate alternative approaches. For example, in [21] the graph Fourier basis is built as the set of orthonormal vectors that minimize a continuous extension of the graph cut size, known as the Lovasz extension. Other alternatives to the Jordan form are presented in [22] and [23].

Modifications for irregular topologies: The solutions mentioned above face another problem, which is the lack of physical interpretation. For example, the idea presented in [20] offers an alternative that modifies the graph adjacency matrix (based on numerical approximation criteria), obtaining results based on a graph with a different topology from the original one, since it only modifies the algebraic representation of the graph adding or changing the elements of the matrix, causing a modification of the graph edges and weights. Therefore, another approach taken to address defective directed graphs, where the GFT cannot be calculated, is to find a slight topological modification that results in a more feasible graph. The clearest example of this methodology is implemented in the Pagerank algorithm [24], with the so called Pagerank Teleportation [25].

Wavelet configurations: Finally, we want to mention another alternative, common also in the field of classic signal processing, consisting on the grouping of frequencies by forming

wavelets. An approach based on wavelets is proposed by [26] where they propose a novel method for constructing wavelet transforms of functions defined on the vertices of arbitrary finite weighted graphs. Similar ideas are presented in [27] and [28]. Our proposed method is inspired by the Diffusion Wavelets method [13] [29]. This model offers a tool to build a spectral decomposition for diffusion operators, such as the adjacency matrix of a graph.

Outline and main contributions

The previously mentioned problems that GSP techniques face when it comes to defective directed graphs led to the development of this thesis. Its main purpose its to find a method that allows to systematically adapt those already known techniques so they can be used in almost every graph, and especially on graphs with a non-diagonalizable graph operator.

This thesis has been structured in four parts. In Chapter 1 we begin by presenting the framework in which GSP is developed, giving some basic definitions in the field of graph theory and its algebraic and polynomial representation. Then, we introduce how concepts from classical signal processing are approached in the graph environment, describing how graph filters are created and presenting the concept of a Graph Fourier Tranform (GFT), which will be of great importance in the proposed solution.

Chapter 2 presents the actual motivation of the research: Why the computation of the GFT is problematic for some directed graphs, and the specific cases in which this happen. This allows us to understand the problem in order to find a solution. We will see that the problem of a defective adjacency matrix cannot be associated to a specific topology, and therefore the developed solution should be applicable to any directed graph.

In Chapter 3 we introduce our proposed new method. We start by briefly describing the ideas that inspired it: The Schur decomposition, which is an algebraic method used to obtain the spectral decomposition of a matrix (valid for defective matrices), and the Diffusion Wavelets formulation, which shows a possible spectral decomposition of a diffusion operator.

We built a method that can be used to form, based on the spectral decomposition of a matrix obtained through its Schur decomposition, a complete basis of vectors presented as an invertible matrix that can be used as a replacement of the previously mentioned Graph Fourier Transform. The proposed method, the Graph Schur Transform (GST), aims to offer a valid operator to perform a spectral decomposition of a graph that can be used even in the case of defective matrices.

Finally, in Chapter 4 we study the main properties of the proposed method. We describe, in particular, its properties regarding subspace invariance, orthogonality and spectral localization, and compare them with the corresponding properties offered by the Diffusion Wavelets. In the last section we prove, with a large set of directed graphs, that the GST provides a valid solution for the problem.

Chapter 1

Fundamentals of Graph Signal Processing

1.1 Graphs

We start this section by introducing basic definitions related to graphs and their algebraic representation. Next, we describe the main graph operators and present the role of invariant subspaces and polynomials in graph signal processing.

1.1.1 Basic definitions

Definition 1.1. A **graph** $\mathcal{G}(V, E)$ is a discrete structure defined as a set of nodes V and edges E, where an edge e_{ij} represents a link between node i and node j.

Definition 1.2. A graph can be **weighted**, if any edge e_{ij} can take a real positive weight ω_{ij}, or **unweighted**, if the weight for all its edges is 1. An inexistent edge will be represented by a weigth zero, both in weighted and unweighted graphs.

Definition 1.3. A graph is **undirected** if the edge e_{ji} exists whenever e_{ij} exists and $\omega_{ji} = \omega_{ij}$. A graph where e_{ji} or e_{ij} may not exist and in general $\omega_{ji} \neq \omega_{ij}$ is a **directed** graph. Edges on directed graphs are represented by arrows from node i to node j.

Definition 1.4. A **subgraph** is defined as a subset of nodes and edges from a larger graph. Given a graph $\mathcal{G}(V, E)$, a subgraph $\mathcal{G}_s(V_s, E_s)$ is such that $V_s \subset V$ and $E_s \subset E$.

A possible characterization of graphs comes from the number of neightbors linked to each node, defined as the node degree [5].

Definition 1.5. The **node degree** is defined for undirected graphs as the total sum of the weights of its edges:

$$d_i = \sum_j \omega_{ij} \tag{1.1}$$

For directed graphs the in-degree and the out-degree are defined:

$$d_i^{in} = \sum_j \omega_{ij}, \quad d_i^{out} = \sum_j \omega_{ji} \tag{1.2}$$

From the definition of node degree we can define a notion of regularity depending on the variation of degree leading to the definition of regular graph.

Definition 1.6. A **regular graph** is an unweighted graph where all the nodes have the same number of neighbours [30]. A cycle graph (Definition 1.9) would be an example of a regular graph.

In most cases, real graphs will not be exactly regular. However, the node degree will provide a measure to define the regularity of a graph, which will be highly regular when the degree is similar for all nodes, and irregular when we can find very different values for different nodes. This work will focus on simple graphs, generally but not always unweighted, and in particular directed graphs. Hypergraphs, which are defined as graphs where there can be multiple edges between any two nodes [5], will not be considered, and even if self-loops may appear (edges e_{ii}), their presence will not affect the purpose of the research.

The following concepts, specific for directed graphs, will appear regularly throughout the work and will be helpful for a better understanding.

Definition 1.7. For directed graphs, we define a **sink** as a node with $d_i^{out} = 0$ and a **source** as a node with $d_i^{in} = 0$.

Definition 1.8. A **path** is a set of nodes such that two nodes are connected if and only if they are consecutive in a list. In directed graphs, all nodes except the ones in the extremes have $d^{in} = d^{out} = 0$ while in the case of undirected graphs, they have $d = 2$.

We define a **sink-path** as a directed set of nodes where the first one can have multiple outgoing and incoming edges and the last one has $d^{out} = 0$. In the same vein, a **source-path** is a directed path where the first node has $d^{in} = 0$ and the last one can have multiple edges.

Definition 1.9. A **cycle** graph can be defined as a path graph where an additional edge is added from the sink node to the source node, having the same number of nodes and edges. For an undirected cycle the degree will be $d = 2$ for all nodes, while for a directed cycle all nodes will have degree $d^{in} = d^{out} = 1$.

Definition 1.10. We refer to a directed set of nodes as a **strongly connected component** if there exists a directed path, in both directions, between any two nodes. Conversely, a **weakly connected component** exists when the corresponding undirected graph is connected.

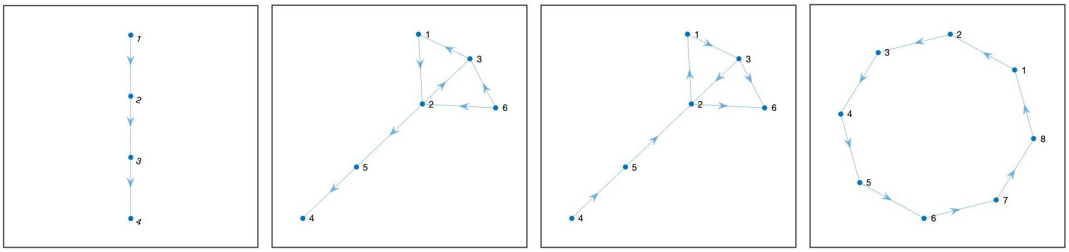
Whenever a graph can be divided into two connected components where there are no edges connecting nodes belonging to two different components, each subgraph can be treated as an independent graph. Therefore, signals on these graphs can be processed separately for each of the subgraphs, since there isn't any influence between them.

1.1.2 Algebraic representation of graphs

Graph operators can be defined by representing graph connections in a matrix form. These matrix representations will allow us to define frequencies for graph signals in Section 1.2 [5].

Definition 1.11. Given a graph $\mathcal{G}$ with N nodes, the **adjacency matrix A** is an $N \times N$ square matrix where the entry a_{ij} will correspond to the weight w_{ij}. This is, the weight of the edge from node i to node j.

From this definition, we can derive that for all graphs, $a_{ii} = 0$ for all i if there are no self-loops, and that for undirected graphs $a_{ij} = a_{ji}$ so that the adjacency matrix is symmetric.



(a) Path graph (b) Sink-path (c) Source-path (d) Cycle

Figure 1.1: Examples of a directed path graph, directed graphs with a sink-path and a source-path, and a cycle of 8 nodes, which can be an example of a regular graph.

Therefore, in the general case of an unweighted directed graph, $\mathbf{A}$ will have 1's in the positions where an edge exists, and 0's in the rest.

This is the most basic matrix representation of a graph and will be the one used on the proposed method in Chapter 3. As an example, the adjacency matrix of the directed path shown in Fig. 1.1a can be written as:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

It is important to remark that the node indexing order is not important, since it will not affect the result of matrix operations. A different labeling of nodes in a graph would only result on a different permutation of the graph representing the same connections, only with different node labels. Therefore, if a simple indexing exists, it can be used without loss of generality [5].

When the adjacency matrix $\mathbf{A}$ is defined, we can build the degree matrix of a graph.

Definition 1.12. The **degree matrix D** of a graph is a diagonal matrix where each term of the diagonal corresponds to the total degree of the corresponding node. For an undirected graph $d_{ii} = \sum_j a_{ij}$, adding together all the terms in the corresponding row of the adjacency matrix. For directed graphs we can separate the in-degree matrix and the out-degree matrix, adding together, respectively, the rows and the columns of the adjacency matrix.

We can represent the node degree of an undirected graph in a matrix form as:

$$\mathbf{D} = \text{diag}(\mathbf{A1}), \tag{1.3}$$

where diag($\mathbf{v}$) builds a square matrix with the elements of $\mathbf{v}$ along its diagonal and the symbol $\mathbf{1}$ corresponds to the vector $[1\ 1\ ...\ 1]^\mathsf{T}$.

In the case of directed graphs, the following equations represent, respectively, the in-degree and out-degree matrices for directed graphs:

$$\mathbf{D}_{out} = \text{diag}(\mathbf{A1}), \tag{1.4}$$

$$\mathbf{D}_{in} = \text{diag}(\mathbf{1}^\mathsf{T}\mathbf{A}) \tag{1.5}$$

Normalization

For undirected graphs, it is possible, and recommended, to normalize the adjacency matrix [31] and [32]. This normalization is performed, respectively, by the following operations:

$$\tilde{\mathbf{A}} = \mathbf{D}^{-1}\mathbf{A} \tag{1.6}$$

or

$$\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \tag{1.7}$$

However, it is not possible to use this normalization on directed graphs, since two different degree matrices exist and it is not always possible to compute the inverse of any of them. Therefore, a different normalization method is needed for directed graph, and an example will be introduced in Chapter 3.

1.2 Signals on Graphs

A graph signal is defined as a vector $\mathbf{x}$ containing scalars corresponding to the values of the signal at each node on the graph. More precisely:

Definition 1.13. A **graph signal** is a real vector $\mathbf{x} \in \mathbb{R}^N$, where the entry $x(i)$ is the real scalar corresponding to the signal associated to node i.

1.2.1 Graph filters and operators

A linear graph filter is represented as a linear operator that gives an output signal $\mathbf{y}$ when applied to a graph signal $\mathbf{x}$. The most basic example of a linear filter is the adjacency matrix, which gives

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{1.8}$$

where the i-th element of the output $\mathbf{y}$ corresponds to the sum of the values of all its neighbors. More specifically, in the case of a directed graph the result of applying this linear filter to a graph signal $\mathbf{x}$ would be, for each node, the sum of all the signals connected to it by directed edges in an incoming direction.

Graph operators

There are other graph operators represented by matrices that, when multiplied by a graph signal, give a new graph signal (the output). Therefore, the adjacency matrix performs a 1-hop diffusion through the graph edges when applied to a graph signal.

Other graph operators can be defined on graph signal processing to carry out different operations to work with graphs. The Graph Laplacian ($\mathbf{L}$), and the Random Walk Laplacian ($\mathcal{T}$) are examples of popular graph operators used in GSP [32]. Both of these common operators are defined from $\mathbf{A}$ and $\mathbf{D}$. However, these operators cannot be used for directed graphs. Even though adapted versions of these operators have been developed in [33], they are not used in this thesis and we focus on the adjacency matrix s the main operator.

All the graph operators mentioned in the previous section are defined as 1-hop operators, which means they can be used as 1-hop filters. From now on we will define $\mathbf{Z}$ as a generic 1-hop operator. It is also important to remark that if $\mathbf{Z}$ is a 1-hop operator, then $\mathbf{Z}^2$ is a 2-hop operator, so $\mathbf{Z}^k$ is a k-hop operator. Therefore, arbitrary polynomials $P(\mathbf{Z})$ of degree k will be localized to the $k-$neighborhood of a node.

