

**Federal University of Pernambuco**  
Technology and Geoscience Center  
Electrical Engineering Graduate Program

**Foundations of Quaternion Graph Signal  
Processing and related contributions to  
fractional-order operators**

Doctoral thesis

**Guilherme Boaviagem Ribeiro**  
MSc in Electrical Engineering  
by the Federal University of Pernambuco

Supervisor: Prof. Dr. Juliano Bandeira Lima

Recife  
2022



**Guilherme Boaviagem Ribeiro**

## **Foundations of Quaternion Graph Signal Processing and related contributions to fractional-order operators**

This work was presented to the Electrical Engineering Graduate Program, at the Technology and Geoscience Center of the Federal University of Pernambuco, as a partial requisite to obtain a doctorate degree on Electrical Engineering, in the concentration field of Communications.

Supervisor: Prof. Dr. Juliano Bandeira Lima

Recife  
2022



*À minha esposa, Bárbara, e aos meus filhos, Rafael e Gianna.*



# Agradecimentos

*Insisto: na simplicidade do teu trabalho habitual, nos detalhes monótonos de cada dia, tens que descobrir o segredo – para tantos escondido – da grandeza e da novidade: o Amor.*

— São Josemaria Escrivá, Sulco, n. 489.

Ao toque do Amor, uma Pessoa divina que lhe chama e move o coração em meio às atividades corriqueiras, é inevitável o sentimento de gratidão. Especialmente, aqui, agradeço pelas graças concedidas para realizar este trabalho e frutificar parte do tempo que Ele me concedeu. Sua benignidade não tem fim.

Fujo da língua inglesa, por um momento, para dirigir também meus profundos e sinceros agradecimentos a algumas pessoas em nossa língua materna; amigos e familiares que foram fundamentais para tornar possível o desenvolvimento e conclusão deste trabalho.

Agradeço a Bárbara, minha esposa, por ser a cada dia ocasião de encontrar o nosso Amor mais profundo. Além de dividir comigo a vida e o futuro, sua dedicação nas pequenas coisas tornaram possível que eu concluisse esta jornada.

Agradeço a meus pais e meu irmão, pela educação e exemplo com que me nutriram ao longo da vida. Deram-me a possibilidade de ser quem sou, e por isso serei eternamente grato. Também a meus sogros, cunhadas e compadres, cujo carinho forte tal qual laços de sangue se manifesta no suporte tão grande que dão a mim e minha família.

Agradeço ao prof. Juliano Bandeira, meu orientador, que com incrível paciência e lucidez me ajudou a reencontrar o caminho da pesquisa, quando este se mostrava tão esquecido e desafiador. Sua orientação me ajudou a enxergar o caminho e reunir forças para as últimas milhas dessa corrida de longa distância.

Agradeço também a Petrônio Braga e meus colegas da Neurotech, pelas conversas dentro e fora do trabalho, que de uma forma ou de outra me ajudaram a amadurecer meu olhar e não deixar para trás minha veia acadêmica.

Agradeço, por fim, à CAPES e à Pró-reitoria de Pesquisa e Pós-graduação pelo auxílio financeiro durante parte dessa pesquisa (sob o projeto n. 88882.380374/2019-01), e ao Programa de Pós-graduação em Engenharia Elétrica da UFPE pela oportunidade concedida.

*“Algebra is generous, she often gives more than is asked of her”.*

(Jean le Rond d'Alembert, Bulletin American Mathematical Society, Vol. 2 (1905), p.  
285)



# Abstract

The field signal processing, at its core, is concerned with exploring how different representations of a signal may provide useful ways of manipulating it. These representations may arise from a change in the subjacent algebra over which the signal samples are defined, for example when embedding three- and four-dimensional signals into the quaternion space; or maybe from a different model of the signal domain, as it happened with the development of graph signal processing to deal with network-like data; or even by exploring new linear transforms that map the signal onto a domain in which tasks such as compression, filtering or feature extraction are easier or more efficient.

This thesis traverses exactly through these paths, aiming at answering the question of how to extend graph signal processing to the case in which signals and edge weights are quaternion-valued. It proposes a new set of tools which are a basis for what may be called Quaternion Graph Signal Processing (QGSP) and, as byproducts of the research journey, it contributes to the field of fractional transforms in two fronts: by proposing a new approach to the fractionalization of the quaternion discrete Fourier transform (QDFT), alongside the proposition of its multiparametric version, and by proposing a new fractional graph shift operator (GSO).

As highlights of the main results, we can mention: (1) the polynomial representation of the fractional GSO for arbitrary graphs was obtained, and it was shown that its use in filter design of FIR LSI filters improve the overall filter quality for a given filter length; (2) the new multiparametric fractional QDFT was used to create a holistic encryption scheme for color images with opacity layer, which was shown to provide satisfactorily large key space and key sensitivity; and (3) the main aspects of spectral analysis, filtering and compression in the context of QGSP were formulated, along extensive practical examples on real-world data computed through a custom-made and open-source Python package.

**Keywords:** quaternions, fractional quaternion discrete Fourier transform, fractional graph shift operator, quaternion graph signal processing.



# List of Figures

|     |  |    |
|-----|--|----|
| 2.1 | Illustration of the multiplication rule between the imaginary units $i$ , $j$ and $k$ . . . . .  | 9  |
| 2.2 | Representation of $\lambda = 3i + k$ in the $ik$ plane. . . . .  | 14 |
| 2.3 | <i>Standard eigenvalues</i> of a quaternion matrix $\mathbf{A}$ , indicated as a subset of the eigenvalues of the complex adjoint matrix. . . . .  | 18 |
| 3.1 | Illustration of the FrFT property of rotation in the time-frequency plane. Adapted from (Almeida 1994). . . . .  | 25 |
| 3.2 | Example of signal defined over a graph. The height of the vertical bars indicate the value of the signal samples, which are indexed by the graph vertices. The graph edges capture similarity relations between samples. How could one define spectral analysis and processing techniques in such a signal domain? . . . . .   | 27 |
| 3.3 | Examples of (a) directed and (b) undirected graphs, defined over the same vertex set. . . . .  | 29 |
| 3.4 | (a) A graph and (b) the set of vertices $\mathcal{N}(i, 2)$ shown in red, with $v_i$ being depicted in white. The edges linking $v_i$ to the elements in $\mathcal{N}(i, 2)$ are also highlighted in red. . . . .  | 30 |
| 3.5 | Examples of depictions of graph signals over (a) a directed ring graph, (b) an undirected regular grid graph and (c) a graph of cities from the Brazilian Northeastern region, over which was defined a signal of temperature measurements from February 1 <sup>st</sup> of 2012, retrieved from the <i>Banco de Dados Meteorológicos para Ensino e Pesquisa</i> (BD-MEP, freely translated as Meteorological Database for Teaching and Research), available at: <a href="http://www.inmet.gov.br/portal/index.php?r=bdmep/bdmep">http://www.inmet.gov.br/portal/index.php?r=bdmep/bdmep</a> . . . . . | 31 |
| 3.6 | The same signal defined over two similar graphs, one of them being (a) the undirected ring graph. In (b) and (d) are depicted the Fourier spectra of the signals in (a) and (c), respectively. . . . .   | 32 |
| 3.7 | (a) Signal defined over an undirected ring graph and (b) its spectrum. The frequencies are indicated by the eigenvalues of the adjacency matrix.   | 37 |

|      |   |    |
|------|---|----|
| 3.8  | Frequency ordering of graph signals, from low to high frequencies, in the complex plane (Sandryhaila and Moura 2014b). . . . .  | 39 |
| 3.9  | (a) Directed sensor graph, with $N = 100$ vertices and no loops or multiple edges. (b) Number of zero crossings and (b) total variation of the eigenvectors $(\mathbf{v}_i)_{i=0,\dots,N-1}$ of the adjacency matrix $\mathbf{A}$ of the graph in (a), ordered so that the respective eigenvalues appear from the closest to the farthest from the real point $ \lambda_{max} $ in the complex plane. That is, according to (3.35) and Fig. 3.8, the eigenvectors are disposed in ascending order of frequency. . . . . | 40 |
| 4.1  | Fractional shift by $a = 0.3$ of a sample of a signal on a directed ring graph with unit weights. (a) Original signal on a directed ring graph. (b) Graph in which the 5 <sup>th</sup> sample delayed by $a = 0.3$ appears as an interpolated sample between the 4 <sup>th</sup> and the 5 <sup>th</sup> samples of the original signal. (c) Original discrete signal and the delayed sample. . . . .   | 57 |
| 4.2  | Fractional shift of a signal, (originally) with 10 non-zero samples, defined on a graph formed by 80 cities of Pernambuco state, Brazil. Note that the shifted signal is similar to the original signal, if $a$ is close to 0, and similar to the unit-shifted signal, if $a$ is close to 1. . . . .  | 57 |
| 4.3  | Fractional shift for a signal (a) without and (b) with abrupt variations (descontinuities). . . . .   | 57 |
| 4.4  | Percent error (normalized by the energy of $\mathbf{h}_{LPF}$ ) of $\mathbf{h}_a$ related to $\mathbf{h}_{LPF}$ , for different (odd) values of $N$ and the fractional shift parameter $a$ . . . . .  | 58 |
| 4.5  | Relative mean error between $\mathcal{R}e\{\mathbf{h}_a\}$ and $\mathbf{h}_{LPF}$ for $N$ even, in terms of the fractional shift parameter $a$ . . . . .  | 58 |
| 4.6  | Modulus of the relative mean error between $\mathbf{h}_a$ and $\mathbf{h}_{LPF}$ for $N$ even, in terms of the fractional shift parameter $a$ . . . . .   | 58 |
| 4.7  | Directed graph used to illustrate how the corresponding fractional shift operator can be computed and represented in polynomial form. . . . .   | 59 |
| 4.8  | Graph of a network formed by 230 weather stations measuring the temperature across the United States. The snapshot of all measurements taken on February 1 <sup>st</sup> , 2003 is the corresponding graph signal. . . . .  | 59 |
| 4.9  | Magnitude of the graph Fourier transform of the signal in Fig 4.8. The graph frequencies correspond to the eigenvalues of $\mathbf{A}$ ; low (resp. high) graph frequencies are associated with higher (resp. lower) eigenvalues. . . . .   | 59 |
| 4.10 | Fractional parameters providing the minimum approximation errors, for different values of $L$ , between the ideal LPF and the filter designed by using the fractional graph shift operator $\mathbf{A}^a$ . . . . .   | 60 |

|      |  |    |
|------|--|----|
| 4.11 | Minimum approximation (mean squared) errors, for different values of $L$ , between the ideal LPF and the filters designed by using the fractional graph shift operator $\mathbf{A}^a$ and the non-fractional operator $\mathbf{A}$ . . . . .   | 60 |
| 4.12 | Ideal filter response superimposed on the responses obtained when $\mathbf{A}$ and $\mathbf{A}^a$ , $a = 0.855$ , are used to design a filter with $L + 1 = 10$ coefficients. . . . .  | 60 |
| 4.13 | Reconstruction (mean squared) errors after a noise removal procedure is performed by using graph filters with $L + 1 = 10$ coefficients and designed from $\mathbf{A}$ and $\mathbf{A}^a$ , $a = 0.855$ . . . . .  | 61 |
| 5.1  | (a) 2D-QDFT of the PNG image in Fig. 5.2a, according to (5.13), with axis $\mu = \frac{1}{\sqrt{3}}(i+j+k)$ . (b) 2D-QDFT of the same image, same transform axis, following the approach in (5.14). . . . .  | 69 |
| 5.2  | (a) Test PNG image. Visualization of each layer in the (b) test image and (c) in its FrQDFT spectrum, computed with transform axis $\mu = \frac{1}{\sqrt{3}}(i + j + k)$ and $a = 0.3$ . . . . .   | 70 |
| 5.3  | Proposed encryption scheme, exploring the 2D-MFrQDFT. . . . .  | 73 |
| 5.4  | (a) Encrypted image. (b) Image decrypted with the correct key $s_0$ . (c) Image decrypted with the wrong key $\tilde{s}_0 = s_0 + \epsilon$ , with $\epsilon = -1,6 \cdot 10^{-80}$ . . . . .  | 73 |
| 5.5  | Mean squared error, in dB, between the original and the decrypted images as function of the key error $\epsilon$ . . . . .   | 76 |
| 5.6  | (a) Test image in PNG format, with $256 \times 256$ pixels. (b) Color and opacity layers of the <i>decrypted</i> image (which coincides with the original in (a)). (c) Histograms of each layer of the <i>decrypted</i> image. (d) Encrypted image, in PNG format. (e) Color and opacity layers of the <i>encrypted</i> image. (f) Histograms of each layer of the <i>encrypted</i> image. . . . . | 78 |
| 5.8  | Set of 13 PNG test images, $256 \times 256$ pixels, alongside the histograms of each layer (colors and opacity) of their encrypted version. Images downloaded from the online free database <a href="https://purepng.com/">https://purepng.com/</a> , under CC0 license. . . . .   | 80 |
| 6.1  | Illustration of two connected vertices in a graph with Hermitian adjacency matrix (left-hand side). In the center and on the right, respectively, are displayed the edges created by the simplex and perplex parts of the adjacency matrix. . . . .  | 90 |
| 6.2  | Graph total variation (pseudocolor scale) of the normalized eigenvectors, as a function of their respective eigenvalues. . . . .   | 95 |
| 6.3  | Graph used in the numerical example. The edges are represented by double-headed arrows to indicate the adjacency matrix is not symmetrical. . . . .  | 96 |
| 6.4  | Graph created using the 145 towns in England and Wales. . . . .  | 98 |