Definition 1.14. For a given graph operator $\mathbf{Z}$, a polynomial **graph filter** $P(\mathbf{Z})$ is an $N \times N$ matrix of the form

$$P(\mathbf{Z}) = \sum_{i=0}^{k} a_i \mathbf{Z}^i \tag{1.9}$$

where the terms a_i correspond to the coefficients of the polynomial, and $\mathbf{Z}^0 = \mathbf{I}$

1.2.2 Polynomials and invariant subspaces

Minimal polynomial of a vector

We start by considering $\mathbf{x} \in \mathbb{R}^N$, and apply the operator $\mathbf{Z}$ successively to this vector. This operator could be any of the operators introduced in the previous section. For a given $\mathbf{x}$, we can find a certain p such that

$$[\mathbf{x}, \mathbf{Z}\mathbf{x}, \mathbf{Z}^2\mathbf{x}, ..., \mathbf{Z}^p\mathbf{x}] \tag{1.10}$$

are linearly dependent [5]. For this p, we can express

$$\mathbf{Z}^p\mathbf{x} = \sum_{k=0}^{p-1} a_k \mathbf{Z}^k\mathbf{x} \tag{1.11}$$

and therefore there is a polynomial, named the minimal polynomial, such that

$$P_x(\mathbf{Z})\mathbf{x} = (-\sum_{k=0}^{p-1} a_k \mathbf{Z}^k + \mathbf{Z}^p)\mathbf{x} = 0 \tag{1.12}$$

According to (1.11), $\mathbf{Z}^p$ can be expressed in terms of lower powers of p. This means that a graph filter $P(\mathbf{Z})$, of degree higher than p, provides the same output for $\mathbf{x}$ as a polynomial of degree p or lower. From this, we can express $P(\mathbf{Z})$ as

$$P(\mathbf{Z}) = Q(\mathbf{Z})P_x(\mathbf{Z}) + R(\mathbf{Z}) \tag{1.13}$$

where the degree of $Q(\mathbf{Z})$ and $R(\mathbf{Z})$ is less than that of $P_x(\mathbf{Z})$. Therefore, by definition of $P_x(\mathbf{Z})$,

$$P(\mathbf{Z})\mathbf{x} = R(\mathbf{Z})\mathbf{x} \tag{1.14}$$

Eigenvalues and eigenvectors

Given (1.11), if $\mathbf{u}$ is a vector $\mathbf{x}$ such that $p = 1$, then

$$\mathbf{Z}\mathbf{u} = \lambda\mathbf{u} \tag{1.15}$$

for some scalar λ. Then $\mathbf{u}$ and $\mathbf{Zu}$ are linearly dependent and $\mathbf{u}$ is defined as an eigenvector of $\mathbf{Z}$, where λ is the corresponding eigenvalue. In this case, the minimal polynomial of $\mathbf{Z}$ for the vector $\mathbf{u}$ is

$$P_u(\mathbf{Z}) = \mathbf{Z} - \lambda\mathbf{I} \tag{1.16}$$

and from this we can define the eigensubspace $E_u = \text{span}(\mathbf{u})$, which is invariant to $\mathbf{Z}$.

It is possible for a specific eigenvalue λ to be repeated in the set of eigenvalues of the matrix, and therefore be associated to more than one eigenvector. In this case, we define:

Definition 1.15. The **algebraic multiplicity** m_a of an eigenvalue λ is the number of times it appears in the spectral decomposition of a matrix.

Definition 1.16. The **geometric multiplicity** m_g ($\leq m_a$) of an eigenvalue λ is the number of linearly independent eigenvectors associated to it.

Invariant subspaces

More generally, for a vector $\mathbf{u}$ so that the minimal polynomial $P_u(\mathbf{Z})$ has degree p, we can redefine E_u as the span of

$$\mathbf{u}, \mathbf{Zu}, \mathbf{Z}^2\mathbf{u}, ..., \mathbf{Z}^{p-1}\mathbf{u}$$

which represent p linearly independent vectors. Since $\mathbf{Z}^p\mathbf{u}$ can be rewritten in terms of E_u (according to (1.11)), we have that E_u is invariant to $\mathbf{Z}$ or any polynomial $P(\mathbf{Z})$, and we know that all polynomials applied to signals in E_u can have degree no greater than $p-1$ [5].

1.3 Frequency representation of graph signals

The main idea of this section is that any graph signal can be represented as a weighted sum of elementary signals, each corresponding to a graph frequency, which together form the Graph Fourier Transform (GFT).

1.3.1 Graph Fourier Transform (GFT)

If the matrix $\mathbf{Z}$ of a graph operator is diagonalizable we can find N linearly independent eigenvectors and we can form a basis for $\mathbb{R}^N$ using these eigenvectors. In the case of non-diagonalizable, or defective, matrices it is not possible to obtain N linearly independent eigenvectors, and thus we cannot construct a basis for $\mathbb{R}^N$ with eigenvectors of $\mathbf{Z}$. The main purpose of this research, is to construct a basis for these cases, and will be explained in Chapter 3. Therefore, the following definitions will only apply to diagonalizable matrices $\mathbf{Z}$, but will be valid for both directed and undirected graphs.

For diagonalizable $N \times N$ matrices, N linearly independent eigenvectors exist, each corresponding to a certain eigenvalue λ_i, which can be interpreted as the frequency associated to that vector. More specifically, since it is possible for an eigenvalue to have algebraic multiplicity greater than 1, we say that λ_i is the frequency associated to the subspace E_i, which

is defined as the span of the linearly independent eigenvectors of eigenvalue λ_i. For a diagonalizable matrix $\mathbf{Z}$ we can construct $\mathbf{U}$, an invertible matrix where each column is one of the eigenvectors of $\mathbf{Z}$, and write

$$\mathbf{Z} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1} \tag{1.17}$$

where $\mathbf{\Lambda}$ is the diagonal matrix with the eigenvalues of $\mathbf{Z}$, and $\mathbf{U}$ a matrix containing its eigenvectors.

Definition 1.17. Now we can write any graph signal $\mathbf{x}$ in terms of its graph frequencies:

$$\mathbf{x} = \mathbf{U}\tilde{\mathbf{x}} \tag{1.18}$$

where

$$\tilde{\mathbf{x}} = \mathbf{U}^{-1}\mathbf{x} \tag{1.19}$$

is defined as the **Graph Fourier Transform** (GFT) of the graph signal $\mathbf{x}$ [3].

Note that for undirected graphs, $\mathbf{Z}$ is a symmetric matrix and therefore has a full set of orthogonal vectors, and it holds that

$$\mathbf{U}^{-1} = \mathbf{U}^{\mathsf{T}}$$

so that we can define the GFT for undirected graphs as

$$\tilde{\mathbf{x}} = \mathbf{U}^{\mathsf{T}}\mathbf{x} \tag{1.20}$$

A possible interpretation of the GFT can be seen in Fig. 1.2. The eigenvectors of $\mathbf{U}$ corresponding to the graph frequencies can be visually shown in a graph by assigning to each node the value of the corresponding element of the eigenvectors. Plotting this representation in a color map shows a sense of the variation on the graph of eigenvectors corresponding to each frequency. Variation will usually be lower for the lowest frequencies and higher for increasing values of λ_i.

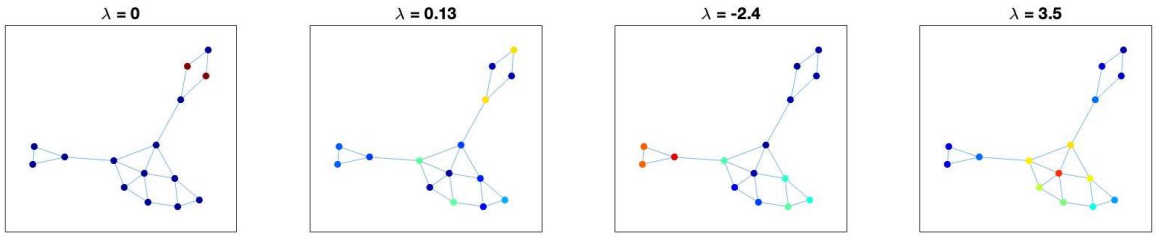


Figure 1.2: Representation of the eigenvectors corresponding to some of the frequencies of an undirected graph. In a color scale from blue to red, blue showing low values and red being a value of ≈ 0.6 The first two images show the result for low frequencies while the last two images correspond to the highest frequencies of the graph.

As described in the introduction, many forms to construct the GFT of an specific graph exist. Firstly, a different version can be created depending on the election of the operator $\mathbf{Z}$. Also, in cases where the operator cannot be diagonalized, or when it can be diagonalized but

some eigenvalues have multiplicity greater than one, we have multiple choices to construct basis.

The most important factor when choosing which operator to use should be the properties of signals to be processed: it will be highly desirable for the elementary bases, or at least some of them, to have a meaningful interpretation.

As we previously mentioned, the presented formulation of a Graph Fourier Transform cannot be used on defective matrices, even though they may appear in many directed graphs as we will see in Chapter 2. For these cases, different methods have been proposed, using mathematical tools such as the Jordan form [17], and ideas from classical signal processing such as wavelet packets [13] [29] (as mentioned in the introduction). Thus, a new method to obtain a spectral decomposition usable on directed graphs is presented in the next chapter.

Chapter 2

The problem: Non-diagonalizable graphs

The first step to obtain the GFT of a signal on a graph is to diagonalize the matrix representation of the corresponding graph operator. Focusing our operations on the adjacency matrix $\mathbf{A}$, we define that:

Definition 2.1. A matrix $\mathbf{A}$ is **diagonalizable** if it is similar to a diagonal matrix. That is, if there exists an invertible matrix $\mathbf{U}$ and a diagonal matrix Λ such that $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$. The matrix $\mathbf{\Lambda}$ contains the eigenvalues of $\mathbf{A}$ in its diagonal, and $\mathbf{U}$ is a matrix whose columns correspond to the eigenvectors of $\mathbf{A}$.

Since $\mathbf{U}$ must be an invertible matrix to compute the GFT of a signal, N linearly independent eigenvectors are needed, so that $\text{rank}(\mathbf{U}) = N$. Therefore, to have a GFT basis for a graph we need the adjacency matrix $\mathbf{A}$ to be diagonalizable.

For a non-diagonalizable graph (also called defective), a complete set of linearly independent eigenvectors does not exist, so a complete basis for $\mathbb{R}^N$ cannot be formed. This means that, even though the $\mathbf{U}$ matrix can be computed, its rank will be $\text{rank}(\mathbf{U}) < N$ and this would make the matrix $\mathbf{U}$ not invertible (or singular). Therefore, the matrix $\mathbf{U}$ can not be used to form the GFT of a signal in graphs with a defective $\mathbf{A}$ matrix.

Many techniques have been used to obtain a decomposition of these graphs in the frequency domain, such as the Jordan canonical form [34], but the main problem for this approach is its numerical instability. To develop a solution to address this diagonalization problem (as the one proposed in chapter 3), it is important to first identify and characterize the properties of graphs having this problem. By definition, it is straightforward that the adjacency matrix $\mathbf{A}$ for an undirected graph is always symmetric, since when an edge exists between nodes i and j we have that $a_{ij} = a_{ji}$.

Theorem 2.1. (Spectral theorem) Every symmetric matrix with real coefficients is orthogonally diagonalizable as $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$ [35].

This means that undirected graphs are, by definition, always diagonalizable. In addition, the corresponding $\mathbf{U}$ matrix will be formed by N orthonormal vectors and the eigenvalues of the matrix will be real [35]. Since it is trivial that every undirected graph will have a GFT matrix they will not be considered in the coming sections. But this property does not always hold for directed graphs. The adjacency matrix of a directed graph is, by definition, asymmetric.

2.1 Diagonalization of directed graphs

Directed graphs, also called digraphs, are present in a wide range of science and technology fields. The topology and edge density of these graphs can vary widely having, for example, graphs with many sinks and sources (the World Wide Web) or extremely irregular graphs (social networks) where node degree may vary from hundreds to millions.

2.1.1 Generating synthetic digraphs

From this point on, random synthetic digraphs will often be generated in order to test various digraph properties. The method used to generate these graphs, called Erdõs-Rényi graphs [36], consists on, for a given number of nodes N, assigning to each pair of nodes an edge with probability p. For simplicity, the resulting graphs will be unweighted and will not have self-loops. Even though this model may not correspond to real world graphs, it is very useful to study connectedness and degree distribution [5]. For this types of graphs we will obtain strongly and weakly connected graphs and, for the second case, graphs with sinks, sources and cycles. This variety will be enough to qualitatively determine the main properties of this types of graphs and especially the one we want to focus on: matrix diagonalization. To give a sense on how likely it is for the adjacency matrix of a directed graph not to be diagonalizable, some statistical results on Erdõs-Rényi graphs, for a variety of sizes (N) and probabilities (p), are presented.

	Edge probability (p)				
	2/N	4/N	6/N	8/N	10/N
N=100	100	79.7	3.7	0	0
N=200	100	98.3	19	0.3	0
N=300	100	99.6	34	5	0
N=400	100	100	41.2	9.7	0

Table 2.1: Percentage of defective adjacency matrices in a set of random graphs of different graph sizes N (1000 graphs for each N) and varying the edge probability p. Note that the edge probability is defined as a function of the number of nodes, in the form $p = k/N$.

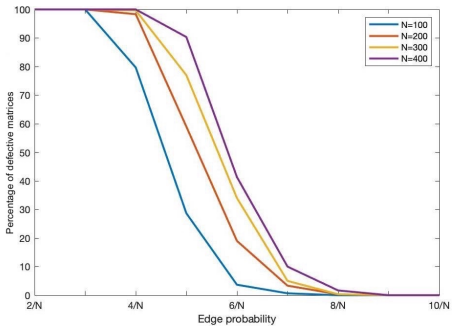


Figure 2.1: Graphic representation of Table 2.1

It can be observed in Fig. 2.1.1 that it is highly probable for a graph adjacency matrix to be defective when the edge density is relatively low, since the percentage of defective graphs obtained is high for small edge probabilities. Considering that Erdõs-Rényi graphs are likely to be regular, this suggests that for real-world graphs it is even more probable to have a defective adjacency matrix.

2.1.2 Strongly and weakly connected graphs

In Chapter 1 the difference between strongly connected graphs (SCG) and weakly connected graphs (WCG) was defined. The first group is formed by directed graphs where at least one path exists, in both directions, between any two nodes of the graph. The most basic example

of a SCG could be a directed cycle, were all nodes can be reached from any other node. On the other hand, the only necessary condition for a graph to be to be weakly connected is that the corresponding undirected graph is connected. This leads to the possibility to have a node with in-degree or out-degree zero leading to the existence of sinks and sources, which, by definition, cannot exist in SCG. However, sources and sinks are not the only elements that prevent graphs from being strongly connected. A graph containing unidirectional connections between two connected components will always be a WCG, since there will not exist any path from the target group to the source one. Algebraically, the clearest way to identify a graph as strongly or weakly connected graph is the (ir)reducibility of its adjacency matrix.

Definition 2.2. We call an $N \times N$ complex square matrix **A reducible** if $N \geq 1$ and there exists a permutation matrix $\mathbf{P}$ such that

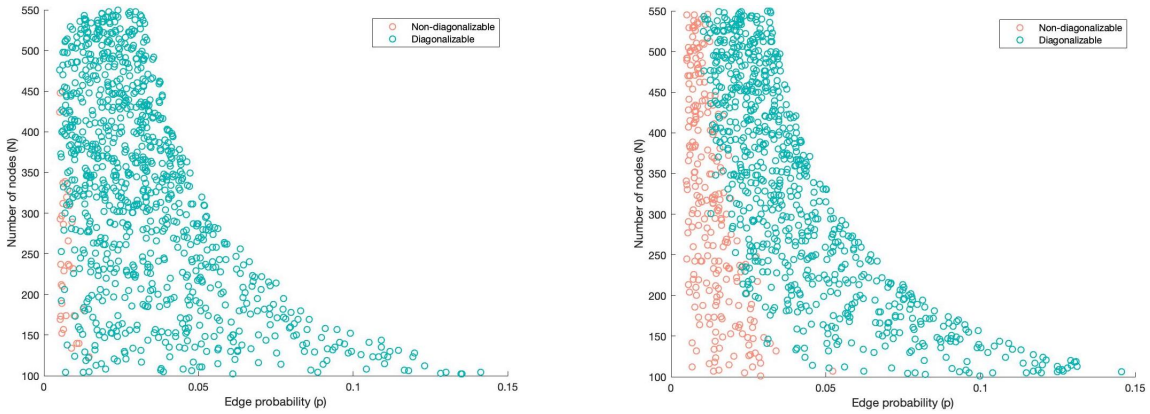
$$\mathbf{P}^{\mathsf{T}}\mathbf{A}\mathbf{P} = \begin{pmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{0} & \mathbf{D} \end{pmatrix} \tag{2.1}$$

where $\mathbf{B}$ and $\mathbf{D}$ are non-empty square matrices. Thus, we call a square matrix **A irreducible** when it does not exist such permutation matrix $\mathbf{P}$ that can put $\mathbf{A}$ in the form of (2.1)

The adjacency matrix $\mathbf{A}$ is always defined as irreducible for SCG and reducible for WCG [37]. By inspection, it can easily be seen that a reducible $\mathbf{A}$ matrix leads to a WCG, if we notice that any connection exists from a node in the lower-left block to a node in the upper-left block (submatrix $\mathbf{C}$) and, consequently, no paths can exist in that direction. However from the experiments shown in if Fig. 2.2 we can state that some irreducible matrices may not be diagonalizable.

Statistical results

Repeating the statistical study performed in Table 2.1 for both SCG and WCG gives some clear results. Creating in each case 1000 Erdõs-Rényi graphs of random sizes between $N = 10$ and $N = 550$ and with random probability $p \in [0.001, 0.2]$, 24% of WCG were not diagonalizable, while only a 3.80% of SCG were non-diagonalizable. This result is enough to state that while SCGs are less likely to be defective, some may be defective.



(a) Strongly connected graphs

(b) Weakly connected graphs

Figure 2.2: Relation between edge probability and graph size for diagonalizable and defective graphs for SCG in a) and WCG in b).

2.2 Properties of directed graphs

Certain graph characteristics can guarantee that a graph is diagonalizable or be defective. These graph properties and characteristics will be described in this section to help building a better classification. It is interesting to identify, for example, which graph elements will result in the existence of a defective matrix, since a possible solution to obtain a diagonalizable equivalent without losing essential graph properties could be to modify the graph avoiding those specific characteristics.

2.2.1 Diagonalizable directed graphs

Remark 2.1. A digraph, either strongly or weakly connected, whose eigenvalues are all different is, by definition, always diagonalizable. However, this property does not imply that graphs where a certain eigenvalue λ has algebraic multiplicity $m_a > 1$ is necessarily defective, since m_a linearly independent eigenvectors can exist and correspond to λ (whenever $m_g = m_a$).

Remark 2.2. Strongly regular digraphs are always diagonalizable (as developed by Godsil, Hobart and Martin in [38]). Recalling the definition of regular graph in Section 1.1.1, a node in a regular graph must satisfy the degree condition: $d^{in} = d^{out}$. This is a very restrictive condition that will not generally be satisfied by real-world graphs.

Remark 2.3. Cyclic graphs are always diagonalizable. In addition in the case of unweighted graphs the set of eigenvalues of the adjacency matrix of a cycle will be the N complex solutions to $\sqrt[N]{1}$.

2.2.2 Defective directed graphs

Remark 2.4. The existence of a source or a sink in a graph is not a reason for it to be defective (as can be seen in Fig. 2.2b, many diagonalizable WCG exist). However, the existence of path-sources or path-sinks (defined in Section 1.1.1), is a sufficient condition for a directed graph to be defective.

Remark 2.5. In fact, adding a source/sink to a diagonalizable graph gives a resulting diagonalizable $\mathbf{A}$ matrix, and adds an eigenvalue zero to the spectrum, but adding a path-source/sink automatically makes it defective, since it adds two zero eigenvalues with linearly dependent eigenvectors.

Directed acyclic graphs

We define a directed acyclic graph as a finite digraph with no directed cycles (Fig. 2.3a). That is, it is not possible from any node of the graph, to find a directed path that leads back to itself. In addition, the adjacency matrix of a DAG will always be nilpotent [39].

Definition 2.3. An $N \times N$ matrix $\mathbf{A}$ is **nilpotent** if and only if $\mathbf{A}^k = \mathbf{0}$ for $k \leq N$ [40].

Note that taking the parameter k in the previous expression, we can state that the length of the longest path of a DAG will be $k - 1$. Its acyclic nature gives DAGs a linear structure where, ultimately, paths only exist in one direction. To build this structure it is necessary for at least one source and one sink to exist in a DAG.

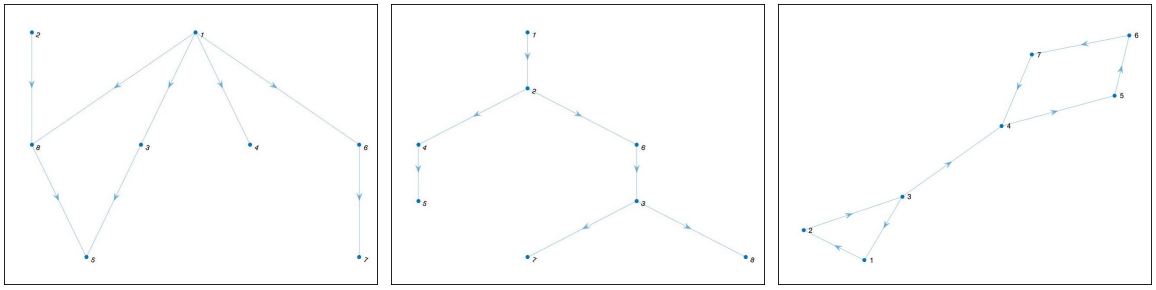
In addition, any nilpotent matrix can be transformed into an upper triangular matrix with zeros along its diagonal. And, by definition, the diagonal parameters of an upper triangular matrix are its eigenvalues. Thus, all the eigenvalues of a DAG's adjacency matrix are zero.

Remark 2.6. The adjacency matrix $\mathbf{A}$ for a DAG is always defective [41]. The eigenvalues of $\mathbf{A}$ are N zeros and there only exist as linearly independent eigenvectors as the number of sources in the graph.

A specific case of a directed acyclic graph are the directed trees, defined as digraphs with only one source and at least one sink, from where all the nodes in the graph can be reached, but where one node can not be reached through two different paths (Fig. 2.3b). From Remark 2.6, we can state that for directed trees all the eigenvalues of the adjacency matrix will be zero and, by definition, only one linearly independent eigenvector will exist. Therefore, they will always be defective.

Remark 2.7. We can derive from the previous statements that path graphs are always defective, since we can think of them as DAGs and, more precisely, as directed trees with one source, one sink and only one path of length N.

Remark 2.8. Weakly connected graphs formed by the unidirectional connection of two (or more) connected components, can be interpreted as groups of nodes connected in one single direction, therefore behaving as DAGs (Fig. 2.3c). Thus, even this group of graphs do not present N zeros as eigenvalues, they are also always defective.



(a) Directed acyclic graph (b) Directed tree (c) Unidirectional connection

Figure 2.3: Examples of some directed graphs mentioned in section 2.2.2.

Chapter 3

Proposed solution: Graph Schur Transform (GST)

In this chapter we present our method to find a spectral decomposition for defective directed graphs. The final result of the proposed algorithm could be an equivalent to the previously mentioned Graph Fourier Transform, for signal processing on the spectral domain. The basis obtained by using our method, where frequencies are grouped into subspaces, shows relevant properties such as invariance, subspace orthogonality and spectral localization.

3.1 Main concepts behind the GST

Before presenting the proposed method, we will briefly introduce the two concepts in which it has been inspired. Firstly, we present the Schur decomposition (Section 3.1.1), an algebraic method to write a complex matrix as unitarily equivalent to an upper triangular matrix whose diagonal elements are the eigenvalues of the original matrix [42]. The second concept is the Diffusion Wavelets formulation (Section 3.1.2), presented in [13], which shows a possible spectral decomposition of a diffusion operator, such as the adjacency matrix of a graph.

3.1.1 Schur Decomposition

Denoting $\mathbf{Z}$ a fundamental graph operator, such as the adjacency matrix, the Schur decomposition of $\mathbf{Z}$ can be obtained even if $\mathbf{Z}$ is not diagonalizable:

$$\mathbf{Z} = \mathbf{U}\mathbf{T}\mathbf{U}^{\mathsf{H}} \tag{3.1}$$

where $\mathbf{U}$ is unitary, $\mathbf{U}^{\mathsf{H}}$ is the Hermitian transpose of $\mathbf{U}$ and $\mathbf{T}$ is an upper triangular matrix with the eigenvalues of $\mathbf{Z}$ along its diagonal.

The Schur decomposition is not unique, a different one is obtained for any given ordering of eigenvalues. In addition, the representation may not be unique if there are eigenvalues with algebraic multiplicity m_a greater than one [5]. We can interpret the $\mathbf{U}$ matrix obtained in a Schur decomposition as follows:

Denote $\mathbf{u}_i$ the i-th column of $\mathbf{U}$ and let $E_i = \text{span}(\mathbf{u}_i)$. Then, because $\mathbf{U}$ is unitary, we

have that $\mathbb{C}^N = \bigoplus_{i=1}^{N} E_i$, i.e., $\mathbb{C}^N$ is the direct sum of the E_i subspaces and it holds that:

$$F_k = \bigoplus_{i=1}^{k} E_i,\ k = 1 \dots N \tag{3.2}$$

are subspaces of $\mathbb{R}^N$ invariant under $\mathbf{Z}$, as will be shown next. That is, if $\mathbf{x} \in F_k$ then $\mathbf{Zx} \in F_k$.