|      |  |     |
|------|--|-----|
| 6.5  | Quaternion graph signal, visualized in the vertex domain. The panels (a) to (d) depict, respectively, the real, $i$ , $j$ and $k$ components of each quaternion-valued sample. . . . . | 100 |
| 6.6  | Spectrum of the quaternion graph signal, defined over the vertices of the graph with Hermitian adjacency matrix. . . . .   | 101 |
| 6.7  | Spectrum of the noisy signal. . . . .  | 101 |
| 6.8  | Total variation of the eigenvectors of the Hermitian graph adjacency matrix, for each respective eigenvalue. . . . .   | 102 |
| 6.9  | QLMS cost function for each iteration and for each step size. . . . .  | 102 |
| 6.10 | Frequency response of the FIR low-pass filter after running the QLMS. .  | 103 |
| 6.11 | Spectrum of the original signal in the symmetric graph. . . . .  | 104 |
| 6.12 | Total variation of eigenvectors of the symmetric graph adjacency matrix.   | 105 |
| 6.13 | Frequency response of the FIR low-pass filter after running the QLMS in the scenario with symmetric graph adjacency matrix. . . . .  | 106 |
| 6.14 | QLMS cost function for each iteration and for each step size, in the scenario with symmetric graph adjacency matrix. . . . .   | 107 |
| 6.15 | Nearest-neighbors graph of a few of the United States counties. . . . .  | 107 |
| 6.16 | Sociodemographic quaternion-valued graph signal, split into its components: (a) real part, (b) $i$ , (c) $j$ and (d) $k$ . . . . .   | 108 |
| 6.17 | Total variation for each eigenvector of the (Hermitian) adjacency matrix of the US graph. . . . .  | 109 |
| 6.18 | US graph signal spectrum. . . . .  | 110 |
| 6.19 | Normalized mean squared error as a function of the compression rate. .   | 110 |

\*

# List of Tables

|     |  |    |
|-----|--|----|
| 5.1 | DFT and QDFT eigenvectors. . . . .                               | 68 |
| 6.1 | Sample of the dataset containing UK towns' weather data. . . . . | 99 |



# List of abbreviations

|                |  |
|----------------|--|
| <b>DSP</b>     | Digital Signal Processing.   |
| <b>LTI</b>     | Linear and Time-invariant.   |
| <b>FIR</b>     | Finite Impulse Response.   |
| <b>DFT</b>     | Discrete Fourier Transform.  |
| <b>QFT</b>     | Quaternion Fourier Transform.                                      |
| <b>FrFT</b>    | Fractional Fourier Transform.                                      |
| <b>FrDFT</b>   | Fractional Discrete Fourier Transform.                             |
| <b>FrQDFT</b>  | Fractional Quaternion Discrete Fourier Transform.                  |
| <b>MFrQDFT</b> | Multi-parametric Fractional Quaternion Discrete Fourier Transform. |
| <b>GSP</b>     | Graph Signal Processing.   |
| <b>GSO</b>     | Graph Shift Operator.  |
| <b>QGSP</b>    | Quaternion Graph Signal Processing.                                |
| <b>LSI</b>     | Linear and Shift-invariant.  |
| <b>NMSE</b>    | Normalized Mean Squared Error.                                     |



# List of symbols

|   |   |
|---|---|
| <b>A</b>                                  | Graph weighted adjacency matrix.  |
| <b>L</b>                                  | Graph Laplacian matrix.   |
| <b>Λ, J</b>                               | Respectively the eigenvalue matrix (if it exists) and the Jordan matrix of the graph adjacency matrix.  |
| <b>V</b>                                  | The (possibly generalized) eigenvector matrix of the graph adjacency matrix.  |
| $\mathcal{R}e\{x\}$                       | Real part of the complex number $x$ .   |
| $\mathcal{I}m\{x\}$                       | Imaginary part of the complex number $x$ .  |
| $\bar{x}$                                 | Conjugate of the complex (or quaternion) $x$ . If vectors or matrices are used instead of $x$ , the conjugation is realized in each of its entries. |
| $\mathbf{M}^T$                            | Transpose of the matrix $\mathbf{M}$ .  |
| $\mathbf{M}^H$                            | Conjugate transpose of the matrix $\mathbf{M}$ .  |
| $\mathbb{R}, \mathbb{C}$ and $\mathbb{H}$ | Respectively the set of real, complex and (Hamiltonian) quaternion numbers. By association, we may also refer to the respective (skew-)field.       |
| $\mathbb{F}^n$                            | Set of the vectors of length $n$ with entries in the (possibly skew) field $\mathbb{F}$ .   |
| $ \cdot $                                 | Modulus of a real, complex or quaternion number.  |
| $\chi_A$                                  | Complex adjoint of the quaternion matrix $\mathbf{A}$ .   |
| $S(q)$                                    | Scalar part of the quaternion $q$ .   |
| $\mathbf{V}(q)$                           | Vector part of the quaternion $q$ .   |
| $\mathbf{V}(\mathbb{H})$                  | Set of all pure quaternions (i.e. with null scalar part).   |
| $i, j$ and $k$                            | Unit pure quaternions which constitute the canonical basis for the set of pure quaternions.   |

$\mathbb{C}_\mu$  Set of numbers  $\{a + b\mu \mid a, b \in \mathbb{R}\}$  with  $\mu \in V(\mathbb{H})$  and  $|\mu| = 1$ , isomorphic to the complex numbers.

$\|\mathbf{x}\|_p$   $\ell_p$ -norm of the vector  $\mathbf{x} \in \mathbb{C}^n$ , defined as  $\left(\sum_{k=0}^{n-1} |x_k|^p\right)^{1/p}$ .

# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>   | <b>1</b>  |
| 1.1      | Organization . . . . .  | 3         |
| 1.2      | Contributions . . . . .   | 3         |
| 1.3      | Publications . . . . .  | 4         |
| 1.4      | Notation . . . . .  | 5         |
| <b>2</b> | <b>A review on quaternion algebra and its Fourier transform</b> | <b>7</b>  |
| 2.1      | Introduction to the quaternion algebra . . . . .                | 8         |
| 2.1.1    | Quaternion similarity and rotation . . . . .                    | 13        |
| 2.2      | On the theory of quaternion matrices . . . . .                  | 15        |
| 2.2.1    | Eigenvalues . . . . .   | 16        |
| 2.2.2    | Diagonalizability . . . . .                                     | 17        |
| 2.3      | The quaternion Fourier transform . . . . .                      | 19        |
| 2.4      | Summary . . . . .   | 22        |
| <b>3</b> | <b>A review on fractional and graph Fourier transforms</b>      | <b>23</b> |
| 3.1      | The fractional Fourier transform . . . . .                      | 23        |
| 3.2      | Fundamentals of graph signal processing . . . . .               | 26        |
| 3.2.1    | The challenge of graph-like domains . . . . .                   | 28        |
| 3.3      | Principles and definitions . . . . .                            | 28        |
| 3.3.1    | Graph theory: a brief terminology . . . . .                     | 28        |
| 3.3.2    | Defining a graph signal . . . . .                               | 30        |
| 3.3.3    | Graph inference . . . . .                                       | 32        |
| 3.4      | Formulating GSP based on the graph adjacency matrix . . . . .   | 33        |
| 3.4.1    | Graph filters . . . . .   | 34        |
| 3.4.2    | Graph Fourier transform . . . . .                               | 35        |
| 3.4.3    | The frequency domain . . . . .                                  | 37        |
| 3.5      | Summary . . . . .   | 41        |
| <b>4</b> | <b>The fractional graph shift operator and its applications</b> | <b>43</b> |
| 4.1      | Fractional Shift on Graphs . . . . .                            | 45        |
| 4.1.1    | Computation of $\mathbf{A}^a$ . . . . .                         | 46        |

|                     |   |            |
|---------------------|---|------------|
| 4.1.2               | Interpreting the graph fractional shift . . . . .   | 47         |
| 4.1.3               | Consistency with classical approach: the ideal fractional delay filter . . . . .                    | 49         |
| 4.1.4               | Polynomial representation . . . . .   | 50         |
| 4.2                 | Numerical Results . . . . .   | 51         |
| 4.2.1               | Example: Polynomial Representation of $\mathbf{A}^a$ . . . . .                                      | 51         |
| 4.2.2               | Least-square approximation of LSI filters . . . . .   | 52         |
| 4.2.3               | Example: LS Approximation using $\mathbf{A}^a$ . . . . .  | 54         |
| 4.2.4               | Example: noise removal via LSI low-pass filtering . . . . .   | 55         |
| 4.3                 | Concluding remarks . . . . .  | 56         |
| <b>5</b>            | <b>The fractional quaternion discrete Fourier transform and its applications</b>                    | <b>63</b>  |
| 5.1                 | Eigenstructure of the QDFT . . . . .  | 65         |
| 5.1.1               | 2D-QDFT and color images with opacity layer . . . . .   | 68         |
| 5.2                 | The fractional quaternion discrete Fourier transform . . . . .                                      | 69         |
| 5.3                 | The multiparametric FrQDFT with application to color image encryption                               | 72         |
| 5.3.1               | Encryption scheme using MFrQDFT . . . . .   | 74         |
| 5.4                 | Concluding remarks . . . . .  | 77         |
| <b>6</b>            | <b>Quaternion graph signal processing</b>   | <b>83</b>  |
| 6.1                 | Laying the foundations for QGSP . . . . .   | 84         |
| 6.1.1               | Eigendecomposing the shift operator . . . . .   | 85         |
| 6.1.2               | Inversion of the eigenvector matrix and frequency ordering . . . . .                                | 86         |
| 6.1.3               | On the existence of a class of graphs with diagonalizable adjacency matrix . . . . .                | 89         |
| 6.1.4               | Does a Hermitian QGSO have unitary eigenvector matrix? . . . . .                                    | 91         |
| 6.1.5               | Numerical example: computing the QGFT matrix . . . . .  | 92         |
| 6.2                 | Filter design via QLMS optimization . . . . .   | 95         |
| 6.3                 | Examples . . . . .  | 97         |
| 6.3.1               | Denoising a quaternion graph signal via QLMS low-pass filtering                                     | 98         |
| 6.3.2               | Spectral analysis and compression of quaternion graph signal  | 105        |
| 6.4                 | Summary . . . . .   | 108        |
| <b>7</b>            | <b>Conclusion</b>   | <b>111</b> |
| 7.1                 | Challenges and future works . . . . .   | 112        |
| <b>A</b>            | <b>Mapping a system of linear equations from <math>\mathbb{H}</math> to <math>\mathbb{C}</math></b> | <b>115</b> |
| <b>BIBLIOGRAPHY</b> |   | <b>117</b> |

# Introduction

Signal processing is a field which often interweaves pure mathematics and engineering. One of its concerns is the *representation* of signals (functions) and how different representations may be explored to better manipulate such signals. The spectral analysis via Fourier and similar transforms, for instance, aims to project the signal onto a domain in which the energy support is more compact (compression), or in which some frequencies are easier to be removed (filtering), or yet in which some relevant features may be created (feature engineering for machine learning problems), among others (Oppenheim 1999; Rabiner and Schafer 2010; Graf et al. 2015; Vergin et al. 1999).