Note that since $\mathbf{U}$ is unitary, we have that $\mathbf{U}^{\mathsf{H}} = \mathbf{U}^{-1}$ so that $\mathbf{ZU} = \mathbf{UT}$. Thus, defining the upper triangular matrix $\mathbf{T}$ as the sum of a diagonal matrix of eigenvalues $\mathbf{D}$ and a nilpotent matrix $\mathbf{N}$ we have that:

$$\mathbf{Z}[\mathbf{u}_1\mathbf{u}_2\ \cdot\cdot\ \mathbf{u}_N] = [\mathbf{u}_1\mathbf{u}_2\ \cdot\cdot\ \mathbf{u}_N](\mathbf{D}+\mathbf{N}) = [\mathbf{u}_1\mathbf{u}_2\ \cdot\cdot\ \mathbf{u}_N] \begin{pmatrix} \lambda_1 & n_{12} & n_{13} & \cdot\cdot \\ 0 & \lambda_2 & n_{23} & \cdot\cdot \\ 0 & 0 & \lambda_3 & \cdot\cdot \\ : & : & : & \cdot \end{pmatrix} \tag{3.3}$$

where $\lambda_1, \dots, \lambda_N$ represent the eigenvalues of the $\mathbf{Z}$ operator and n_{ij} the corresponding element of the $\mathbf{N}$ matrix. Now the multiplication of each of the $\mathbf{u}_i$ by the operator $\mathbf{Z}$ can be expressed as

$$\mathbf{Zu}_1 = \lambda_1\mathbf{u}_1\ \in F_1$$

$$\mathbf{Zu}_2 = n_{12}\mathbf{u}_1 + \lambda_2\mathbf{u}_2\ \in F_2$$

and in general

$$\mathbf{Zu}_i = \sum_{j=1}^{i-1} n_{ji}\mathbf{u}_j + \lambda_i\mathbf{u}_i\ \in F_i \tag{3.4}$$

Therefore, subspaces F_i are invariant but there is an overlap, since $F_{i-1} \subset F_i$.

Denote $\mathbf{v}_i = \mathbf{Zu}_i$ and $P_{E_i}\mathbf{v}_i$ the projection of $\mathbf{v}_i$ into the subspace E_i, which can be described as the vector in the subspace F_i that is closest to $\mathbf{u}_i$ (the vector representing a basis for E_i). We define the approximation error for each F_i as the norm of the difference between the projection of $\mathbf{v}_i$ into the subspace E_i and $\mathbf{v}$. Thus, the error ϵ_i is

$$\epsilon_i = ||P_{E_i}\mathbf{v}_i - \mathbf{v}_i||_2 = \Big|\Big| \sum_{j=1}^{i-1} n_{ji}\mathbf{u}_j \Big|\Big|_2 = \sqrt{n_{1i}^2 + n_{2i}^2 + n_{3i}^2 + \dots} \tag{3.5}$$

Polynomials of S

Assuming that $\mathbf{Z}$ is a 1-hop localized graph operator, such as an adjacency matrix, a polynomial of degree k, $P(\mathbf{Z})$ of $\mathbf{Z}$ would be localized (Section 1.2.1). We now discuss how this operation can be interpreted in terms of the Schur decomposition.

First, note that we only need to study the behavior of $\mathbf{T}$, since

$$\mathbf{Z}^k = \mathbf{UT}^k\mathbf{U}^{\mathsf{H}}. \tag{3.6}$$

Rewriting again $\mathbf{T} = \mathbf{D} + \mathbf{N}$, we have that $\mathbf{N}$ is a nilpotent matrix so that $\mathbf{N}^m = \mathbf{0}$ for some integer $m \leq N$ where N is the size of the $\mathbf{Z}$ operator[40]. Next, we define $P(\mathbf{T})$, a

polynomial of $\mathbf{T}$ in terms of $\mathbf{D}$ and $\mathbf{N}$ where $P(\cdot)$ has degree k. Then, we can use the Taylor series expansion to write:

$$P(\mathbf{D}+\mathbf{N}) = P(\mathbf{D}) + P^{(1)}(\mathbf{D})\mathbf{N} + \frac{1}{2!}P^{(2)}(\mathbf{D})\mathbf{N}^2 + \frac{1}{3!}P^{(3)}(\mathbf{D})\mathbf{N}^3 + \dots \tag{3.7}$$

where $P^{(i)}(\mathbf{D})$ denotes the i-th derivative of P. This allows us to represent $P(\mathbf{Z})$ in the spectral domain:

$$P(\mathbf{Z}) = \mathbf{U}P(\mathbf{D}+\mathbf{N})\mathbf{U}^{\mathsf{H}} = \mathbf{U}P(\mathbf{D})\mathbf{U}^{\mathsf{H}} + \mathbf{U}P^{(1)}(\mathbf{D})\mathbf{N}\mathbf{U}^{\mathsf{H}} + \frac{1}{2!}\mathbf{U}P^{(2)}(\mathbf{D})\mathbf{N}^2\mathbf{U}^{\mathsf{H}} + \dots \tag{3.8}$$

Note that $\mathbf{N}$ is strictly upper triangular so that the first column of $\mathbf{N}$ is $\mathbf{0}$, the first two columns of $\mathbf{N}^2$ are $\mathbf{0}$ and $\mathbf{N}$ is nilpotent, so that for some m we will have that $\mathbf{N}^m = \mathbf{0}$. This allows us to express the output of $P(\mathbf{Z})$ for each of the orthogonal basis vectors in $\mathbf{U}$.

Note that $\mathbf{U}^{\mathsf{H}}\mathbf{u}_i = \mathbf{e}_i$, the i-th canonical basis vector, which produces the $i-th$ column of a matrix that multiplies it. Thus, based on properties of $\mathbf{N}$ we have that:

$$\mathbf{N}^m\mathbf{e}_1 = \mathbf{0}, \ \ \forall m \geq 1$$

and in general

$$\mathbf{N}^m\mathbf{e}_i = \mathbf{0}, \ \ \forall m \geq i \tag{3.9}$$

Based on this expression, the output when $\mathbf{u}_1$ is the input depends only on $P(\lambda_1)$ that is:

$$P(\mathbf{Z})\mathbf{u}_1 = P(\lambda_1)\mathbf{u}_1.$$

Similarly multiplication by $\mathbf{u}_2$ can be written as:

$$P(\mathbf{Z})\mathbf{u}_2 = P(\lambda_2)\mathbf{u}_2 + P^{(1)}(\lambda_1)n_{12}\mathbf{u}_1,$$

where n_{12} is the entry at the first row, second column of $\mathbf{N}$. And following a similar argument we can write:

$$P(\mathbf{Z})\mathbf{u}_3 = P(\lambda_3)\mathbf{u}_3 + P^{(1)}(\lambda_2)n_{23}\mathbf{u}_2 + P^{(1)}(\lambda_1)n_{13}\mathbf{u}_1 + P^{(2)}(\lambda_1)n_{13}^{(2)}\mathbf{u}_1,$$

where n_{13} and n_{23} are non zero values on the 3rd column of $\mathbf{N}$ while $n_{13}^{(2)}$ in the third column of $\mathbf{N}^2$. Therefore, for $\mathbf{u}_i$ all the terms including powers of $\mathbf{N}$ greater than $i-1$ will be zero. From this we can see that, for any column vector $\mathbf{u}_i$ of the matrix $\mathbf{U}$ obtained in the Schur decomposition of $\mathbf{Z}$ the multiplication of any vector $\mathbf{u}_i$ by any polynomial $P(\mathbf{Z})$ is also invariant, with resepect to $\mathbf{Z}$ to the subspace F_i, since the result will be a linear combination of the vectors that form the basis for F_i.

3.1.2 Diffusion Wavelets

The diffusion wavelets (DW) design [13] uses an approach that combines both spectral and vertex domain characteristics, but without providing exact localization in either of these domains [5]. The key observation in the DW design is that successive powers of the diffusion operator $\mathbf{Z}$ will have increasingly lower numerical rank. This is because when an operator $\mathbf{Z}$ is normalized so that the magnitude of the eigenvalues ranges from 0 to 1, the eigenvalues of $\mathbf{Z}^k$ are λ_i^k and these can become arbitrarily small as k increases. This leads to the definition of ϵ-span of a set of vectors.

Definition 3.1. Consider a set of vectors

$$\Phi_v = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\}$$

which could be for example the columns of $\mathbf{Z}^k$. Then, define a set of vectors

$$\Phi_u = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_j\}$$

with $j \leq N$. Then Φ_u ϵ-spans Φ_v if for all $i = 1, \ldots, N$:

$$||\mathbf{P}_{\Phi_u}\mathbf{v}_i - \mathbf{v}_i||_2 \leq \epsilon \tag{3.10}$$

where $\mathbf{P}_{\Phi_u}$ computes the projection of a vector onto the span of Φ_u.

Intuitively if Φ_u ϵ-span Φ_v with $j < N$ this means that not much of an error is made by approximating the span with a smaller set of vectors. In the design of diffusion wavelets, this idea is used by selecting sets:

$$\Phi_{\mathbf{Z}^{2^i}} = \{\lambda_{s,1}^{2^i}\mathbf{v}_1, \ldots \lambda_{s,b}^{2^i}\mathbf{v}_N\} \tag{3.11}$$

where $\{\mathbf{v}_1, \ldots \mathbf{v}_N\}$ are the eigenvectors of $\mathbf{Z}$. Each of these sets corresponds to the eigenvalues of the matrices obtained from consecutive dyadic powers of the operator $\mathbf{Z}$ such that $\lambda < \epsilon$. Then $V_0 = \mathbb{R}^N$ and $V_i = \text{span}(\Phi_i)$, where Φ_i ϵ-spans $\Phi_{\mathbf{Z}^{2^i}}$. At any stage we then find a subspace W_i such that

$$V_i \oplus W_i = V_{i-1}. \tag{3.12}$$

Then choosing a specific i we write

$$V_0 = V_i \oplus W_i \oplus W_{i-1} \oplus \ldots \oplus W_1 \tag{3.13}$$

so that the basis for the W_i spaces form the orthogonal wavelets and the basis for V_i correspond to the scaling function.

From this design we can finally obtain $M + 1$ orthogonal subspaces formed by orthogonal vectors and corresponding to the eigenvalues of $\mathbf{Z}$ in a increasing order:

$$\{W_1,\ W_2,\ \ldots\ , W_M,\ V_M\} \tag{3.14}$$

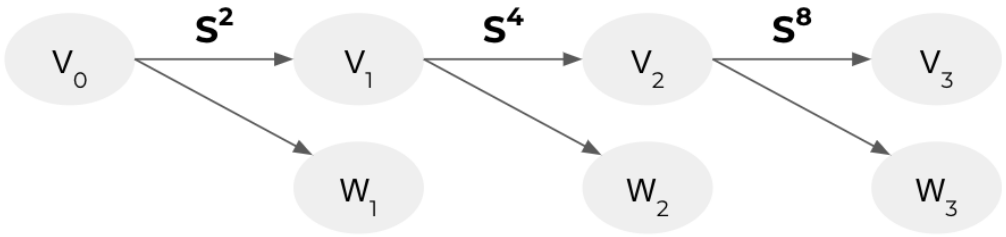


Figure 3.1: Diagram of how subspaces are created by applying dyadic powers of $\mathbf{Z}$

The procedure followed is shown in Algorithm 1.

Algorithm 1 Diffusion Wavelets

Input:
A: Adjacency matrix
M: Expected number of subspaces
ϵ: precision of the algorithm
Ouput:
V_0: Matrix with the complete basis
W: Cell of arrays storing the basis for the subspaces, in the order $W_1, W_2, ..., V_1$
E: Cell of M vectors storing the eigenvalues corresponding to each subspace

1: $\mathbf{D} \leftarrow$ Spectral decomposition of $\mathbf{A}$
2: $\tilde{\mathbf{A}}, \tilde{\mathbf{D}} \leftarrow$ Divide $\mathbf{A}$ and $\mathbf{D}$ by max($\mathbf{D}$)) to rescale the range of eigenvalues to be [0,1]
3: Order eigenvalues increasingly
4: Tree (struct) $\leftarrow$ `DWPTree`$(\tilde{\mathbf{A}}, M, \epsilon)$[1]
5: **for** each tree-level j **do**
6: $W_j \leftarrow$ Store the set of vectors forming the basis of subspace W_j obtained in step 4
7: $\mathbf{E}_j \leftarrow$ Store the range of eigenvalues corresponding to the subspace j
8: $\mathbf{V}_0 \leftarrow$ Update the final basis
9: **end for**
10: $W_{j+1} \leftarrow$ Store the basis corresponding to the subspace $V_k(= W_k + 1)$
11: $\mathbf{V}_0 \leftarrow$ Add $V_k(= W_k + 1)$ to the final basis

This approach can be very useful in several applications, as proposed in [13] and [29]. However, this design is not very flexible (due to its dyadic structure), and thereforer not so adaptative to actual eigenvalues. This is why in the next section we propose a new method, with the intention to offer a slightly different outcome, specifically in relation to the invariance of its subspaces or the regularity on the defined distribution (properties developed in further detail in chapter 4).

3.2 Wavelets derived from the Schur Decomposition

Both, Diffusion Wavelets and Schur Decomposition decompose the space $V_0 = \mathbb{C}^N$ into the direct sum of orthogonal subspaces. To understand better the connection let us consider them again:

– For the Schur Decomposition $\mathbb{C}^N = \bigoplus_{i=1}^{N} E_i = E_1 \oplus E_2 \oplus ... \oplus E_N$, where the subspaces $E_i = \text{span}(\mathbf{u}_i)$ are orthogonal, but they are not invariant. Instead, we have that $F_k = \bigoplus_{i=1}^{k} E_i$ is invariant, but clearly the F_k spaces have overlaps and are not orthogonal.

[1]The `DWTree` function used has been obtained from the Diffusion Wavelets Toolbox presented in [13] and can be found in https://github.com/aweinstein/dw. From the many outputs of the Diffusion Wavelets Tree, we will use the basis of vectors formed in each case, the *ExtBasis* variable assigned to each subspace W_k.

– In the Diffusion Wavelets approximation, for a sufficiently large i, V_i contains only signals that are in the subspace corresponding to the largest eigenvalue, while W_1 corresponds to the smaller eigenvalues, and for increasing levels the subspace W_i corresponds to higher eigenvalues. Then we can define the ambient space V_0 as:

$$V_0 = V_i \oplus W_i \oplus W_{i-1} \oplus W_{i-2} \oplus \ldots \oplus W_1 \tag{3.15}$$

Also for this method subspaces have orthogonal basis of vectors, and are always orthogonal to each other, but they are not exactly invariant to multiplication by the graph operator.

The initial connection between the two is that, if the eigenvalues are ordered in decreasing order and i is sufficiently large, then $F_1 = E_1 = V_i$.

Therefore, a strong relation between both methods may be possible, and a decomposition of $\mathbf{A}$ into subspaces can be constructed based on the Schur Decomposition. This new approach can have the following advantages compared with the DW method:

1. Instead of an approximate invariance (as the one that gives the Diffusion Wavelets approximation for vectors in the W_1 subspaces), the proposed method ensures that vectors within the computed basis are exactly invariant to the subspace they belong to.

2. Invariant subspaces can be built in a more regular and flexible way depending on the purpose of the study, in contrast to the arbitrary grouping of eigenvalues that is obtained with the Diffusion Wavelets method.

3.2.1 Graph Schur Transform (GST)

Our proposed method consists on the generation of a set of subspaces based on the iterative use of the Schur Decomposition. The output of this method is a group of M subspaces $U_1, U_2, ...$ with respective basis that are orthonormal, and such that subspaces are invariant.

Proposed idea

For a given graph, we start by normalizing its adjacency matrix $\mathbf{A}$. Since the conventional normalization described in Section 1.1.2 is not possible for all kinds of adjacency matrices, and our main purpose is to have a normalized set of eigenvalues with magnitudes limited in the range $[0, 1]$, we proceed by dividing the matrix by its largest eigenvalue obtaining the normalized (or scaled) matrix $\tilde{\mathbf{A}}$.

Using the notation from Section 3.1.1, we denote a series of spaces F_k, such that $F_1 \subset F_2 \subset \ldots \subset F_k$ where F_1 contains the basis from Schur decomposition associated to eigenvalues λ_1 to λ_{i_1}, F_2 contains the basis λ_1 to $\lambda_{i_1+i_2}$ and F_k contains λ_1 to $\lambda_{i_1+\ldots+i_k}$, where $i_1, i_2, ...$ represent the dimension of the subspace formed by the vectors added in the last iteration. Thus, we can see the spaces are embedded in each other.

At the same time, we will introduce a series of subspaces $G_1, ..., G_k$, such that

$$G_1 = F_1 \text{ and } G_1 \oplus G_2 = F_2. \tag{3.16}$$

so that, in general,

$$G_{i-1} \oplus G_i = F_i. \tag{3.17}$$

The spaces G_i are defined as follows. G_1 contains the basis in Schur associated to eigenvalues λ_1 to λ_{i_1}, but G_2 contains a basis corresponding to the eigenvalues from λ_{i_1+1} to $\lambda_{i_1+i_2}$ (that will be renamed $\lambda_1^{(2)}$ to $\lambda_{i_2}^{(2)}$). Then to form a complete basis for $\mathbb{R}^N$ we can represent this space as:

$$G_1 \oplus G_2 \ldots \oplus G_k = F_k = \mathbb{R}^N \tag{3.18}$$

where F_k corresponds to the last subspace of dimension N.

To find the most appropiate criteria to apply to separate the graph frequencies into subspaces, many options have been considered, looking for the one whose output offers better properties on subspace invariance and orthogonality. The reason why the procedure shown below has been chosen will be developed in further detail in the next chapter.

Building the subspaces

The next step is to define the subspace G_1. To start we find the Schur decomposition of the adjacency matrix $\mathbf{A}$ by ordering the eigenvalues from smallest (in magnitude) to largest. This will give us the Schur matrix $\mathbf{T}$ and the transformation matrix $\mathbf{U}$. This subspace will correspond to the lowest energies of the graph, so we can define it as:

$$G_1 = \text{span}(u_1, ..., u_{i_1}), \tag{3.19}$$

where $\mathbf{u}_1, \mathbf{u}_2, ...$ correspond to the first columns of the matrix $\mathbf{U}$ and $\mathbf{u}_{i_1}$ the column corresponding to the last eigenvalue included in the first subspace. Therefore, the basis for this subspace will be

$$\mathbf{U}_1 = [\mathbf{u}_1 \ \mathbf{u}_2 \ ... \ \mathbf{u}_{i_1}] \tag{3.20}$$

Due to the orthogonal nature of the matrix $\mathbf{U}$ obtained in the Schur decomposition, we have obtained an orthogonal invariant basis for the vectors in G_1.

To build the next subspace, we start by reordering the eigenvalues as

$$\{\lambda_{i_1+1}, \ \lambda_{i_1+2}, \ \ldots, \lambda_1, \lambda_2, \ \ldots, \lambda_{i_1}\}$$

and renaming the eigenvalues as following:

$$\lambda_{i_1+1} = \lambda_1^{(2)} \ ,$$

$$\lambda_1 = \lambda_{N-i_1+1}^{(2)} \ ,$$

$$\lambda_{i_1} = \lambda_N^{(2)}.$$

Now we can rebuild the Schur matrix in the new order:

$$\mathbf{T}^{(2)} = \begin{pmatrix} \lambda_1^{(2)} & n_{12}^{(2)} & \cdot & n_{1N}^{(2)} \\ 0 & \lambda_2^{(2)} & \cdot & n_{2N}^{(2)} \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \lambda_N^{(2)} \end{pmatrix} = \mathbf{U}^{(2)\mathsf{T}} \mathbf{Z} \mathbf{U}^{(2)} \tag{3.21}$$

and repeating the procedure used for G_1, we can build a basis for the space G_2 taking the first i_2 columns of the matrix $\mathbf{U}^{(2)}$. The first column will be an eigenvector for $\lambda_1^{(2)}$ and the other $i_2 - 1$ vectors will be invariant to the subspace they form. With these vectors we will have an orthogonal invariant basis, G_2, which will form the orthogonal matrix $\mathbf{U}_2$:

$$\mathbf{U}_2 = [\mathbf{u}_1^{(2)}\ \mathbf{u}_2^{(2)}\ ...\ \mathbf{u}_{i_2}^{(2)}]$$

At this point we have built 2 different orthogonal and invariant basis for two sets of energies of the graph: $\lambda_1, ..., \lambda_i$ and $\lambda_1^{(2)}, ..., \lambda_{i_2}^{(2)}$. Repeating this procedure as many times as necessary will result in a set of k invariant subspaces $G_1, ..., G_k$ corresponding to the k groups of energies of the graph, where

$$G_1 \oplus ... \oplus G_i = F_i, \tag{3.22}$$

$$F_i \oplus G_{i+1} = F_{i+1} \text{ and } F_k = \mathbb{R}^N \tag{3.23}$$

Finally, we can define the Graph Schur Transform $\mathbf{U}$ as the following squared NxN matrix:

$$\mathbf{U} = [\mathbf{U}_1\ \mathbf{U}_2\ \mathbf{U}_3\ ...\ \mathbf{U}_k] \tag{3.24}$$

In order to decide how we will group the normalized eigenvalues, we start by ordering them increasingly, from zero to one. In the case of complex eigenvalues, we will order by their magnitude ($|\lambda|$). The criteria followed to find the eigenvalues $\lambda_{i_1}, \lambda_{i_2}, ...$ which will represent the higher limit for each subspace is to, given the desired number of subspaces k, find the $k-1$ points where the distance between consecutive eigenvalues is greater. This configuration computes a set of subspaces G_i invariant to $\tilde{\mathbf{A}}$ build with unitary and orthogoanl vectors. Therefore, all the basis for $G_1, ..., G_k$ are orthonormal and invariant. However, vectors from different subspaces may not be orthogonal to each other, so the final basis $\mathbf{U}$, which will represent the GST of the graph, is not an orthogonal basis. This property will be further explained in Section 4.1.2.

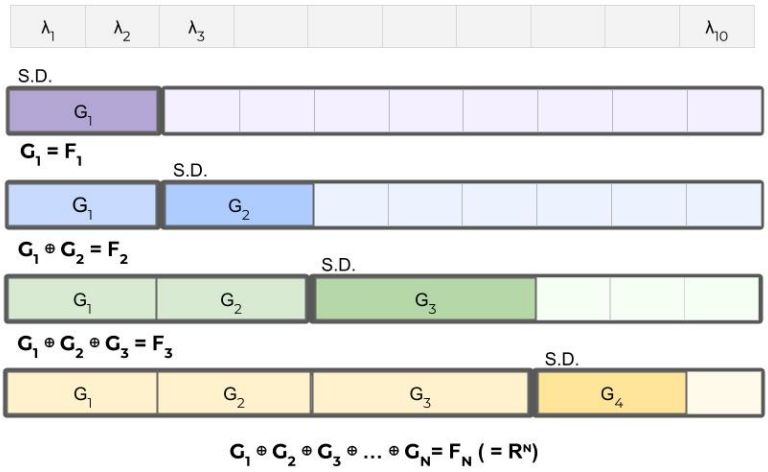


Figure 3.2: Diagram showing how subspaces are created in the GST method for an example with N=10 where λ_1, λ_2, etc represent the normalized eigenvalues.

As we already mentioned, this approach offers an output that can be preferable for some applications. Its main properties and benefits are presented in the next chapter, and compared with the results obtained on the Diffusion Wavelets approach to observe the strengths of each method.

The procedure followed to computationally build the basis of the GST method is described in Algorithm 2. We include, in the computation of the GST, the creation of a polynomial filter for each subspace. The expression of this polynomial for the *k-th* group of eigenvalues $\lambda_{k,1}, \lambda_{k,2}, ...$ (subspace k) we define:

$$P_k(\mathbf{A}) = (\mathbf{A} - \lambda_{k,1}\mathbf{I})(\mathbf{A} - \lambda_{k,2}\mathbf{I})...$$

where we have that $P_k(\mathbf{A})\mathbf{x} = 0$ for those $\mathbf{x}$ such that $\mathbf{x} \in \text{span}(\mathbf{v}_{k,1}, \mathbf{v}_{k,2}, ...)$, so that $\mathbf{x}$ is invariant to multiplication by its corresponding polynomial $P_k(\mathbf{A})$ (We will study this property in further detail in section 4.1.1).

Algorithm 2 Graph Schur Transform

Input:
$\mathbf{A}$: Adjacency matrix
M: Expected number of subspaces
Ouput:
$\mathbf{U}_f$: Matrix with the complete basis
U: Cell of M arrays storing the basis for the M subspaces
$\mathbf{E}$: Cell of M vectors storing the eigenvalues corresponding to each subspace
P: Cell of M arrays storing the polynomials corresponding to each subspace

1: $\mathbf{D} \leftarrow$ Absolute value of the eigenvalues of $\mathbf{A}$
2: $\tilde{\mathbf{A}}, \tilde{\mathbf{D}} \leftarrow$ Divide $\mathbf{A}$ and $\mathbf{D}$ by $\max(\mathbf{D})$ to rescale the range of eigenvalues of $\mathbf{A}$ so that $|\lambda_i| \in [0, 1] \; \forall \; \lambda_i$.
3: Order eigenvalues increasingly
4: $\mathbf{B}$, $\mathbf{K} \leftarrow$ Calculate the distance between consecutive eigenvalues and find the M-1 greater separation points and their position.
5: $\mathbf{U}_c, \mathbf{T}_c \leftarrow$ Ordered Schur Decomposition of $\tilde{\mathbf{A}}$
6: **for** each subspace j **do**
7: $\quad k \leftarrow$ Find the corresponding range of eigenvalues
8: $\quad U_j \leftarrow [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k]$ Create the basis for subspace j with the first k columns of $\mathbf{U}_c$
9: $\quad \mathbf{E}_j \leftarrow$ Store the range of eigenvalues corresponding to the subspace j
10: $\quad \mathbf{U}_f \leftarrow$ Update the final basis.
11: $\quad P_j(\mathbf{A}) \leftarrow$ Build the polynomial $P_j(\mathbf{A})$
12: $\quad \mathbf{U}_c, \mathbf{T}_c \leftarrow$ Rebuild the Schur Decomposition of $\tilde{\mathbf{A}}$, reordering the eigenvalues
13: **end for**

Chapter 4

Properties of the Graph Schur Transform

4.1 Comparing the GST method with Diffusion Wavelets

Basis built both with the Graph Schur Transform (GST) method and the Diffusion Wavelets (DW) have invariance and orthogonality properties. For example, in both methods subspaces are formed by orthonormal vectors. In this section we are going to go over the most interesting properties of the developed method (GST), and compare it with the DW idea.

4.1.1 Invariance

While the basis built in DW present an approximated invariance, the proposed method shows that subspaces are exactly invariant. A detailed explanation of the property that holds in each case is given below.