One way to explore different signal representations is to alter the algebra over which its sample values are defined. That is the case with number-theoretic transforms (Lima and Campello de Souza 2011; Blahut 2010; Pedrouzo-Ulloa et al. 2017; Chandra and Svalbe 2014), which deal only with finite algebraic structures instead of the usual complex (e.g. Fourier transform) or real fields (e.g. cosine transform). As the underlying algebra is extended, e.g. going from  $\mathbb{R}$  to  $\mathbb{C}$ , it is possible to process signals with more information per sample. Such was the motivation behind Sangwine's (Sangwine 1996) discrete version of a family of bidimensional transforms over the *quaternions*, previously created by Ell (Ell 1993): to exploit this class of hypercomplex numbers with four real components to perform holistic color image processing. Upon mapping each color channel inside a pixel into an imaginary component of a quaternion number, the color image may be processed as a single 2D signal – instead of three, one per color channel. The holistic (as opposed to separate) processing of the color signal allows to explore the correlation and coupling between the channels, and this is generally the advantage intended when using quaternion signal processing to handle three- or four-dimensional data (Took and Mandic 2008). Computationally, the spectral decomposition requires only two complex discrete Fourier transforms (DFTs), instead of three (one per color channel). Ever since, quaternion transforms have been employed not only on color image processing (Ell and Sangwine 2007; Chen et al. 2018b; Li 2013; Evans et al. 2000), but also on other tasks such as bivariate signal analysis (Flamant et al. 2017b; Flamant et al. 2017a; Flamant et al. 2018).

Algebra extensions change the signal representation by looking at the signal *samples* but, reminding the signal is nothing more than a function, one may also shift the attention to the *domain*. In fact, such a reflection has led to the creation of graph signal processing (GSP). This field, which emerged in the last decade, is concerned with extending the concepts of classical discrete signal processing (DSP) to the case in which the signal samples *reside in graph vertices*. It converges to the classical case of discrete-time domain when using a path or ring graph – depending on the graph shift operator (GSO) at hand, as will soon be clear – with equally weighted edges. The *signal*, therefore, is a mapping from the vertex set to the real (or complex) numbers, and the fixed sampling frequency is represented as graph edges having the same weights (i. e., as if the samples were at fixed “distances”, one could say). GSP deals with the consequences of connecting more vertices in this graph, moving from the queue-shaped domain into an arbitrary network. Many definitions from classical digital signal processing (DSP) have already found their way into GSP, thanks to the contributions of many scholars who proposed ways to implement filtering (Sandryhaila and Moura 2013b), linear transforms (Sandryhaila and Moura 2013c; Sardellitti et al. 2017), interpolation (Segarra et al. 2015), sampling theorem on graphs (Wang et al. 2015a; Chen et al. 2016; Tsitsvero et al. 2016), among others.

The endeavour to experiment with different algebras or domain topologies has primarily theoretical motivations, but often reveal new solutions to real world problems. Taking GSP for instance, its applications spread from coding (Su et al. 2017) and light field image super resolution schemes (Rossi and Frossard 2017), to regulatory genetic networks, in which graph-based methods have already improved three state-of-the-art network inference schemes (Pirayre et al. 2015; Pirayre et al. 2017), also reaching recommender systems, both collaborative filtering and content-based, using regularization of graph total variation (Benzi et al. 2016a), semi-supervised learning through adaptive graph filters (Chen et al. 2014a), community detection using graph wavelets (Tremblay and Borgnat 2014), among others. The latter examples hint at the intersection between GSP and the fields of data science and machine learning – well established and highly active, both in Academia and industry –, to the point where we can join Benjamin Ricaud and say that *Fourier could very well be, today, a data scientist* (Ricaud et al. 2019).

As happened to GSP, the study of quaternion-valued signal processing has also found applications. Beyond the already mentioned use in manipulating color image and bivariate signals, the work on adaptive quaternion filters has been useful, for example, in wind profile prediction (Jiang et al. 2014). Or yet the remarkable union of graphs and quaternions in the work by (Zhang et al. 2019) – the only one

the author is aware of –, in which quaternion embeddings are used to form a low dimensional representation of entities and relations in knowledge graphs.

In this context, the current thesis aims to tackle the following problem: is it possible to perform signal processing having quaternion-valued samples residing on graphs with quaternion-weighted edges? What limitations and new possibilites may arise when exploring what we may call *quaternion graph signal processing* (QGSP)? That is the spark which ignited the studies and results presented in the following chapters.

As it will become clear to the reader, however, the pathway to the basis of QGSP was not a straight line. Sometimes what seemed to be dead ends emerged, and during the reflections upon ways to bypass them, other interesting results started to appear. For example, it was certain from the beginning that the problem of eigendecomposing quaternion matrices would be at the heart of QGSP, since the very definition of a graph signal spectrum requires the eigenvalues and eigenvectors of a GSO (e.g. the adjacency matrix), which is a quaternion matrix. Since it took a while to make progress with general graph matrices, it seemed promising to investigate the quaternion discrete Fourier transform (QDFT); after all, ring graphs should have the graph Fourier transform (GFT) and the classical discrete Fourier transform (DFT) coinciding. By looking at the QDFT matrix, however, a new way to diagonalize it was found and a new multiparametric fractional QDFT was proposed. When shifting the focus to the graph domain, revisiting the study of graph shift operators and their spectral decomposition, interesting results regarding the fractional GSO were discovered. For instance, this new operator acts as a graph filter, and its polynomial representation was fully determined, demonstrating that it can be implemented as a linear and shift-invariant (LSI) graph filter. These unforeseen results are featured in the thesis, as they paved the way for the foundations of QGSP.

## 1.1 Organization

## 1.2 Contributions

The contributions of this doctorate research, featured in this thesis, are:

- The proposition and discussion of the fractional graph shift and its application on improving least-squares approximation of linear and shift-invariant ideal filters.

- A theorem which proves that the QDFT and the DFT share symmetric eigenvalues, and which enables a novel approach to fractionalizing the quaternion discrete Fourier transform (QDFT).
- The definition of a multiparametric fractional QDFT and its application in a novel image encryption scheme.
- The proposition of quaternion graph signal processing (QGSP), a new tool designed for specialized scenarios in which both the signal samples and their quantifiable relationship may be expressed as quaternions (thus having at most four dimensions or information sources).
- `gspx`<sup>1</sup>, an open-source Python library with implementation of the core concepts from QGSP.

## 1.3 Publications

The paper “*The Cosine Number Transform: A Graph Signal Processing Approach*”, presented in the 2019 edition of GlobalSIP, was created during short attempts to further generalize GSP to other algebraic structures, in particular to graphs with edge weights in finite fields. However, it has not evolved into more substantial findings and thus, although it stems from this work, it is not featured in this thesis. The full references of the mentioned publications are:

- Ribeiro, Guilherme and Lima, Juliano. “The Cosine Number Transform: A Graph Signal Processing Approach”. *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2019. DOI: 10.1109/GlobalSIP45357.2019.8969165
- Ribeiro, Guilherme B., and Lima, Juliano B. “Eigenstructure and fractionalization of the quaternion discrete Fourier transform”. *Optik* 208 (2020): 163957. DOI: 10.1016/j.ijleo.2019.163957.
- Ribeiro, Guilherme B., José R. De Oliveira Neto, and Lima, Juliano B. “On the Fractionalization of the Shift Operator on Graphs”. *IEEE Access* 10 (2022): 16468-16478. DOI: 10.1109/ACCESS.2022.3149755.
- Lima, Juliano, et al. *Processamento de Sinais sobre Grafos: Fundamentos e Aplicações*. Notas em Matemática Aplicada, v. 92. Sociedade Brasileira de Matemática Aplicada e Computacional (SBMAC), 2022.

---

<sup>1</sup>The library is available as an open repository: <https://github.com/gboaviagem/gspx>.

## 1.4 Notation

The use of symbols is a great way to communicate ideas concisely. However, instead of improving communication, they may make it cumbersome or even not communicate at all if the reader is not adequately familiar with them. Let us go through the main notation in this thesis, to avoid confusion instead of clarity.

Simple numbers, either real-, complex- or quaternion-valued, are represented in italics, as in  $q$ . Special treatment is given to unit pure quaternions, represented in bold italics, e.g.  $\mathbf{i}$ ,  $\mathbf{\nu}$ . Numerical sets (and the related fields or skew-fields) are denoted using blackboard bold face (such as  $\mathbb{R}$ ). Vectors and matrices are written with bold face: the former with lower case (e.g.  $\mathbf{v} \in \mathbb{R}^n$ ), and the latter with upper case (e.g.  $\mathbf{V} \in \mathbb{R}^{n \times n}$ ).

The function  $\text{diag}(\cdot)$  has context-dependent meaning: when the argument is a matrix, it returns the main diagonal as a column vector; whereas when it is a vector, it produces a diagonal matrix with this vector in its main diagonal. The symbol  $\stackrel{\Delta}{=}$  represents *equality by definition*.

The *symplectic decomposition* of a quaternion number  $q$  (or, by extension, of a quaternion-valued vector or matrix) will be *usually* represented as  $q = q_1 + q_2\mathbf{j}$ , with subscripts 1 and 2 indicating the simplex and perplex components. When there is risk of mistaking the subscripts for indices (e.g. in summation), the alternative (more bulky) notation will employ superscripts  $(s)$  and  $(p)$ , as in  $q = q^{(s)} + q^{(p)}\mathbf{j}$ .



# A review on quaternion algebra and its Fourier transform

*And here there dawned on me the notion that we must admit, in some sense, a fourth dimension of space for the purpose of calculating with triples... An electric circuit seemed to close, and a spark flashed forth.*

— Sir William Hamilton.

In 1833, at the age of 28, Willian Rowan Hamilton presented to the Royal Irish Academy (RIA) a work in which complex numbers were treated as ordered pairs of real numbers, given the appropriate definition of operations<sup>1</sup>. In the following years, he struggled to extend the complex field into a normed division algebra over triples, but soon realized that, as much as his attempts were inventive, the resulting algebra<sup>2</sup> was not closed under multiplication. We can see this through a simple example (Santos and Ferreira 2011): let it be the set  $\mathbb{F} = \{a + bi + cj \mid (a, b, c) \in \mathbb{R}^3\}$ , with  $i^2 = j^2 = -1$  and  $i \neq j$ . Since  $i, j \in \mathbb{F}$ , so there should exist  $x, y, z \in \mathbb{R}$  so that

$$ij = x + yi + zj. \quad (2.1)$$

Multiplying by  $i$  both sides of the equation,

$$i^2j = ix + i^2y + z(ij), \quad (2.2)$$

and using (2.1) it yields,

$$-j = ix - y + z(x + yi + zj) \iff (zx - y) + i(x + zy) + j(z^2 + 1) = 0. \quad (2.3)$$

That is:  $z \notin \mathbb{R}$  and  $ij \notin \mathbb{F}$ , proving that such algebraic structure is not closed under multiplication.

---

<sup>1</sup>The results were published in 1837, in the paper *Theory of Conjugate Functions, or Algebraic Couples; with a Preliminary and Elementary Essay on Algebra as the Science of Pure Time* (Hamilton 1837).

<sup>2</sup>An *algebra over a field*, or simply *algebra*, is a vector space over a field with a bilinear multiplication (that is, the multiplication distributes over the addition and the associativity is valid for multiplication) (Schafer 1955).

Only a decade later, in 1843, while walking by the roads in Dublin towards the RIA, “an electric circuit seemed to close, and a spark flashed forth”, as he would say. He had conceived the four-dimensional structure required to the desired algebra, creating the quaternions. Moved by excitement, he craved on the stone below Broome Bridge, in Cabra (Dublin), the equations that define the relations between the canonical basis elements of quaternions<sup>3</sup>. This creation, made possible by an insight in 1843, is found across most of this work. The following sections lead the reader through the foundations of quaternion algebra and quaternion signal analysis.