Invariant subspaces for dyadic powers of A in DW

Define $\mathbf{x}$ as a vector from the subspace W_1, the orthogonal complement of the first scaling function, built in the first iteration and corresponding to eigenvalues of $\mathbf{A}$ such that $\lambda < \epsilon$. Then it holds that

$$\tilde{\mathbf{A}}\mathbf{x} \approx 0$$

For a vector $\mathbf{x} \in W_2$, we have that $\tilde{\mathbf{A}}^2\mathbf{x} \approx 0$, for $\mathbf{x} \in W_3$, $\tilde{\mathbf{A}}^4\mathbf{x} \approx 0$, etc. An in general,

$$\text{For } \mathbf{x} \in W_i, \quad \tilde{\mathbf{A}}^{2^{i-1}}\mathbf{x} \approx 0$$

So subspaces in this method are approximately invariant.

Polynomials and invariant subspaces from the GST

Consider an eigenvalue λ_i such that $\mathbf{A}\mathbf{v}_i = \lambda_i\mathbf{v}_i$ where $\mathbf{v}_i$ is the corresponding eigenvector. Then define $P_i(\mathbf{A}) = (\mathbf{A} - \lambda_i\mathbf{I})$, so that $P_i(\mathbf{A})\mathbf{v}_i = 0$. Let $P(\mathbf{A})$ be a filter such that, as introduced in Chapter 1,

$$P(\mathbf{A}) = Q_i(\mathbf{A})P_i(\mathbf{A}) + R_i(\mathbf{A})$$

where the residue $R_i(\mathbf{A}) = 0$ if $P(\mathbf{A})$ can be divided by $P_i(\mathbf{A})$.

Then

$$P(\mathbf{A})\mathbf{v}_i = R_i(\mathbf{A})\mathbf{v}_i.$$

Let us consider the case where we have several eigenvalues grouped together. In this case for the *k-th* group of eigenvalues $\lambda_{k,1}, \lambda_{k,2}, ...$ we define a polynomial

$$P_k(\mathbf{A}) = (\mathbf{A} - \lambda_{k,1}\mathbf{I})(\mathbf{A} - \lambda_{k,2}\mathbf{I})...$$

In this case, we have that $P_k(\mathbf{A})\mathbf{x} = 0$ for those $\mathbf{x}$ such that $\mathbf{x} \in \text{span}(\mathbf{v}_{k,1}, \mathbf{v}_{k,2}, ...)$.

Another way to see the invariance of the vectors of the $k-$th basis is as following. As explained in Section 3.1.1 we have, by definition of the construction of the Schur decomposition, that for $\mathbf{x} \in F_k$, $\mathbf{Ax} \in F_k$, where F_k would correspond to the subspace formed by the k first columns of the transformation matrix build with the Schur decomposition. Therefore, using the notation presented in Section 3.1.1,

$$\mathbf{A}(\alpha_1\mathbf{v}_{k,1} + \alpha_2\mathbf{v}_{k,2} + ...) = \alpha_1\lambda_1\mathbf{v}_{k,1} + \alpha_2\lambda_2\mathbf{v}_{k,2} + \alpha_2 n_1\mathbf{v}_{k,1} + ... = \beta_1\mathbf{v}_{k,1} + \beta_2\mathbf{v}_{k,2} + ...$$

so we have invariance for F_k.

Design of filters using the GST basis

For the $k-$th subspace we can design a filter of the form

$$P(\mathbf{A}) = Q_k(\mathbf{A})P_k(\mathbf{A}) + R_k(\mathbf{A})$$

and for any vector $\mathbf{x} \in \text{span}(\mathbf{v}_{k,1}, \mathbf{v}_{k,2}, ...)$ we have that

$$P(\mathbf{A})\mathbf{x} = R_k(\mathbf{A})\mathbf{x}$$

By designing filters that assign the same gain (a scalar value γ_k) to all points in the $\text{span}(\mathbf{v}_{k,1}, \mathbf{v}_{k,2}, ...)$ we have that the filters we design have the form:

$$P(\mathbf{A}) = Q_k(\mathbf{A})P_k(\mathbf{A}) + \gamma_k\mathbf{I} \quad \forall k$$

Therefore, for any $\mathbf{x} \in \text{span}(\mathbf{v}_{k,1}, \mathbf{v}_{k,2}, ...)$,

$$P(\mathbf{A})\mathbf{x} = \gamma_k\mathbf{x}$$

4.1.2 Orthogonality

As already mentioned, in both methods subspaces are built with orthonormal basis of vectors, so that for the GST we have that $\mathbf{U}_i^\mathsf{T}\mathbf{U}_i = \mathbf{I}$, and for DW it holds that $\mathbf{W}_i^\mathsf{T}\mathbf{W}_i = \mathbf{I}$.

Another important aspect of the method is the orthogonality between different subspaces. For this property, while DW shows exact orthogonality, the DW method does not present exactly orthogonal basis. Below are detailed the main orthogonality properties for each method.

Orthogonality for basis in DW

In this construction subspaces are orthogonal to each other by definition. We have that $V_0 = V_1 \oplus W_1$ and, decomposing V_1, $V_0 = V_2 \oplus W_2 \oplus W_1$. Since $W_2 \in V_1$, and W_1 is the orthogonal complement of V_1, W_2 is necessarily orthogonal to W_1. The same idea applies for the rest of subspaces, so we finally get

$$V_0 = V_k \oplus W_k \oplus W_{i-1} \oplus W_{i-2} \oplus \ldots \oplus W_1$$

where all the subspaces W_i are orthogonal to each other, and all of them are also orthogonal to the subspace V_k. Therefore, it holds that $W_i^\mathsf{T}W_j = \mathbf{0}$, and also that $V_k^\mathsf{T}W_i = \mathbf{0}$

Approximate orthogonality for basis in the GST

In this case, we chose invariance over orthogonality, so that $\mathbf{U}_i^\mathsf{T}\mathbf{U}_j = \mathbf{0}$ does not necessarily hold. However, we can obtain basis close to satisfy this orthogonality.

To measure the orthogonality between two basis obtained with the GST, the procedure will be to calculate both the mean value (μ) of the scalar product between any two vectors from different basis, and calculate also the maximum value (m) that can be obtained from this scalar product for a certain graph. To mathematically define this parameters we start by defining the matrix of inner products $\mathbf{B}$:

$$\mathbf{B} = \mathbf{U}^\mathsf{T}\mathbf{U} \tag{4.1}$$

More precisely, since all the elements of the diagonal of $\mathbf{B}$ correspond to the product of $\mathbf{u}_i^\mathsf{T}\mathbf{u}_i = 1$, we adjust $\mathbf{B}$ so that $\tilde{\mathbf{B}} = \mathbf{B} - \mathbf{I}$.

Each element b_{ij} of $\tilde{\mathbf{B}}$ corresponds to the inner product between vectors $\mathbf{u}_i$ and $\mathbf{u}_j$. To calculate the mean μ of the values of this matrix that correspond to the inner product of vectors from different basis, we define n as the difference between the total number of products and the number corresponding to the product between vectors in the same subspace, so that $n = N^2 - \sum_k {}_k$, where ${}_k$ is the dimension of subspace k. Now we can define the mean of the inner product as:

$$\mu = \frac{\sum_i \sum_j |b_{ij}|}{n} \tag{4.2}$$

We calculate the maximum value m that the inner product can take, as following:

$$m = \max_{i,j \in [1,N]} |b_{ij}| \tag{4.3}$$

From the calculation of these parameters for different types of graphs, the following results were obtained:

- The m value generally corresponds to the scalar product of two vectors in consecutive subspaces, usually in a range of eigenvalues close to zero. This property can be observed in Fig. 4.1.

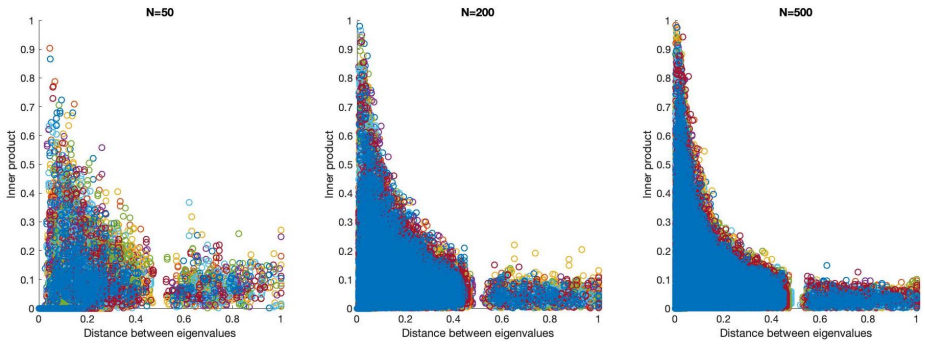


Figure 4.1: Scatter plots representing the relation between the result of the inner product between any two vectors from the final basis $\mathbf{U}_f$ of the GST, and the distance between the corresponding eigenvalues. The test has been performed on 15 random synthetic graphs of 50, 200 and 500 nodes respectively, with a number of subspaces of N/10.

- For a certain graph, by decreasing the number of subspaces created, we get a smaller μ and, even m does not necessarily change, when it does it also decreases.

- Increasing the number of nodes and maintaining the ratio of subspaces (for example, as N/5 for each graph), the m value tends to increase, while μ decreases.

In the histograms on Fig. 4.2 these properties can be observed. It can also be easily seen that the number of pairs of vectors presenting an inner product greater than 0.2 is practically negligible in all cases.

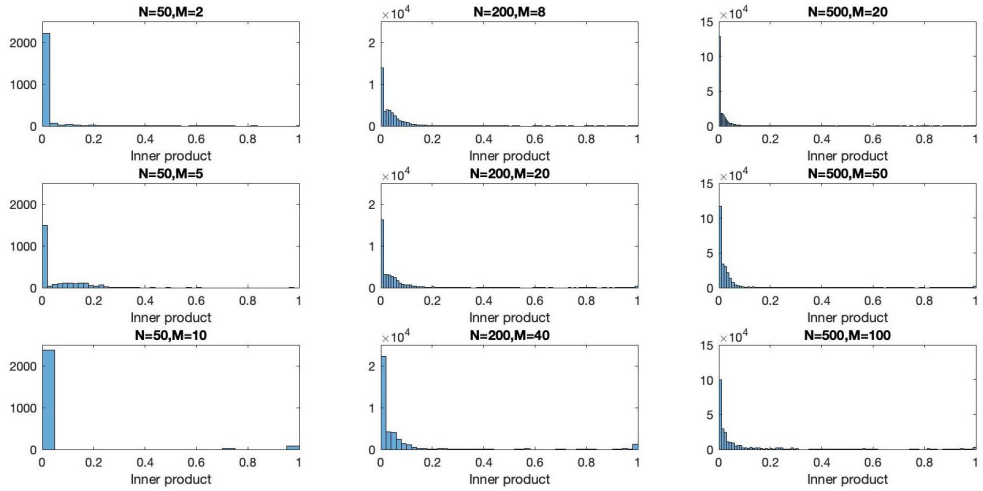


Figure 4.2: Histogram representing the result of the inner product between all the possible combinations of two vectors from different subspaces from the final basis $\mathbf{U}_f$ of the GST. The test has been performed on random synthetic graphs of 50, 200 and 500 nodes. For each size, three different distributions have been tested, for a number of subspaces equal to N/25, N/10 and N/5.

4.1.3 Spectral Localization

The proposed method offers, compared to DW, a more regular and localized distribution of frequencies into subspaces. We can see this property from different perspectives. For example when building the GST we can decide the exact number of basis created, so that the dimensions of subspaces can be adjusted according to the purpose of the processing needed and also separated in ranges with similar magnitude. This characteristic not only gives more flexibility to our method compared to others, it also allows us to easily create countless variations of the method, adaptable to the interest of the study.

Building subspaces in DW

The criteria in which subspaces W_i are created in the DW method is the following. First, two parameters are chosen, the algorithm precision ϵ, and the maximum number of levels M.

Then, the subspace V_1 that ϵ-spans V_0, will contain the largest eigenvalues of the graph, so that

$$\lambda_i \in E_{V_1} \quad \text{if} \quad \lambda_i \geq \epsilon$$

where E_{V_1} defines the set of eigenvalues corresponding to the subspace V_1.

These frequencies correspond to the eigenvalues of the normalized adjacency matrix $\tilde{\mathbf{A}}$, used as a diffusion operator. Therefore, its orthogonal complement W_1 will correspond to the rest of the eigenvalues of $\mathbf{A}$, that is,

$$\lambda_i \in E_{W_1} \quad \text{if} \quad 0 \leq \lambda_i \leq \epsilon$$

Knowing that $V_1 = V_2 \oplus W_2$, we can establish the same condition for W_2. This subspaces are created from the diffusion operator $\tilde{\mathbf{A}}^2$, so in this case V_2 will contain the frequencies of $\tilde{\mathbf{A}}^2$ that satisfy $\lambda_i^{(2)} \geq \epsilon$. It is straightforward that if λ is an eigenvalue of $\tilde{\mathbf{A}}$ with eigenvector $\mathbf{v}$, then λ^2 is an eigenvalue of $\tilde{\mathbf{A}}^2$ for the same eigenvector. Thus, since $\lambda_i^{(2)} = \lambda_i^2$ we can reformulate the condition as:

$$\lambda_i \in E_{V_2} \quad \text{if} \quad \lambda_i \geq \epsilon^{1/2} \quad \text{and} \quad \lambda_i \in E_{W_2} \quad \text{if} \quad \epsilon \leq \lambda_i \leq \epsilon^{1/2}$$

and in general, the expression for any subspace W_k, is:

$$\lambda_i \in E_{W_k} \quad \text{if} \quad \epsilon^{1/2^{k-2}} \leq \lambda_i \leq \epsilon^{1/2^{k-1}} \tag{4.4}$$

Note that in the DW formulation $V_0 = V_i \oplus W_i \oplus W_{i-1} \oplus \ldots \oplus W_1$. So for the highest value of k, V_k will be considered as one of the orthogonal subspaces needed to form a basis of $\mathbb{R}^N$, renaming it as $V_k = W_f$. Therefore, f orthogonal subspaces will be ultimately formed.