## 2.1 Introduction to the quaternion algebra

Quaternions are numbers  $q \in \mathbb{H}$  in the form

$$q = a + bi + cj + dk, \quad (2.4)$$

in which  $a, b, c, d \in \mathbb{R}$ , holding true the fundamental relations:

$$i^2 = j^2 = k^2 = ijk = -1. \quad (2.5)$$

The multiplication rules between  $i$ ,  $j$  and  $k$  follow directly from (2.5), resembling those between orthonormal basis vectors from  $\mathbb{R}^3$  and the vector product: the product between two of them yields the third, the sign being determined from the operands order. For instance, to find the result of  $ij$  one may start from (2.5) and write

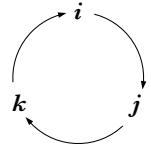
$$\begin{aligned} ijk &= -1 \\ ij \underbrace{kk}_{=-1} &= -k \\ ij &= k. \end{aligned} \quad (2.6)$$

Similarly, to find  $ji$ ,

$$\begin{aligned} ijk &= -1 \\ iijk &= -i \\ -jk &= -i \\ jjk &= ji \\ -k &= ji. \end{aligned} \quad (2.7)$$

---

<sup>3</sup>Close to the original site of the inscriptions, the RIA placed a commemorative plaque in 1958, with the same writings.



**Fig. 2.1.:** Illustration of the multiplication rule between the imaginary units  $i$ ,  $j$  and  $k$ .

Fig. 2.1 depicts the order in which the product between any pair in the triplet  $i$ ,  $j$  and  $k$  yields the third one, with positive sign. All three units commute with real numbers. The most relevant consequence, therefore, of (2.5), is that the quaternion product is *noncommutative*. In fact, it is the first example of noncommutative normed division algebra in history (Kleiner 2007).

The exploration of the quaternion multiplication rules gives rise to a useful property, related to products between complex numbers and elements from the quaternion canonical basis. In particular, the product between  $j$  and any complex number  $x = a + bi$ ,  $a, b \in \mathbb{R}$ , satisfies  $jk = \bar{x}j$ , since

$$\begin{aligned} jx &= j(x_r + x_i i) = x_r j + x_i j i = x_r j - x_i i j \\ &= (x_r - x_i i) j = \bar{x} j. \end{aligned} \tag{2.8}$$

This will come up a couple of times during manipulations in this thesis.

The elements  $i$ ,  $j$  e  $k$  may be perceived as orthogonal imaginary units, generating the 3D space of quaternion *imaginary parts*,

$$\mathbf{V}(q) = bi + cj + dk. \tag{2.9}$$

The *real part* of a quaternion  $q$  is defined as

$$S(q) = a. \tag{2.10}$$

The imaginary part is usually referred to as *vector part*, while the real part may also be called *scalar part*. A quaternion with null real part is said to be *pure* — the set of which is represented by  $\mathbf{V}(\mathbb{H})$ . The product between pure quaternions may be written similarly to that between  $\mathbb{R}^3$  vectors, i. e., if  $v_1 = b_1 i + c_1 j + d_1 k$  and  $v_2 = b_2 i + c_2 j + d_2 k$ , then

$$v_1 \times v_2 = \begin{vmatrix} i & j & k \\ b_1 & c_1 & d_1 \\ b_2 & c_2 & d_2 \end{vmatrix}. \tag{2.11}$$

Based on the similarity between the  $\mathbb{R}^3$  set, equipped with the cross product, and the set of pure quaternions, equipped with their quaternion product, it is usual to refer to  $\mathbf{i}$ ,  $\mathbf{j}$  and  $\mathbf{k}$  as *axis*. The term references the interpretation of these imaginary units as being coordinate axis in the 3D space of pure quaternions.

The analogy with the  $\mathbb{R}^3$  vector operations also extends to the definition of inner product between pure quaternions  $v_1 = b_1\mathbf{i} + c_1\mathbf{j} + d_1\mathbf{k}$  and  $v_2 = b_2\mathbf{i} + c_2\mathbf{j} + d_2\mathbf{k}$ :

$$\langle v_1, v_2 \rangle = b_1b_2 + c_1c_2 + d_1d_2. \quad (2.12)$$

The quaternion multiplication distributes over quaternion addition and is also associative. That is, if  $q_1 = a_1 + b_1\mathbf{i} + c_1\mathbf{j} + d_1\mathbf{k}$  and  $q_2 = a_2 + b_2\mathbf{i} + c_2\mathbf{j} + d_2\mathbf{k}$ , then their sum is simply

$$q_1 + q_2 = (a_1 + a_2) + (b_1 + b_2)\mathbf{i} + (c_1 + c_2)\mathbf{j} + (d_1 + d_2)\mathbf{k}, \quad (2.13)$$

whereas their product is

$$\begin{aligned} q_1q_2 &= (a_1 + b_1\mathbf{i} + c_1\mathbf{j} + d_1\mathbf{k})(a_2 + b_2\mathbf{i} + c_2\mathbf{j} + d_2\mathbf{k}) \\ &= (a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2) \\ &\quad + \mathbf{i}(b_1a_2 + a_1b_2 - d_1c_2 + c_1d_2) \\ &\quad + \mathbf{j}(c_1a_2 + d_1b_2 + a_1c_2 - b_1d_2) \\ &\quad + \mathbf{k}(d_1a_2 - c_1b_2 + b_1c_2 + a_1d_2). \end{aligned} \quad (2.14)$$

Finally, it is possible to write the quaternion product in (2.14) as a function of the operands scalar and vector parts,

$$\begin{aligned} q_1q_2 &= S(q_1)S(q_2) - \langle \mathbf{V}(q_1), \mathbf{V}(q_2) \rangle \\ &\quad + S(q_1)\mathbf{V}(q_2) + S(q_2)\mathbf{V}(q_1) + \mathbf{V}(q_1) \times \mathbf{V}(q_2). \end{aligned} \quad (2.15)$$

The lack of commutativity in the cross product is another demonstration of the fact that quaternion multiplication is noncommutative.

Quaternion conjugation, as defined in the complex numbers, is obtained changing the sign of the imaginary part:  $\bar{q} \triangleq S(q) - \mathbf{V}(q)$ . The quaternion *norm* is defined as  $\|q\| \triangleq a^2 + b^2 + c^2 + d^2 = q\bar{q} = \bar{q}q$ , whereas the *modulus* of  $q$  (Ell et al. 2014) is

$$|q| \triangleq \sqrt{a^2 + b^2 + c^2 + d^2} = \|q\|^{1/2}. \quad (2.16)$$

From the definition of quaternion conjugation, the product between the conjugate of two quaternions  $q$  and  $p$  equals

$$\begin{aligned}\bar{q}\bar{p} &= [S(q) - \mathbf{V}(q)] [S(p) - \mathbf{V}(p)] \\ &= S(q)S(p) - \mathbf{V}(q)S(p) - S(q)\mathbf{V}(p) + \mathbf{V}(q)\mathbf{V}(p) \\ &= S(q)S(p) - S(p)\mathbf{V}(q) - S(q)\mathbf{V}(p) + \mathbf{V}(q)\mathbf{V}(p).\end{aligned}\quad (2.17)$$

From (2.15),

$$\mathbf{V}(q)\mathbf{V}(p) = \mathbf{V}(q) \times \mathbf{V}(p) - \langle \mathbf{V}(q), \mathbf{V}(p) \rangle = -\mathbf{V}(p)\mathbf{V}(q), \quad (2.18)$$

therefore

$$\bar{q}\bar{p} = S(q)S(p) - S(p)\mathbf{V}(q) - S(q)\mathbf{V}(p) - \mathbf{V}(p)\mathbf{V}(q) = \bar{p}\bar{q}, \quad (2.19)$$

showing that  $\bar{q}\bar{p} = \bar{p}\bar{q}$ . This result allows to prove that  $\|pq\| = \|p\|\|q\|$ , since

$$\|pq\| = \|\bar{p}\bar{q}pq\| = \|\bar{q}\bar{p}pq\| = \|\bar{q}\|\|p\|\|q\| = \|p\|\|\bar{q}q\| = \|p\|\|q\|. \quad (2.20)$$

A *unit* quaternion has, by definition, unit norm, and the norm definition also leads to the quaternion multiplicative inverse (if  $q \neq 0$ ),

$$q^{-1} = \frac{\bar{q}}{\|q\|}. \quad (2.21)$$

From the analogy between pure quaternions and the elements in  $\mathbb{R}^3$ , the idea of perpendicularity between pure quaternions presents itself naturally. Given  $\mu, \nu \in \mathbf{V}(\mathbb{H})$ , they are said orthogonal — we write  $\mu \perp \nu$  — if and only if

$$S(\mu\nu) = \langle \mu, \nu \rangle = 0. \quad (2.22)$$

For two orthogonal unit pure quaternions  $\mu$  and  $\nu$ , it follows from (2.15) that  $\mu\nu = \mu \times \nu$ , therefore  $\mu\nu \perp \mu$  and  $\mu\nu \perp \nu$ . Since  $(\mu, \nu, \mu\nu)$  is a triplet of orthogonal unit pure quaternions, they form a basis for  $\mathbf{V}(\mathbb{H})$ . Hence it is possible to rewrite (2.4) as

$$q = a + b'\mu + c'\nu + d'\mu\nu, \quad (2.23)$$

$a, b', c', d' \in \mathbb{R}$ , which represents the so called *generalized quaternion*. The *classical Hamiltonian quaternions* are those written in terms of the canonical basis  $(1, \mathbf{i}, \mathbf{j}, \mathbf{k})$ .

Besides the cartesian representation (2.4), quaternions allow the so called *Euler form* (Ell et al. 2014), a polar representation commonly expressed as

$$q = |q|e^{\mu\theta} = |q|\cos\theta + |q|\mu \sin\theta, \quad (2.24)$$

in which  $\mu$  is a unit pure quaternion, parallel to the vector part of  $q$ .

As a remarkable consequence of (2.24), it follows that every unit pure quaternion is a square root of  $-1$ . For instance, let  $\nu$  be a unit pure quaternion. From (2.24),

$$\nu = |\nu|e^{\nu\theta}, \quad (2.25)$$

but since  $|\nu| = 1$ , then

$$\nu = e^{\nu\theta} = \cos\theta + \nu \sin\theta \Rightarrow \theta = \frac{\pi}{2}, \quad (2.26)$$

hence,

$$\nu^2 = \left(e^{\nu\frac{\pi}{2}}\right)^2 = e^{\nu\pi} = \cos\pi + \nu \sin\pi = -1. \quad (2.27)$$

This property leads to the conclusion that numbers such as  $a + \mu b$  form an isomorphism with the complex numbers. For this reason, the set composed by such numbers is represented by  $\mathbb{C}_\mu \triangleq \{a + \mu b \mid a, b \in \mathbb{R}\}$  (and hence  $\mathbb{C}_i$  can be identified, for all practical ends, as the usual set of complex numbers, by making the extrapolation that the unit pure quaternion  $i$  equals the complex imaginary unit).

Another crucial representation of quaternion numbers is called *symplectic decomposition*. Every quaternion  $q = a + bi + cj + dk \in \mathbb{H}$  can be represented as

$$q = q_1 + q_2 i, \quad q_1, q_2 \in \mathbb{C}_i, \quad (2.28)$$

in which the complex numbers  $q_1 = a + bi$  and  $q_2 = c + di$  are commonly named *simplex* and *perplex* parts, respectively. In general, the decomposition does not need to take  $i$  as reference axis. Take for example  $\mu$  and  $\nu$  as arbitrary orthogonal unit pure quaternions, hence any quaternion  $q$  may be decomposed as

$$q = q_1 + q_2 \nu, \quad q_1, q_2 \in \mathbb{C}_\mu. \quad (2.29)$$

That is a hugely important tool when it comes to computing the QDFT (see Section 2.3) and handling quaternion matrices (cf. Chapter 6).

### 2.1.1 Quaternion similarity and rotation

Two quaternions  $q$  and  $r$  are said to be *similar* if it exists a non-zero quaternion  $v$  such that  $v^{-1}qv = r$ . In this case, one can write  $q \sim r$ . Similarity between quaternions constitutes an equivalence relation (Zhang 1997), and all elements from a same similarity class possess the same norm, since, from (2.20),  $|v^{-1}qv| = |v^{-1}| \cdot |q| \cdot |v| = |q|$ .

It matters to notice that an important property of quaternion similarity transformations, which justifies its great use in mechanics and graphics computing industry, is that it performs a rotation on the  $ijk$  space. Given  $v, q \in \mathbb{H}$ ,  $v = |v|e^{\mu\alpha}$ , the similarity transformation

$$\phi_v(q) = vqv^{-1} \quad (2.30)$$

produces the rotation of the vector part of  $q$  along the axis  $\mu$  (which is parallel to the vector part of  $v$ ) through an angle  $2\alpha$  (Ward 2012), following the right-hand rule. Let us illustrate this property with an example.

**Example 1.** Let  $\lambda = 3i + k$  and

$$\mathbf{v} = \begin{pmatrix} 1+i \\ 2j+k \end{pmatrix} \quad (2.31)$$

be, respectively, an eigenvalue and its eigenvector of a matrix  $\mathbf{A} \in \mathbb{H}^{2 \times 2}$ . The eigenvalue problem over the quaternions will be introduced in the next section, so let us not consider its details for now. It suffices to know that  $\lambda$ ,  $\mathbf{A}$  and  $\mathbf{v}$  satisfy the equation

$$\mathbf{Av} = \mathbf{v}\lambda. \quad (2.32)$$

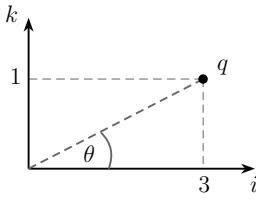
Let us restrict the first column of  $\mathbf{A}$  to be equal to  $(1 \ 1)^T$ , so that the matrix is now fully determined,

$$\mathbf{A} = \begin{pmatrix} 1 & -\frac{1}{5} + \frac{3}{5}i + 2j \\ 1 & -3i + j \end{pmatrix}. \quad (2.33)$$

Given an invertible quaternion  $q$ , the equation may be rewritten as

$$\mathbf{A}(\mathbf{v}q) = (\mathbf{v}q)q^{-1}\lambda q, \quad (2.34)$$

which presents the similarity transformation  $q^{-1}\lambda q$ . Since similar quaternions differ only by a rotation of their vector parts, it is clearly possible



**Fig. 2.2.:** Representation of  $\lambda = 3\mathbf{i} + \mathbf{k}$  in the  $ik$  plane.

to find a quaternion  $q$  so that the vector part of  $q^{-1}\lambda q$  is parallel to  $\mathbf{i}$  — that is,  $q^{-1}\lambda q$  is a complex number. Let us go through that process.

Since the vector part of  $\lambda$  belongs to the  $ik$  plane, the idea is to use (2.30) to rotate  $\lambda$  along the  $j$  axis (orthogonal to the rotation plane) by an angle of  $\theta = \tan^{-1} \frac{1}{3}$  radians, towards  $\mathbf{i}$  (see Fig. 2.2). The required quaternion to enable such rotation is, therefore,

$$v = e^{j\alpha}, \quad 2\alpha = \theta = \tan^{-1} \frac{1}{3}, \quad (2.35)$$

using the mapping  $\lambda \mapsto v\lambda v^{-1}$  in (2.30).

Recalling that our desired transformation is written as  $\lambda \mapsto q^{-1}\lambda q$ , then

$$q = v^{-1} = e^{-j\alpha} = e^{-j\frac{\theta}{2}}. \quad (2.36)$$

As a sanity check, let us find the value of  $\alpha$  which results in  $q^{-1}\lambda q \in \mathbb{C}_i$ . Since  $v = q^{-1} = \cos \alpha + j \sin \alpha$ ,

$$\begin{aligned} q^{-1}\lambda &= (\cos \alpha + j \sin \alpha)(3\mathbf{i} + \mathbf{k}) \\ &= \mathbf{i}(3 \cos \alpha + \sin \alpha) + \mathbf{k}(\cos \alpha - 3 \sin \alpha). \end{aligned} \quad (2.37)$$

$$\begin{aligned} q^{-1}\lambda q &= [\mathbf{i}(3 \cos \alpha + \sin \alpha) + \mathbf{k}(\cos \alpha - 3 \sin \alpha)](\cos \alpha - j \sin \alpha) \\ &= \mathbf{i}(3 \cos 2\alpha + \sin 2\alpha) + \mathbf{k}(\cos 2\alpha - 3 \sin 2\alpha). \end{aligned}$$

Therefore,  $q^{-1}\lambda q \in \mathbb{C}_i$  if and only if

$$\begin{aligned} \cos 2\alpha - 3 \sin 2\alpha &= 0 \\ \Rightarrow 2\alpha &= \tan^{-1} \frac{1}{3}, \end{aligned} \quad (2.38)$$

as determined by (2.35).

Consequently, it was demonstrated that the unit pure quaternion  $q = e^{-j\alpha}$ , with  $\alpha = -1/2 \tan^{-1} 1/3$ , produces, through a similarity

transformation, the complex-valued eigenvalue  $q^{-1}\lambda q \in \mathbb{C}_i$ . Since  $q = \cos \alpha - \mathbf{j} \sin \alpha$ , (2.32) can be rewritten as in (2.34),

$$\mathbf{A} \underbrace{\begin{pmatrix} \cos \alpha + \mathbf{i} \sin \alpha - \mathbf{j} \sin \alpha - \mathbf{k} \sin \alpha \\ 2 \sin \alpha + \mathbf{i} \sin \alpha + \mathbf{j} 2 \cos \alpha + \mathbf{k} \cos \alpha \end{pmatrix}}_{=\mathbf{v}q} = (\mathbf{v}q) \cdot \underbrace{\mathbf{i}(3 \cos 2\alpha + \sin 2\alpha)}_{=q^{-1}\lambda q}. \quad (2.39)$$

Let us leave a remark for the user, as an anticipation of the upcoming section: notice how  $\mathbf{v}$  and  $\mathbf{v}q$  are the same eigenvector up to a scaling factor, and it is always possible to make  $q^{-1}\lambda q$  into a complex number through a similarity transformation, given that the vector part of  $\lambda$  has non-zero norm.

## 2.2 On the theory of quaternion matrices

When analyzing the eigenstructure and subsequent fractionalization of the QDFT matrix, the eigendecomposition of the DFT and the eigenvector sharing served as a convenient shortcut. In order to build the results in QGSP, however, it is required to dive into more general properties of quaternion matrices. The symplectic decomposition, already presented in (2.29), plays an important role in that matter, specially for its use in defining the *complex adjoint matrix*.

**Definition 1** (Complex adjoint matrix (Zhang 1997)). *Given  $\mathbf{A} \in \mathbb{H}^{n \times n}$ , with symplectic decomposition  $\mathbf{A}_1 + \mathbf{A}_2\mathbf{j}$ ,  $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{C}^{n \times n}$ , its complex adjoint matrix is defined as*

$$\chi_A \triangleq \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ -\overline{\mathbf{A}}_2 & \overline{\mathbf{A}}_1 \end{pmatrix}. \quad (2.40)$$

In the following theorems, Zhang brings fundamental results for building QGSP.

**Theorem 1** (Part of Theorem 4.2 in (Zhang 1997) ). *Given the matrix  $\mathbf{A} \in \mathbb{H}^{n \times n}$ , the following sentences are equivalent:*

- $\chi_{AB} = \chi_A \chi_B$ ,
- $\chi_{A^{-1}} = \chi_A^{-1}$ , if  $\mathbf{A}^{-1}$  exists,
- $\chi_A$  is unitary, Hermitian or normal if and only if so is  $\mathbf{A}$ .

**Theorem 2** (Part of Theorem 4.3 in (Zhang 1997) ). *Given the matrix  $\mathbf{A} \in \mathbb{H}^{n \times n}$ , the following sentences are equivalent:*

- $\mathbf{A}$  is invertible.
- $\det(\chi_A) \neq 0$ , i. e.,  $\chi_A$  is invertible.

Differently from other representations of quaternion matrices in  $\mathbb{C}$  or  $\mathbb{R}$ , the complex adjoint allows to establish an important relation between the spectra of  $\chi_A$  and  $\mathbf{A}$ , as will soon be discussed.

## 2.2.1 Eigenvalues

Since quaternion multiplication is noncommutative, it is necessary to distinguish between *left* and *right* eigenvalues of a given matrix  $\mathbf{A} \in \mathbb{H}^{n \times n}$ ,

$$\begin{aligned}\mathbf{A}\mathbf{v} &= \mathbf{v}\lambda, && \text{(right)} \\ \mathbf{A}\mathbf{v} &= \lambda\mathbf{v}. && \text{(left)}\end{aligned}$$

We will restrict this discussion mostly to right eigenvalues, since they hold a broader set of results in literature (Zhang 1997, Cap. 5) and are, for that matter, a safer point of support when developing QGSP. When not explicitly mentioned, the right eigenvalues will simply be referred to as *eigenvalues* of the quaternion matrix.

It matters to highlight the fact that a quaternion matrix possesses a *finite* number of eigenvalues if and only if they are all real-valued. Otherwise, each of them will belong to a similarity class, according to the transformation  $\lambda_1 = q^{-1}\lambda_2 q$ , containing *infinite* quaternions which are also eigenvalues of the same matrix (return to Example 1, for instance). Let  $\lambda$  be an eigenvalue of the matrix  $\mathbf{A}$  associated with the eigenvector  $\mathbf{v}$ . That is,

$$\begin{aligned}\mathbf{A}\mathbf{v} &= \mathbf{v}\lambda \\ \mathbf{A}\mathbf{v}q &= \mathbf{v}\lambda q = \mathbf{v}qq^{-1}\lambda q \\ \mathbf{A}(\mathbf{v}q) &= (\mathbf{v}q)q^{-1}\lambda q,\end{aligned}\tag{2.41}$$

so that  $q^{-1}\lambda q$  is an eigenvalue associated with the eigenvector  $\mathbf{v}q$ , with  $q \in \mathbb{H}^*$ .

## 2.2.2 Diagonalizability

The problem of quaternion matrix diagonalization differs greatly from the case with complex matrices. In the latter scenario, having no pair of equal eigenvalues is a sufficient condition for diagonalizability, but the rule does not hold for the quaternion case. See for instance the counterexample given by Zhang (Zhang 1997, Exemplo 7.4), the matrix

$$\mathbf{A} = \begin{pmatrix} i & 1 \\ 0 & j \end{pmatrix}. \quad (2.42)$$

Although its eigenvalues are distinct –  $i$  and  $j$ , since they lie on the main diagonal of an upper triangular matrix –, the respective eigenvectors do not constitute a linearly independent set. Notice, for example, that an eigenvector associated with  $i$  is  $(1, 0)^T$ , whereas one associated with  $j$  is  $(i + j, 0)^T$ . The reason as to why they form a linearly dependent set is the fact previously stated that, for each eigenvalue and eigenvector  $\lambda$  and  $\mathbf{v}$  from a quaternion matrix, it follows that  $q^{-1}\lambda q$  and  $\mathbf{v}q$  form another eigenvalue-eigenvector pair, with  $q \in \mathbb{H}^*$ . That is, two *similar* eigenvalues are always associated with the same eigenvector (up to a scaling factor). That is why similar quaternions are said to belong to the same *eigenclass* (De Leo and Scolarici 2000). In order to have a linearly independent set of eigenvectors, therefore, their eigenvalues must not be similar. Concluding the counterexample, the reader may show that  $i$  and  $j$  satisfy the similarity relation  $j = qiq^{-1}$  whenever  $q$  belongs to the set  $\{a - ai - aj + ak \mid a \in \mathbb{R}^*\}$ .

Theorem 3 relates the eigendecomposition of a quaternion matrix to that of its complex adjoint, providing a fundamental principle to the study of quaternions matrices eigenstructure.