As can be seen, the presented condition forces a strict grouping of the frequencies, that leads to a fixed distribution without a clear purpose. It can also cause subspace dimensions to differ significantly from each other, and all this properties will depend exclusively on the chosen precision ϵ. For example, Fig. 4.3 shows how frequencies would be distributed when choosing a precision of 10^-3. Any graph, no matter its size, edge density or distribution, will show that exact grouping of eigenvalues.

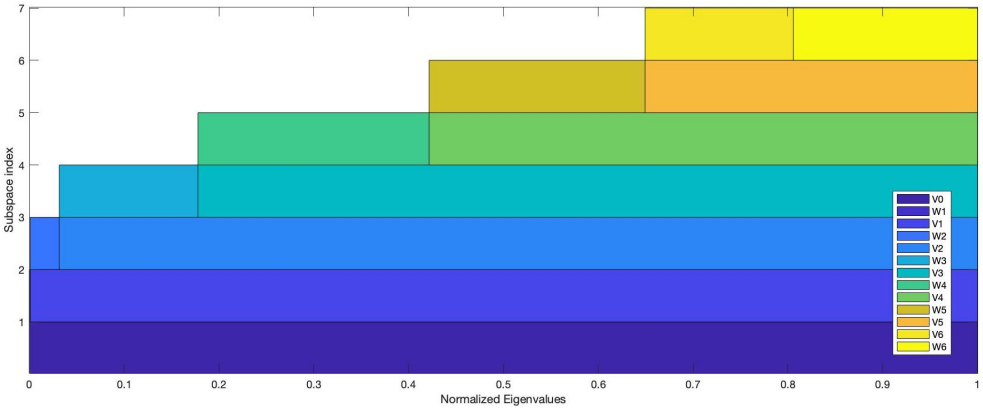


Figure 4.3: Diagram showing how the eigenvalues of any graph are grouped into subspaces in the DW method with a precison $\epsilon = 10^{-3}$. Note that the figure shows only the first 6 sublevels and W_1 can not be appreciated because of its small size (W_1 contains λ such that $0 < \lambda < 0.001$).

It is important to note that, when numerically computing the DW method, some problems may arise: First, if the $k-$th diffusion results on a subspace V_k of dimension 1, it is not possible to create more subspaces, regardless of the amount of levels initially chosen as the parameter M. This causes, in most cases, that the number of subspaces that are actually created stops long before it reaches M. This happens because there is a significant gap between λ_N, which is 1 since eigenvalues are normalized, and λ_{N-1}.

For example, we can see from Fig. 4.3 that with a precision $\epsilon = 10^{-3}$, for a graph with $\lambda_{N-1} = 0.5$ only 5 iterations can be performed, obtaining the subspaces $W_1, W_2, ..., W_5, W_6$. It is also probable that the studied graph adjacency matrix does not have any eigenvalue in the range corresponding to a certain subspace, a situation that will leave an empty basis in the numerical computation of the method. For example, a graph might not have any eigenvalue so that $10^{-3} \leq \lambda \leq 10^{-3/2}$, so computing its DW with $\epsilon = 10^{-3}$ would result in an empty space W_2.

Building subspaces with the GST

On the presented approach a different strategy is used to separate frequencies into subspaces. In this case, the only parameter to decide is the desired number of subspaces M. Then, the criteria to group eigenvalues into different subspaces will be to find the $M-1$ separation points where the distance between consecutive eigenvalues is greater.

This computation would correspond to the 4th step of Algorithm 2, and is described in more detail in the following algorithm:

Algorithm 3 Distribution of eigenvalues into subspaces from the GST

1: $\mathbf{e} \leftarrow$ Sort the set of eigenvalues of $\tilde{\mathbf{A}}$
2: $\mathbf{d} \leftarrow$ Calculate and store the distance between consecutive eigenvalues.
3: $\mathbf{b} \leftarrow$ Find the $M-1$ higher values of $\mathbf{d}$.
4: $\mathbf{k} \leftarrow$ Store the position of the $M-1$ higher values, to know the separation points between subspaces.

Example: Testing both methods for a random synthetic graph of 500 nodes setting the tuning parameters to $M = 30$ and $\epsilon = 10^{-5}$ for the DW algorithm, the distribution of eigenvalues into subspaces is shown in Fig. 4.4. It is clear from the obtained results that the GST shows a better spectral localization with a smaller variance in the magnitude of eigenvalues contained in each subspace, providing a more precise handling of the subspaces.

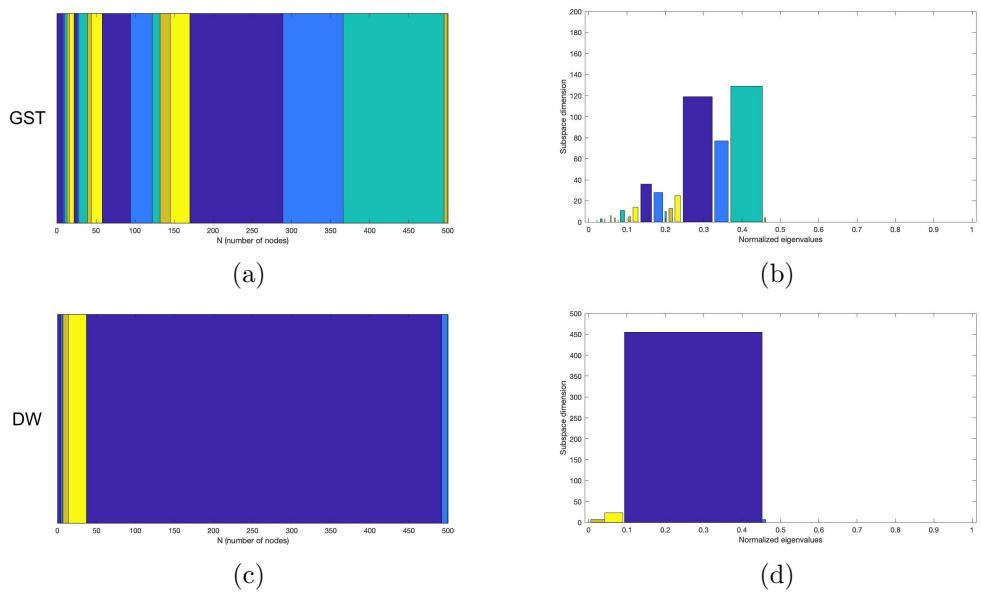


Figure 4.4: a) and c) show how the 500 eigenvalues of the graph are grouped into subspaces. b) and d) show the range filled by each group of normalized eigenvalues and the dimension of its subspace.

Comparing the variance in the dimension of subspaces

To better understand the performance of each method and give a numerical sense of why we consider the distribution on the GST construction is better localized, we compute the variance on the dimension of subspaces build in each case. In the case of the GST construction, that is:

$$S^2_{GST} = \frac{\sum_{k=1}^{M}(s_k - \mu)^2}{M-1}$$

where M corresponds to the total number of subspaces created, s_k defines the dimension of the subspace k and μ is the mean size for a subspace in the defined construction (that is, the average number of eigenvalues). Note that, for a given value of N and M, the mean size μ can be calculated as $\mu = N/M$. Thus, the variance in the GST construction can be expressed as:

$$S^2_{GST} = \frac{\sum_{k=1}^{M}(s_k - N/M)^2}{M-1}$$

For the basis built with the DW method, it is not clear that M subspaces will be constructed, so we define L as the number of levels that are actually created. Also, while for the GST method it could be expected to have an evenly spaced distribution so we could define the mean as $\mu = N/M$, in this case we already know that the size of the subspaces will increase gradually. Recalling that for each subspace W_k the corresponding normalized eigenvalues are found within the space $\epsilon^{1/2^{k-2}} \leq \lambda_i \leq \epsilon^{1/2^{k-1}}$, we define the measure μ_k for each subspace as:

$$\mu_k = N(\epsilon^{1/2^{k-1}} - \epsilon^{1/2^{k-2}}) \quad \text{for} \quad k \geq 2$$

For the specific case of $k = 1$ we have that $\mu_1 = N\epsilon$. Thus, to calculate the variance in the DW case we use:

$$S^2_{DW} = \frac{\sum_{k=1}^{L}(s_k - \mu_k)^2}{L - 1}$$

where, again, s_k defines the dimension of a subspace and μ_k is the amount of eigenvalues expected for each subspace in the DW method.

Calculating the variance for graphs of different sizes setting a different number of subspaces, the same result is obtained: For any graph size N the variance in the dimensions of the subspaces created with the GST method significantly decreases when the parameter M increases. It is important to recall that for DW the number of subspaces created does not necessarily correspond to the desired amount. This can lead to having a very small number of subspaces created in DW, while the GST creates exactly M subspaces. From this we can conclude that GST will always smooth the variation in the dimension of subspaces, giving a more regular distribution.

		Number of subspaces M		
		$N/25$	$N/10$	$N/5$
$N = 100$	DW	995.87	368.01	-
	GST	835.10	331.34	164.21
$N = 150$	DW	875.31	-	-
	GST	1419.36	493.67	235.97
$N = 200$	DW	1542.50	-	-
	GST	1864.43	676.21	318.93
$N = 250$	DW	2422.98	-	-
	GST	2283.09	813.52	395.74

Table 4.1: Variances S^2_{DW} and S^2_{GST} obtained by creating 100 random graphs for each case. Note that the mean number of subspaces built with the DW method for $M = N/10$ and $M = N/5$ was 6.5 so its variance in these cases cannot be compared with the result for GST. Note that the 3 graphics show a gap around $\lambda = 0.5$. This appears because when ordering the normalized eigenvalues increasingly, there exists a considerable difference between λ_{N-1} and $\lambda_N = 1$. Therefore, we see the distance between eigenvalues $\lambda 1 and \lambda_{N-1}$ at the left of the gap, and the distance between $\lambda_N = 1$ and the rest of eigenvalues at the right of the gap.

To sum up, the main advantages of the spectral localization provided by the GST are:

- Similar dimensions of subspaces.
- Better localized ranges, since more subspaces can be build.
- Ensures that the exact desired number of subspaces are build.
- The algorithm can be adjusted to different and more convenient criteria, depending on the purpose of the study.

4.2 The Graph Schur Transform on directed graphs

Let us recall that the ultimate goal of this research was to find a method to build a basis valid to use GSP tools on directed graphs, and specially on those with a defective adjacency matrix. As explained in Chapter 2 the reason why exceptions cannot happen on undirected graphs is because they are, by definition, always diagonalizable. That is, that a complete basis of eigenvectors can be build by diagonalizing any of its graph operators (in our case, the adjacency matrix $\mathbf{A}$). This basis forms a transformation matrix $\mathbf{U}$ such that

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$$

where $\mathbf{\Lambda}$ is a diagonal matrix formed by the eigenvalues of $\mathbf{A}$. The presented $\mathbf{U}$ matrix is needed to build the GFT of any graph signal, as introduced in chapter 1, where

$$\tilde{\mathbf{x}} = \mathbf{U}^{-1}\mathbf{x}$$

In the case of directed graphs, as we observed in Chapter 2, the adjacency matrix is not always diagonalizable and other graph operators such as the Laplacian cannot be computed because of the nature of the graph and the presence of sinks and sources. Our main purpose was to find a solution for this kind of graphs.

For a non-diagonalizable or defective graph a complete set of linearly independent eigenvectors does not exist, so a complete basis for $\mathbb{R}^N$ cannot be formed. This means that, even the $\mathbf{U}$ matrix can be computed, its rank will be $\text{ran}(\mathbf{U}) < N$ and this would make the matrix $\mathbf{U}$ not invertible (also called singular). Therefore the matrix $\mathbf{U}$ cannot be used to form the GFT of a signal in graphs with a defective $\mathbf{A}$ matrix, and our proposed matrix $\mathbf{U}_f$ can be used instead.

4.2.1 Numerical results on directed graphs

To experimentally test if the proposed method could be a solution for the presented problem, we need to see if the $\mathbf{U}_f$ matrix build with the GST method forms a complete basis of $\mathbb{R}^N$ for any directed graph.

We check that this property holds on random synthetic graphs, doing it for graphs of different size and also creating a different number of subspaces on each case to verify that it performs correctly in any type of graph. In particular, the performed verification consisted on the following tests:

- Graphs of 50, 100, 150, 200, 300 and 500 graphs were created.
- For each size, the GST was computed for different numbers of subspaces (M): $N/25$, $N/10$, $N/5$ and $N/2$.
- All graphs where created with a random edge probability $p \in [5/N, 1/N]$.
- For each selected N and M, 100 graphs were created to calculate the proportion of non-diagonalizable graphs.