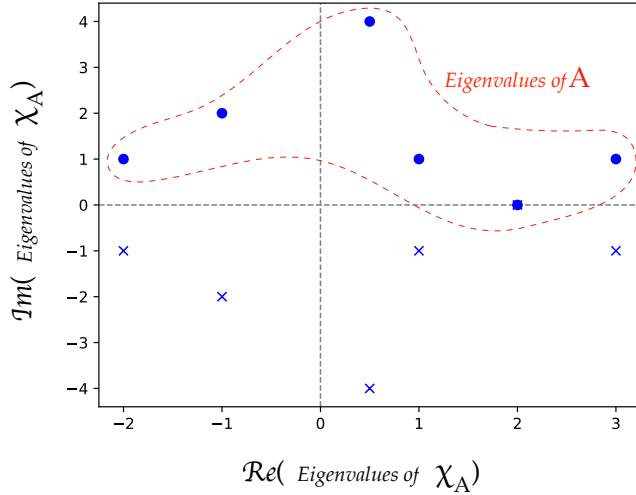
**Theorem 3** ((Zhang 1997)). *Every matrix  $A \in \mathbb{H}^{n \times n}$  has exactly  $n$  right eigenvalues which are complex numbers with nonnegative imaginary parts. These are called the **standard eigenvalues** of  $\mathbf{A}$  and are a subset of the  $2n$  eigenvalues of  $\chi_A$ .*

*Proof.* As demonstrated in the Appendix A, starting from the quaternion eigenvalue equation

$$\mathbf{Av} = \mathbf{v}\lambda, \quad (2.43)$$

in which one can assume without loss of generality that  $\lambda \in \mathbb{C}$ , it is always possible to arrive at the *equivalent* equation

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ -\overline{\mathbf{A}}_2 & \overline{\mathbf{A}}_1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1 \\ -\overline{\mathbf{v}}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{v}_1 \\ -\overline{\mathbf{v}}_2 \end{pmatrix} \lambda, \quad (2.44)$$



**Fig. 2.3.:** Standard eigenvalues of a quaternion matrix  $\mathbf{A}$ , indicated as a subset of the eigenvalues of the complex adjoint matrix.

which involves solely complex-valued matrices and vectors, namely: the components of the symplectic decomposition of  $\mathbf{A}$  and  $\mathbf{v}$ . The equivalence between both equations implies that  $\mathbf{A}$  and its complex adjoint

$$\chi_{\mathbf{A}} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ -\overline{\mathbf{A}}_2 & \overline{\mathbf{A}}_1 \end{pmatrix} \quad (2.45)$$

share all their complex-valued eigenvalues (notice that  $\lambda$  remains unchanged as we move from one equation to its equivalent). As it appears, the eigendecomposition of the complex adjoint seems a promising method for finding some eigenvalues of the quaternion matrix.

Is there, however, any possibility of leaving some eigenvalues behind? In other words, is it possible to exist some eigenvalue of  $\mathbf{A}$  which does not belong to an eigenclass determined by a complex eigenvalue of  $\chi_{\mathbf{A}}$ ? No, it is not, because it is always possible to align a quaternion's vector part with the  $i$  axis through a single rotation (see Section 2.1.1). In other words, all (quaternion-valued) eigenvalues of  $\mathbf{A}$  are similar to a complex number which, as discussed, must also be an eigenvalue of  $\chi_{\mathbf{A}}$ .

Since  $\chi_{\mathbf{A}}$  is a  $2n \times 2n$ -complex matrix, it has  $2n$  complex-valued eigenvalues (possibly repeated). The reason why  $\chi_{\mathbf{A}}$  share  $2n$  eigenvalues with  $\mathbf{A}$ , even though  $\mathbf{A}$  is an  $n \times n$  matrix is, once more, due to the existence of eigenclasses.

According to (Lee 1948, Theorem 5), the complex adjoint eigenvalues appear as  $n$  conjugate pairs (possibly some pairs are real-valued, so they are actually

identical). Let us say that  $2m$  of those eigenvalues are real, leaving  $2(n - m)$  complex eigenvalues. Let  $q \in \mathbb{C}_i$  belong to the latter group. It can be shown immediately that  $q \sim \bar{q}$ , since  $\bar{q}$  is obtained through a  $180^\circ$  rotation of the vector part of  $q$  around the  $j$  axis. This similarity transformation was already shown in (2.8), written as  $jx = \bar{x}j$ , or equivalently  $jx(-j) = \bar{x}$ . In the format of (2.30),  $q \mapsto vqv^{-1} = \bar{q}$  with  $v = e^{j\frac{\pi}{2}} = j$ .

Since two conjugate eigenvalues of  $\chi_A$  are always similar, they belong to the same eigenclass and are therefore associated with the same set of linearly dependent eigenvectors. For that reason, by convention one can take the eigenvalues of  $\chi_A$  and define the standard eigenvalues of  $\mathbf{A}$  as being those complex-valued eigenvalues with positive imaginary part and half those real-valued ones.  $\square$

The following theorem is another fundamental result relating the eigenstructure of a quaternions matrix and that of its complex adjoint.

**Theorem 4** (Theorem 7.4 in (Zhang 1997)). *Given the matrices  $\mathbf{A}, \mathbf{B} \in \mathbb{H}^{n \times n}$ , then  $\mathbf{A}$  is similar to  $\mathbf{B}$  if and only if  $X\chi_A$  is similar to  $\chi_B$ .*

**Corollary 1.** *A matrix  $\mathbf{A} \in \mathbb{H}^{n \times n}$  is diagonalizable if and only if  $\chi_A$  is diagonalizable.*

*Proof.* If  $\mathbf{A} \in \mathbb{H}^{n \times n}$  is diagonalizable, then it is similar to a diagonal matrix  $\Lambda \in \mathbb{C}_i^{n \times n}$  containing its standard eigenvalues in the main diagonal. From Theorem 4, it follows that  $\chi_A$  is similar to

$$\chi_\Lambda = \begin{pmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \bar{\Lambda} \end{pmatrix}, \quad (2.46)$$

which is also a diagonal matrix. Therefore,  $\chi_A$  is diagonalizable.

On the other hand, if  $\chi_A$  is diagonalizable, then it is similar to a diagonal matrix containing its  $2n$  eigenvalues, which appear in  $n$  conjugate complex pairs. Therefore, its eigenvalues matrix may be written as (2.46), what implies, from Theorem 4, that  $\mathbf{A}$  is similar to  $\Lambda$ .  $\square$

## 2.3 The quaternion Fourier transform

The field of quaternion signal processing has leveraged many tools to transform signals with quaternion-valued samples, from the fundamental redefinition of the gradient operator (Jiang et al. 2014) up to algorithms for adaptive filtering

(Jiang et al. 2013). This section focuses on the quaternion Fourier transform (QFT), as it is the basis for spectral analysis of quaternion signals.

Quaternion Fourier transforms have received quite a few definitions. Some are one-dimensional (Flamant et al. 2017b), while others are intrinsically two-dimensional (Guanlei et al. 2008); the latter group can yet be divided into those having kernels oriented towards generic unit pure quaternions, and those using canonic imaginary units, such as  $i$  and  $j$ . The 2D-transformed signals may be placed *between* the two kernels or beside them. In fact, Ell (Ell et al. 2014, sec. 3.2) lists 8 possibilities for the 2D-QFTs. The two dimensional case will not be covered for now, since the discussion on the 1D case already serves the purpose of introducing the main ideas and properties of this family of transforms.

Let  $f$  be a quaternion-valued function  $f : \mathbb{R} \rightarrow \mathbb{H}$  and  $\mu \in \mathbf{V}(\mathbb{H})$ ,  $\mu^2 = -1$ . The *left* 1D QFT can be defined as the family of integral transforms

$$\mathcal{F}_{\mp\mu}^L[f](\omega) = F_{\mp\mu}^L(\omega) \stackrel{\text{def}}{=} \kappa_- \int_{-\infty}^{\infty} e^{\mp\mu\omega t} f(t) dt. \quad (\text{left 1D QFT})$$

It can be proven that the inverse transform exists and is given by

$$\mathcal{F}_{\pm\mu}^{-L}[F^L](t) = f(t) = \kappa_+ \int_{-\infty}^{\infty} e^{\pm\mu\omega t} F^L(\omega) d\omega. \quad (\text{inverse left 1D QFT})$$

In the expressions above, the unit pure quaternion  $\mu$  is called the *eigenaxis* of the transform kernel, we could say it is the reference imaginary unit. The sign in the exponential is arbitrary, as long as the direct and inverse transforms use different signs. The real constants  $\kappa_-$  and  $\kappa_+$  satisfy

$$\kappa_+ \kappa_- = \frac{1}{2\pi}, \quad (2.47)$$

and when  $\kappa_- = \kappa_+$ , the transform is said to be unitary.

The right QFT may be defined likewise, by simply switching the relative positions of the kernel and the operand  $f(t)$ . The reader may notice that, when the eigenaxis is  $i$ , the QFT coincides with the usual continuous-time Fourier transform.

In this work, however, the most relevant version of the QFT is that in which both the original signal and its transformed representation are defined over discrete domains: the 1D quaternion discrete Fourier Transform (QDFT) with axis  $\mu$ , as

defined in (Ell et al. 2014, sec. 3.3.1). If  $\mu$  is a unit pure quaternion, the  $m$ -th component of the transformed vector in the left unitary QDFT with axis  $\mu$  is

$$\hat{v}_m = \text{QDFT}\{\mathbf{v}\}_m \triangleq \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \exp\left(-\mu \frac{2\pi}{N} nm\right) v_n \in \mathbb{C}_\mu, \quad (2.48)$$

with inverse transform given by

$$v_n = \text{QDFT}^{-1}\{\hat{\mathbf{v}}\}_n = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} \exp\left(\mu \frac{2\pi}{N} nm\right) \hat{v}_m. \quad (2.49)$$

The transform analysis and synthesis equations may also be written in matrix form:

$$\hat{\mathbf{v}} = \text{QDFT}\{\mathbf{v}\} = \mathbf{F}\mathbf{v}, \quad (2.50)$$

$$\mathbf{v} = \text{QDFT}^{-1}\{\hat{\mathbf{v}}\} = \mathbf{F}^{-1}\hat{\mathbf{v}}, \quad (2.51)$$

in which  $\mathbf{F}$  is the unitary transform matrix, with entries  $\{\mathbf{F}\}_{n,m} = \sqrt{N}^{-1} \exp\left(-\mu \frac{2\pi}{N} nm\right)$ . Since  $\exp\left(-\mu \frac{2\pi}{N}\right)$  is a  $N$ -th root of unity, just like  $\exp\left(-i \frac{2\pi}{N}\right)$ , it follows that  $\mathbf{F}$  shares many properties with the usual discrete Fourier transform (DFT) matrix, among which the invertibility, validating the existence of (5.2) and (5.4).

As it was done in previous pages, the symplectic decomposition may be applied to each entry of a quaternion matrix, generating complex-valued *simplex* and *perplex* matrices<sup>4</sup>. Using this reasoning, Ell e Sangwine (Ell et al. 2014) demonstrated that the QDFT of a signal  $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}] \in \mathbb{H}^N$  could be easily computed using two (complex) DFT, by decomposing each signal sample along the QDFT eigenaxis:

$$\begin{aligned} \text{QDFT}\{\mathbf{x}\}_m &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \exp\left(-\mu \frac{2\pi}{N} nm\right) x_n \\ &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \exp\left(-\mu \frac{2\pi}{N} nm\right) (x_n^{(s)} + x_n^{(p)} \boldsymbol{\nu}) \\ &= \text{DFT}_\mu\{\mathbf{x}^{(s)}\}_m + \text{DFT}_\mu\{\mathbf{x}^{(p)}\}_m \boldsymbol{\nu}, \end{aligned} \quad (2.52)$$

in which  $\text{DFT}_\mu$  indicates the DFT computed with “complex numbers” having  $\mu$  as imaginary unit. Although that statement is technically incorrect (complex numbers strictly belong to  $\mathbb{C}_i$ ), from a computational point of view, the  $\text{DFT}_\mu$  can be calculated

---

<sup>4</sup>Please note that, in what follows, the usual subscripts 1 and 2 for denoting the simplex and perplex parts were replaced by the superscripts  $^{(s)}$  and  $^{(p)}$ , as in  $q = q^{(s)} + q^{(p)} \boldsymbol{j}$ , to avoid confusion with the summation indices.

using the very same algorithms as the usual DFT (since the imaginary unit *per se* is disregarded in the low-level calculations).

This brief presentation of the QDFT illustrates that

- although the frequency components are quaternion signals, their frequencies are real-valued (see the argument in the transform kernel),
- the eigenstructure of the QDFT likens heavily that of the usual DFT, to the point of allowing the reuse of common DFT algorithms. This resemblance will be exploited when investigating the fractionalization of the QDFT, in Chapter 5.

For a thorough introduction on quaternions and their application to signal processing, the reader may refer to (Grigoryan and Agaian 2018; Zhang 1997; Ell et al. 2014; Flamant et al. 2017a; Jiang et al. 2014).

## 2.4 Summary

This chapter discussed the main topics concerning the quaternion algebra and its Fourier transform. Although they have not been addressed with the depth and details a more curious reader would wish, the discussion was hopefully enough to present a coherent overview of

- how the quaternions extend the complex numbers, by carrying not 2 but 4 real-valued components splitted into a scalar and a vector part,
- how the quaternion multiplication rules effect the quaternion similarity classes, with their rotation property,
- how the symplectic decomposition and the complex adjoint matrices are core concepts in the eigendecomposition of a quaternion matrix, and
- how to compute the left 1D discrete quaternion Fourier transform by using usual DFT calculations.

# A review on fractional and graph Fourier transforms

At the heart of the field of spectral analysis lies the concept of *Fourier transforms*, all of which are creatively tailored to map the original signal onto convenient new spaces. This links directly to the argument presented in the Introduction of this thesis: the power and flexibility of signal processing stem from the many ways in which one can explore different signal representations. While the last chapter ended with a brief presentation of the QFT, the next pages will guide the reader through an overview on two sets of transforms which are also relevant to understanding the contributions in the thesis: the fractional Fourier transform (FrFT) and the graph Fourier transform, the latter included in the broader discussion on fundamentals of graph signal processing.

## 3.1 The fractional Fourier transform

It is quite common in theoretical investigations to expand the definition of some well-established operator and consider the consequences of an intermediate (fractional) application of the underlying operation. Take, for example, the power operation, which makes *sense* only for integer exponents (i. e.,  $x^3 \stackrel{\Delta}{=} x \cdot x \cdot x$ ), but has great use when considering also real or complex exponents. Or the derivative of a function  $f(t)$ , which can be defined to possess a fractional-order equivalent by setting  $n$  to non-integer values in the time differentiation property of the Fourier transform  $\mathcal{F}\{\cdot\}$ :

$$\mathcal{F}\left\{\left[\frac{d^n}{(dt)^n} f(t)\right]\right\} = (j\omega)^n \mathcal{F}\{f(t)\}. \quad (3.1)$$

Mendlovic and Ozaktas made this observation (Mendlovic and Ozaktas 1993) and pointed out that this was the spark behind the creation of the FrFT. However theoretical this endeavor might have been, even at its origin in the 1980's paper on the Journal of Applied Mathematics by the physicist Victor Namias (Namias 1980), the fractional Fourier transform had already potential applications, originally in quantum mechanics. After 40 years, the FrFT has been applied to time-frequency

analysis, compression, digital watermarking, filtering, encryption, let alone its utility in Optics (Bultheel and Martinez Sulbaran 2002; Lima et al. 2018).

The fractional Fourier transform is a generalization of the usual continuous-time Fourier transform and can be interpreted as a rotation of the signal (operand) in the time-frequency plane (Almeida 1994). To understand why, let us consider its definition. The usual (unitary) Fourier transform, with operator represented by  $\mathcal{F}$  as in (3.1), has analysis and synthesis equations given by

$$\mathcal{F} \{f(t)\} = F(j\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) \exp(-j\omega t) dt, \quad (3.2)$$

$$\mathcal{F}^{-1} \{F(j\omega)\} = f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(j\omega) \exp(j\omega t) d\omega. \quad (3.3)$$

The eigenfunctions of the  $\mathcal{F}$  operator are the family of functions  $\exp\left(\frac{-x^2}{2}\right)H_n(x)$  (on the variable  $x$ ), where  $H_n(x)$  are the Hermite polynomials of order  $n$  (Namias 1980). The associated eigenvalues have the form  $\exp(-jn\frac{\pi}{2})$ ; therefore, if we add the notation  $\mathcal{F}_{\frac{\pi}{2}} \triangleq \mathcal{F}$ ,

$$\mathcal{F}_{\frac{\pi}{2}} \left\{ \exp\left(\frac{-x^2}{2}\right) H_n(x) \right\} = \exp\left(-jn\frac{\pi}{2}\right) \exp\left(\frac{-x^2}{2}\right) H_n(x). \quad (3.4)$$

The FrFT is then *defined* as the operator  $\mathcal{F}_\alpha$  satisfying the eigenvalue equation

$$\mathcal{F}_\alpha \left\{ \exp\left(\frac{-x^2}{2}\right) H_n(x) \right\} = \exp(-jn\alpha) \exp\left(\frac{-x^2}{2}\right) H_n(x), \quad (3.5)$$

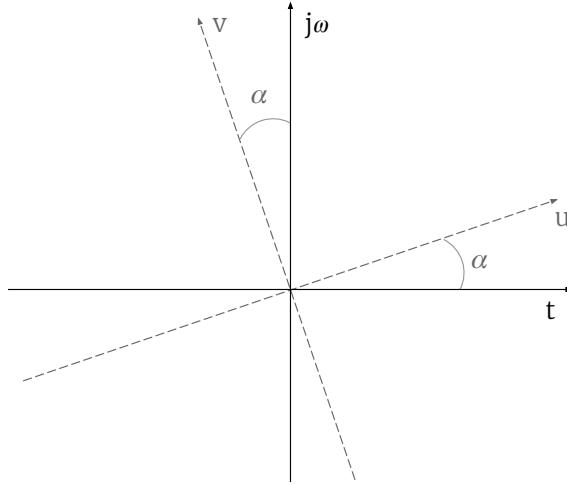
from which follows that  $\alpha = 0$  yields the identity operator and  $\alpha = \frac{\pi}{2}$  makes the FrFT coincide with the usual Fourier transform (by definition). The *order* of the fractional Fourier transform is defined by  $m = \frac{2\alpha}{\pi}$ .

The integral form of the operator  $\mathcal{F}_\alpha$  is

$$\mathcal{F}_\alpha \{f(t)\} = \int_{-\infty}^{\infty} f(t) K_\alpha(t, u) dt, \quad (3.6)$$

in which the transformation kernel  $K_\alpha(t, u)$  is given by

$$K_\alpha(t, u) = \begin{cases} \sqrt{\frac{1-j \cot \alpha}{2\pi}} \exp\left(j \frac{t^2+u^2}{2} \cot \alpha - jut \csc \alpha\right), & \text{if } \alpha \text{ is not a multiple of } \pi, \\ \delta(t-u), & \text{if } \alpha \text{ is a multiple of } 2\pi, \\ \delta(t+u), & \text{if } \alpha + \pi \text{ is a multiple of } 2\pi. \end{cases} \quad (3.7)$$



**Fig. 3.1.:** Illustration of the FrFT property of rotation in the time-frequency plane. Adapted from (Almeida 1994).

Two successive applications on the signal  $f(t)$  of the FrFT with orders  $m_1 = \frac{2\alpha_1}{\pi}$  and  $m_2 = \frac{2\alpha_2}{\pi}$  can be written in a single equation as

$$\mathcal{F}_{\alpha_2} \{ \mathcal{F}_{\alpha_1} \{ f(t) \} \} = \int_{-\infty}^{\infty} \left( \int_{-\infty}^{\infty} f(t) K_{\alpha_1}(t, s) dt \right) K_{\alpha_2}(s, u) ds \quad (3.8)$$

$$= \int_{-\infty}^{\infty} f(t) \left( \int_{-\infty}^{\infty} K_{\alpha_1}(t, s) K_{\alpha_2}(s, u) ds \right) dt. \quad (3.9)$$

After a “rather long derivation”, according to (Almeida 1994), it has been shown that

$$\int_{-\infty}^{\infty} K_{\alpha_1}(t, s) K_{\alpha_2}(s, u) ds = K_{\alpha_1 + \alpha_2}(t, u), \quad (3.10)$$

and this implies

$$\mathcal{F}_{\alpha_2} \{ \mathcal{F}_{\alpha_1} \{ f(t) \} \} = \int_{-\infty}^{\infty} f(t) K_{\alpha_1 + \alpha_2}(t, u) dt = \mathcal{F}_{\alpha_1 + \alpha_2} \{ f(t) \}. \quad (3.11)$$

In other words, the FrFT has the property of additivity of transform orders. Let us see how this fits quite nicely with the known behaviour of successive applications of the usual unitary Fourier transform.

It is known that Fourier-transforming twice the same signal yields the operand with reflected time axis,

$$\mathcal{F}^{(2)} \{ f(t) \} \triangleq \mathcal{F} \{ \mathcal{F} \{ f(t) \} \} = f(-t), \quad (3.12)$$

applying the transform once more yields the signal spectrum with reflected frequency axis,

$$\mathcal{F}\{f(-t)\} = F(-j\omega), \quad (3.13)$$

and further transforming once more gives back the original signal,

$$\mathcal{F}\{F(-j\omega)\} = f(t). \quad (3.14)$$

It is as if the Fourier transform moved the signal back and forth in 90° degrees rotations in the time-frequency plane. From this point of view, the FrFT may be seen as a general rotation in the time-frequency plane, with  $\mathcal{F}_\alpha$  rotating the signal axis by  $\alpha$  degrees. A 360° rotation yields

$$\mathcal{F}_{2\pi}\{f(t)\} = \mathcal{F}_{\frac{\pi}{2} + \frac{\pi}{2} + \frac{\pi}{2} + \frac{\pi}{2}}\{f(t)\} = \mathcal{F}^{(4)}\{f(t)\} = f(t), \quad (3.15)$$

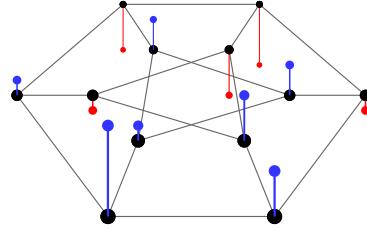
which in fact goes “back” to the original time axis. See a depiction of the rotation in the time-frequency plane in Fig. 3.1.

Finally, the additivity of orders guarantees that the inverse of the  $\mathcal{F}_\alpha$  exists and equals  $\mathcal{F}_{-\alpha}$ . Since the fractional transform order is a real-valued parameter, it is quite hard to recover perfectly the original signal from its spectrum if the value of  $\alpha$  is unknown. This favors the use of discrete fractional transforms in specific steps of encryption schemes (Tao et al. 2010; Kang et al. 2018; Kang and Tao 2018).

## 3.2 Fundamentals of graph signal processing

Multivariate data defined over networks are nowadays ubiquitous, being constantly generated, stored and processed in the most diverse systems in engineering and technology. Measurements in a set of IoT sensors and mobile devices (Alam et al. 2015; Guo et al. 2016; Ma et al. 2016; Yu et al. 2016), number of citations in a scientific collaboration network or social media relations (*collaboration graph*, or *social graph*) (Chung 2010) and interactions between individuals in a ecosystem (*ecological networks*) (Golubski et al. 2016) are some examples of situations in which the acquired data are intimately related to the topology of the network over which they are defined.

Such multivariate network-like systems are not only present in various applications, but are also systematically growing in number, as sensors become cheaper and smaller and concepts such as cloud storage/computing and Big Data consolidate, as indicated by the 2011 report from McKinsey Global Institute (McKinsey



**Fig. 3.2.:** Example of signal defined over a graph. The height of the vertical bars indicate the value of the signal samples, which are indexed by the graph vertices. The graph edges capture similarity relations between samples. How could one define spectral analysis and processing techniques in such a signal domain?

et al. 2011). This document also states that the information acquired from the adequate processing of such massive networked data is a fundamental requisite for the companies to thrive from now on.

Still another motivation that feeds the urge to study processing techniques for data defined over network-like domains, for example, is the growth of research on *smart cities*, which takes advantage of the considerable information (that are or are yet to be) generated in cities to provide (or improve the) solutions for many urban problems (Jain et al. 2014).

All these examples share an important characteristic: the structure over which the data is defined may be modelled by a graph (Mei and Moura 2016), to which vertices are assigned the variables of interest, as depicted in Fig. 3.2. That is the context in which the field of *graph signal processing* (GSP) was developed in the last decade, a theoretical framework aiming to generalize the classical signal processing methods and concepts to scenarios in which the signal is no more defined over a regular domain, but sits on a generally irregular structure, an arbitrary graph. The research is still very active and numerous contributions have been made, but two distinct frameworks consistently grew throughout the years and have been established as default mindsets when dealing with graph signals. The first one is based on algebraic signal processing and uses the graph adjacency matrix as elementary block. This approach imposes no restrictions regarding the graph being directed or undirected, and the edge weights are allowed to be negative or complex numbers (Sandryhaila and Moura 2014a). The second framework draws ideas from spectral graph theory and analyzes signals defined only over undirected graphs with non-negative real edge weights, using the graph Laplacian matrix to build a basis for the signal space (Shuman et al. 2013). This section will cover the first branch of GSP, using the adjacency matrix as default shift operator, since it matches the approach used in the remaining chapters of this thesis.