While for a large amount of graphs the matrix $\mathbf{U}$ presented a rank lower than N (Fig. 4.5), meaning that many non-diagonalizable graphs had been created, the matrix $\mathbf{U}_f$ was fullrank for absolutely all the created graphs and therefore it could be properly used for GSP.

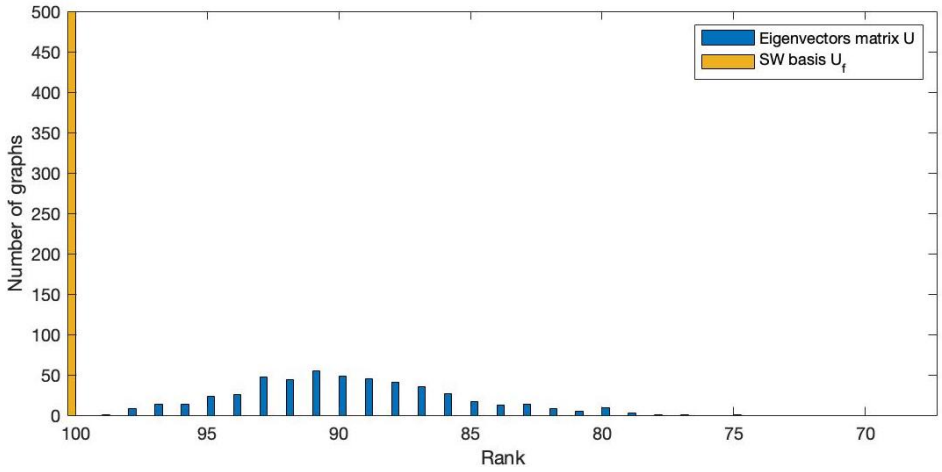


Figure 4.5: Histogram showing the rank of the Schur $\mathbf{U}_f$ matrix (yellow) and the matrix of eigenvectors $\mathbf{U}$ (blue) for 500 graphs with N=100. The graphic clearly shows that, while the adjacency matrix was defective for the 500 tested graphs, the GST always forms a complete basis for $\mathbb{R}^N$, with ran($\mathbf{U}_f$)= N.

This shows that for any directed graph, either strongly or weakly connected, our method builds a complete basis of $\mathbb{R}^N$ that can be used to study the graph in the frequency domain.

Note that in the case of DAGs (directed acyclic graphs, described in section 2.2.2) our method will not perform correctly for various reasons. First, the adjacency matrix $\mathbf{A}$ cannot be divided by the maximum eigenvalue since, by definition, all the eigenvalues are zero for a DAG. Second, remember that the adjacency matrix for a DAG is a nilpotent matrix, that is, an upper triangular matrix with zeros (the eigenvalues) in its diagonal. Therefore, we have that the Schur matrix $\mathbf{T}$ has the same form as the adjacency matrix $\mathbf{A}$. This means that in this case the transformation matrix $\mathbf{U}$ in the expresion $\mathbf{A} = \mathbf{U}\mathbf{T}\mathbf{U}^{\mathsf{H}}$ would be $\mathbf{U} = \mathbf{I}$. Which makes it not useful for our purpose.

Finally, the construction of our method groups eigenvalues into M groups, separating them when their magnitudes are different enough. This is not possible in the case of DAGs, where all the eigenvalues have magnitude zero.

Chapter 5

Conclusions

The main objective of this thesis has been to deepen in the study of defective directed graphs in order to develop a systematic tool to process any graph from this group in the spectral domain, to be able to implement Graph Signal Processing techniques (GSP).

We start in Chapter 1 by giving a brief introduction to fundamental concepts of the GSP field. We first introduced the main definitions regarding the graph structure and its elements and properties and its algebraic representation. Then we presented the most essential ideas regarding graph signals and its frequency representation, defining the concept of graph operators and polynomials in order to present the most important element in GSP: The Graph Fourier Transform (GFT).

In Chapter 2 we introduce the difficulty presented by certain graphs for which the conventional GFT can not be computed. The problem they face is that when the algebraic representation of the studied graph operator is not diagonalizable (defective), a complete basis of eigenvectors can not be used as the GFT of the graph. This issue has led many authors to study different approaches to find an alternative. The most popular solution proposed has been to compute the Jordan form of the adjacency matrix of the graph, but the numerical unstability of this method is well-known in the GSP field.

Before presenting our own contribution, we study in depth the behaviour of different directed graphs, analyzing the diagonalizability of both weakly and strongly connected graphs. The results show that in both groups a high percentage of defective graphs can be found, meaning that in order to offer a systematic solution all graph topologies must be considered. However, we consider important to mention some specific properties for which we can ensure either diagonalizability or defectiveness for any graphs that satisfies them.

The method developed in this research, called Graph Schur Transform (GST), is presented in Chapter 3 after first introducing the two concepts in which it is based: The Schur decomposition and the Diffusion Wavelets configuration (presented in). The construction of the GST is based on the creation of subspaces of $\mathbb{R}^N$ formed by a subset of vectors obtained by recursively computing the Schur decomposition of the adjacency matrix. By deciding the exact number of subspaces (M) that we aim to obtain, the algorithm computes an orthogonal basis for each of them, and a final matrix $\mathbf{U}$, which represents an alternative to the conventional GFT.

We divide Chapter 4 in two parts: The comparison of the main properties of our method with those of the Diffusion Wavelets configuration, and the study of the performance of the GST on random directed graphs.

First, we analyze the most relevant properties of the GST method and compare them with what we observe in the Diffusion Wavelets configuration:

- *Subspace invariance.* We start by studying the invariance of the subspaces created. While subspaces built through the GST method are exactly invariant, we see that the DW approach presents approximate invariance of the subspaces.

- *Orthogonality.* By construction, in the GST method the subspaces are formed by orthogonal vectors. However, while the DW design presents orthogonality between different subspaces, in our case vectors from different subspaces are approximately orthogonal. We observe a higher orthogonality when the number of subspaces created is relatively small, and we also see that for a higher distance between eigenvalues, the inner product between the corresponding vectors gets smaller, meaning that they are closer to be orthogonal.

- *Spectral localization.* We see the highest difference between both methods by analyzing their distribution of eigenvalues into subspaces. In the case of the DW design, the grouping is fixed, and depends only of the precision ϵ, which will define the set of eigenvalues corresponding to each subspace. Also, even the desired maximum number of subspaces M can be defined, this method does not always return M subspaces, since by construction it usually stops at a smaller number. But in the case of the GST, exactly M subspaces are created and the variance in the size of the subspaces is smaller, meaning that their dimension is, in most cases, more regular.

It is also worth mentioning the high flexibility of the developed method, since many different slight changes could be applied to obtain a version of the GST with different properties, that could be more appropriate depending on the purpose of the processing. The development and analysis of other possibilities could be studied in future investigations.

In the last section we run a numerical test to study the performance of the proposed method on a wide set of directed graphs. We observe that while almost all the created graphs are defective (with $\text{rank}(\mathbf{U}) < N$), absolutely all the directed graphs created in the experiment present a GST matrix $\mathbf{U}_f$ with $\text{rank}(\mathbf{U}_f)$=N, meaning that the GST offers a valid alternative to the GFT for cases where the conventional version of the GFT can not be computed because of the defectiveness of the graph operator. We finally remark a limitation of the proposed method, which consists on the incapacity of our method to find a GST for the set of graphs which spectrum is formed by N zeros, called directed acyclic graphs. It would be the object of future work to find an alternative to the conventional GFT valid for this specific group of graphs.

Bibliography

[1] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges and applications," 2017.

[2] Matthew O. Jackson, *Social and Economic Networks*. Princeton University Press, 2008.

[3] A. Sandryhaila and José M. F. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, p. 1644–1656, Apr 2013.

[4] 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 Processing Magazine*, vol. 30, no. 3, p. 83–98, May 2013.

[5] A. Ortega, *An Introduction to Graph Signal Processing*. Cambridge University Press, 2020.

[6] M. Jordan, E. Sudderth, M. Wainwright, and A. Willsky, "Major advances and emerging developments of graphical models [from the guest editors]," *Signal Processing Magazine, IEEE*, vol. 27, pp. 17–138, Nov 2010.

[7] J. Tenenbaum, V. Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science (New York, N.Y.)*, vol. 290, pp. 2319–23, Jan 2001.

[8] M. Newman, *Networks: An introduction*. OUP Oxford, 2010.

[9] M. Wainwright and M. Jordan, *Graphical Models, Exponential Families, and Variational Inference*, Jan 2007.

[10] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, *Distributed regression: An efficient framework for modeling sensor network data*, May 2004.

[11] R. Wagner, Hyeokho Choi, R. Baraniuk, and V. Delouille, "Distributed wavelet transform for irregular sensor network grids," in *IEEE/SP 13th Workshop on Statistical Signal Processing*, 2005, pp. 1196–1201.

[12] D. Hammond, P. Vandergheynst, and R. Gribonval, *The Spectral Graph Wavelet Transform: Fundamental Theory and Fast Computation*, Jan 2019, pp. 141–175.

[13] R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 53–94, 2006.

[14] S. K. Narang and A. Ortega, "Local two-channel critically sampled filter-banks on graphs," Sep 2010, pp. 333–336.

[15] ——, "Perfect reconstruction two-channel wavelet filter banks for graph structured data," vol. 60, Jun 2012, pp. 2786–2799.

[16] A. Agaskar and Y. Lu, "A spectral graph uncertainty principle," *Information Theory, IEEE Transactions on*, vol. 59, pp. 4338–4356, Jul 2013.

[17] R. Bronson, "Chapter 8 - jordan canonical forms," in *Matrix Methods*. Academic Press, 1969, pp. 176 – 220.

[18] J. A. Deri, "Graph signal processing: Structure and scalability to massive data sets," 2016.

[19] J. Deri and J. Moura, "Spectral projector-based graph fourier transforms," *IEEE Journal of Selected Topics in Signal Processing*, vol. PP, Jan 2017.

[20] P. Misiakos, C. Wendler, and M. Püschel, "Diagonalizable shift and filters for directed graphs based on the jordan-chevalley decomposition," Jan 2020.

[21] S. Sardellitti and S. Barbarossa, "On the graph fourier transform for directed graphs," *IEEE Journal of Selected Topics in Signal Processing*, vol. PP, Jan 2016.

[22] R. Shafipour, A. Khodabakhsh, G. Mateos, and E. Nikolova, "A directed graph fourier transform with spread frequency components," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, p. 946–960, Feb 2019.

[23] B. Girault, A. Ortega, and S. S. Narayanan, "Irregularity-aware graph fourier transforms," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, p. 5746–5761, Nov 2018.

[24] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," in *Stanford InfoLab*, 1999.

[25] R. A. Rossi and D. F. Gleich, "Dynamic pagerank using evolving teleportation," in *Algorithms and Models for the Web Graph*, A. Bonato and J. Janssen, Eds. Springer Berlin Heidelberg, 2012, pp. 126–137.

[26] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[27] D. I. Shuman, C. Wiesmeyr, N. Holighaus, and P. Vandergheynst, "Spectrum-adapted tight graph wavelet and vertex-frequency frames," *IEEE Transactions on Signal Processing*, vol. 63, no. 16, pp. 4223–4235, 2015.

[28] S. K. Narang and A. Ortega, "Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 19, pp. 4673–4685, 2013.

[29] J. C. Bremer, R. R. Coifman, M. Maggioni, and A. D. Szlam, "Diffusion wavelet packets," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 95 – 112, 2006.

[30] "The expected eigenvalue distribution of a large regular graph," *Linear Algebra and its Applications*, vol. 40, pp. 203 – 216, 1981.

[31] S. Gutekunst, "Lecture notes in spectral graph theory," Sep 2016.

[32] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[33] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, p. 3775–3789, Jul 2016.

[34] A. A. Mailybaev, "Computation of multiple eigenvalues and generalized eigenvectors for matrices dependent on parameters," *Numerical Linear Algebra with Applications*, vol. 13, no. 5, pp. 419–436, 2006.

[35] J. Gallier, *Spectral Theorems in Euclidean and Hermitian Spaces*. Springer New York, 2011, pp. 343–365.

[36] L. Erdős, A. Knowles, H.-T. Yau, and J. Yin, "Spectral statistics of Erdős-Rényi graphs ii: Eigenvalue spacing and the extreme eigenvalues," *Communications in Mathematical Physics*, vol. 314, no. 3, p. 587–640, Aug 2012.

[37] B. G. Zaslavsky and B.-S. Tam, "On the Jordan form of an irreducible matrix with eventually nonnegative powers," *Linear Algebra and its Applications*, vol. 302-303, pp. 303 – 330, 1999.

[38] C. D. Godsil, S. A. Hobart, and W. J. Martin, "Representations of directed strongly regular graphs," *European Journal of Combinatorics*, vol. 28, no. 7, pp. 1980 – 1993, 2007.

[39] J. Bang-Jensen and G. Gutin, "Digraphs - theory, algorithms and applications," 2002.

[40] Y. Tan, "On nilpotent matrices over antirings," *Linear Algebra and its Applications*, vol. 429, no. 5, pp. 1243 – 1253, 2008.

[41] R. Beezer, *A First Course in Linear Algebra*, ser. Open Textbook Library. Congruent Press, 2012.

[42] C. Paige and C. V. Loan], "A schur decomposition for hamiltonian matrices," *Linear Algebra and its Applications*, vol. 41, pp. 11 – 32, 1981.