### 3.2.1 The challenge of graph-like domains

One of the reasons why GSP has been such a fertile field, allowing the birth of so many different problems and ideas, is that the definition of a signal over a graph leads to a series of obstacles when attempting to use even the most fundamental concepts of signal processing. Let us take the simple but elucidating example given by Shuman *et al.* (Shuman et al. 2013), and consider the unit shift to the right of a discrete-time signal  $x[n]$ , which is done in digital signal processing by the simple variable substitution  $x[n - 1]$ . Good enough, but what does it mean to right-shift the signal in Fig. 3.2, for example? Obviously the sense of right and left are meaningless for general graphs. On this problem, Shuman *et al.* argue that a naïve choice would be to label the  $N$  graph vertices from  $v_0$  to  $v_{N-1}$ , so that the sample  $x[n]$  is assigned to vertex  $v_n$ , for doing so would allow to define the shifted signal as the result of assigning  $x[n]$  to vertex  $v_{(n-1) \bmod N}$ . Such an option, however, is not adequate, for its repeatability depends always on the way the vertices are labeled. This example illustrates how a concept in DSP as simple as signal translation may deserve a cautious study in GSP.

## 3.3 Principles and definitions

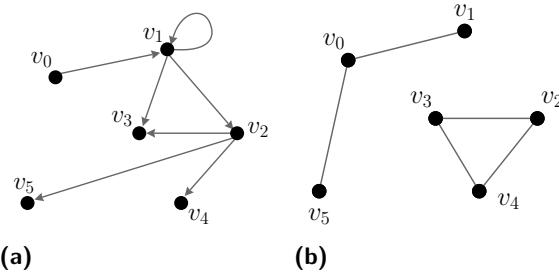
The field of graph signal processing draws basic concepts from the classical theories of digital signal processing and graph theory, aiming to provide a cohesive and useful framework to tackle the aforementioned challenges. In this section, some of the main definitions found in this field are presented.

### 3.3.1 Graph theory: a brief terminology

A graph is commonly defined as the ordered pair  $(\mathcal{V}, \mathcal{E})$ , in which the set  $\mathcal{V}$  contains the so called graph *vertices* and the set of *edges*  $\mathcal{E}$  is a subset of  $\mathcal{V}^2$  (Feofiloff et al. 2011). We will usually indicate by  $|\mathcal{V}| = N^1$  and  $|\mathcal{E}| = E$  the number of vertices and edges of a graph, respectively. For our purposes it is convenient to represent a graph as the structure  $\mathcal{G} = \{\mathcal{V}, \mathbf{A}\}$ , endowed with the (weighted) adjacency matrix  $\mathbf{A}$  which captures the vertex-to-vertex relations: if  $A_{i,j} \neq 0$ , then there is an edge of weight  $A_{i,j}$  from the vertex  $v_j$  to  $v_i$ . It is denoted by  $d_i^-$  the *indegree* of vertex  $v_i$ ,

---

<sup>1</sup>The set operator  $|\cdot|$  means the cardinality, or amount of elements, of the set.



**Fig. 3.3.:** Examples of (a) directed and (b) undirected graphs, defined over the same vertex set.

consisting of the sum of weights of all incoming edges to vertex  $v_i$ . Likewise, the *outdegree*  $d_i^+$  is the sum of weights of edges departing from  $v_i$ .

A graph is called *undirected* if and only if its adjacency matrix is symmetric, in which case it is defined the *degree* of vertex  $v_i$  as  $d_i^- = d_i^+ = d_i$ . In this case, a graph is said to be *d-regular* whenever all graph vertices have degree  $d$ . If  $\mathbf{A}$  is asymmetric, however, the respective graph is *directed* and its pictorial representation depicts the edges as arrows, to account for the unidirectional relation between adjacent vertices. Examples of directed and undirected graphs are shown in Fig. 3.3.

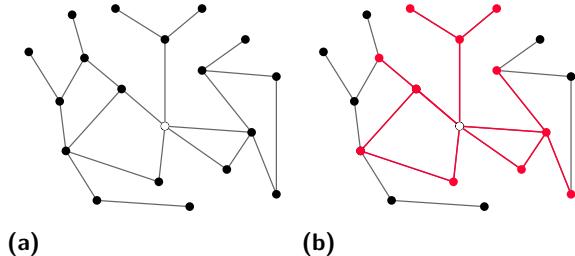
The adjacency matrix is the building block for one of the two main frameworks of GSP, what will be covered soon, but another matrix of great importance, mainly in the branch of GSP originated from spectral graph theory, is the Laplacian matrix

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \quad (3.16)$$

with the *degree matrix*  $\mathbf{D}$  being a diagonal matrix with  $d_i$  as the  $i$ -th entry of its main diagonal. Depending on the context,  $\mathbf{D}$  may be taken as the indegree or outdegree matrix, although when the Laplacian matrix is used the graphs considered are often undirected.

A *path* is a set of distinct edges (with the same orientation, if the graph is directed) linking distinct vertices, so that one could traverse all vertices by walking over the edges without ever repeat an edge or a vertex. A *cycle* is like a path, except that it has equal starting and end vertices, and if a graph has a cycle it is called *cyclic* (*acyclic*, otherwise). If the cycle has only one edge, it is called a *loop*. One refers to *multiple edges* whenever a single pair of vertices is connected by two or more edges. An undirected graph is called *simple* if it has no loops nor multiple edges.

A graph is said to be *complete* if any two of its vertices are adjacent. Graph signal processing over such graphs may be extremely cumbersome, for the compu-



**Fig. 3.4.:** (a) A graph and (b) the set of vertices  $\mathcal{N}(i, 2)$  shown in red, with  $v_i$  being depicted in white. The edges linking  $v_i$  to the elements in  $\mathcal{N}(i, 2)$  are also highlighted in red.

tational complexity of many of its techniques depends heavily on the number of graph edges. For most applications, it is desirable to have a small number of edges while keeping the graph *connected*, i. e., for any pair of vertices there exists a path connecting them.

A graph is said to be *unweighted* if all its edges have unit weight. A *subgraph* of  $\mathcal{G}$  is a graph  $\mathcal{G}' = (\mathcal{V}', \mathbf{A}')$  with edge set  $\mathcal{E}'$ , in which  $\mathcal{V}' \subset \mathcal{V}$  and  $\mathcal{E}' \subset \mathcal{E}$ . A *connected component* of  $\mathcal{G}$  is a connected subgraph  $\mathcal{G}' = (\mathcal{V}', \mathbf{A}')$  in which any vertex in  $\mathcal{V}'$  is linked exclusively to another vertex also in  $\mathcal{V}'$ . This is illustrated by Fig. 3.3b, in which the graph has two connected components.

The *neighbourhood* of a vertex  $v_i$  is the set  $\mathcal{N}_i$  of all vertices adjacent to  $v_i$ . Sometimes it is useful as well to denote by  $\mathcal{N}(i, K)$  the set of vertices connected to  $v_i$  through a path of length  $K$  or less. This notion is represented in Fig. 3.4.

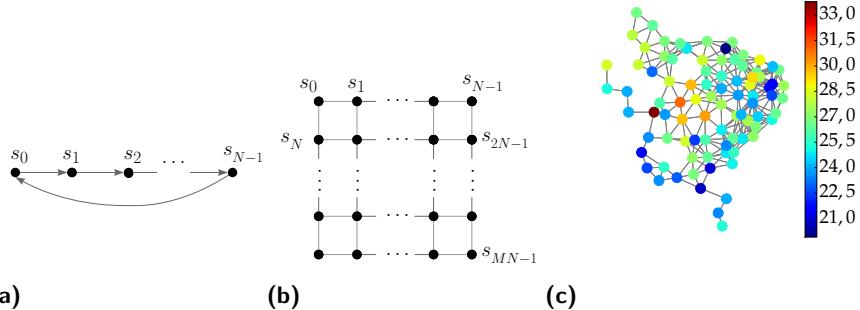
The reader is encouraged to refer to this section whenever necessary. For a broader glossary with a solid introduction to graph theory, the reader may wish to read (Murty and Bondy 2008; Chung 1997).

### 3.3.2 Defining a graph signal

A signal  $s$  defined over  $\mathcal{G} = \{\mathcal{V}, \mathbf{A}\}$ , with  $|\mathcal{V}| = N$ , is a discrete-domain function mapping the graph vertex set to a scalar set, usually the complex or real numbers,

$$s : \mathcal{V} \rightarrow \mathbb{C} \mid s(v_i) = s_i, \quad i = 0, \dots, N - 1, \quad (3.17)$$

so that the graph signal  $s$  can be seen as a column vector in  $\mathbb{C}^N$  indexed by the vertices of  $\mathcal{G}$ . Once the vertices  $\mathcal{V} = \{v_1, \dots, v_N\}$  are clearly labeled, it is not ambiguous to represent the signal as the column vector  $\mathbf{s} = (s_0 \ s_1 \ \dots \ s_{N-1})^T$ ,  $s_i \in \mathbb{C}$ ,  $0 \leq i \leq N - 1$ .



**Fig. 3.5.:** Examples of depictions of graph signals over (a) a directed ring graph, (b) an undirected regular grid graph and (c) a graph of cities from the Brazilian Northeastern region, over which was defined a signal of temperature measurements from February 1<sup>st</sup> of 2012, retrieved from the *Banco de Dados Meteorológicos para Ensino e Pesquisa* (BDMEP, freely translated as Meteorological Database for Teaching and Research), available at: <http://www.inmet.gov.br/portal/index.php?r=bdmep/bdmep>.

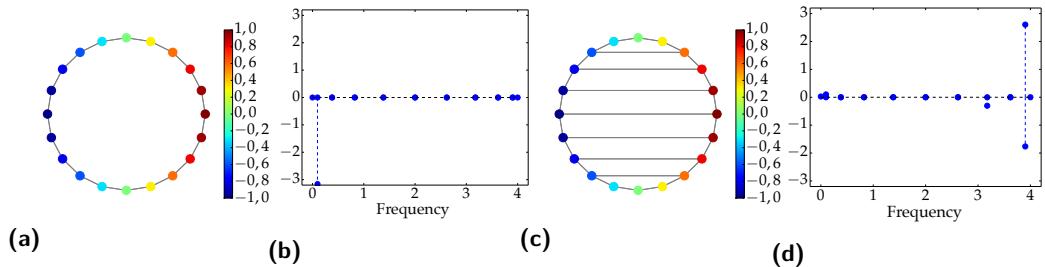
Fig. 3.5 provides examples of graph signal representations, in which the vertex labeling is omitted for the sake of simplicity, as it will be assumed that the signal sample  $s_i$  is assigned to vertex  $v_i$ . The signal values are indicated in two manners: either by writing down its numerical value next to the respective vertex, or by using a pseudocolor scale, the latter of which is the scheme adopted throughout this paper.

It is crucial to stress a certain graph which links GSP to the classical DSP theory: the directed ring graph, shown in Fig. 3.5a, which models the finite-length discrete-time domain. Its directed edges model the causality of time domain, whereas the feedback edge accounts for the boundary condition of periodicity imposed by the DFT analysis. Other signals that arise in practical applications have the respective graphs easily identified: the rectangular lattice in Fig. 3.5b, for example, models the digital image domain (Sandryhaila and Moura 2012), and Fig. 3.5c shows an example of signal defined over a mesh network of sensors, with the edges weighted using the inverse of the euclidian distance, which arises in many scenarios such as IoT applications.

The spectral characteristics of a signal depend heavily on the domain over which it is defined, but one does not need to acknowledge this in the context of DSP, for in this case the domains are always regular and uniform<sup>2</sup>. From the classical theory, the common understanding states that a signal has mostly low frequencies if adjacent samples have similar values, and high frequencies otherwise. When dealing

---

<sup>2</sup>One could argue that, in the theory of *nonuniform sampling*, the signal is defined over an irregular domain, since the samples may be randomly spaced. Even in this case, however, the classical techniques still aim to *recover* the signal so as to represent it in its usual – and *uniform* – domain.



**Fig. 3.6.:** The same signal defined over two similar graphs, one of them being (a) the undirected ring graph. In (b) and (d) are depicted the Fourier spectra of the signals in (a) and (c), respectively.

with signals defined over graphs, it is clear that the adjacency relations depend on the graph topology, and therefore one may foresee that *the same signal may present different spectra when defined over different graphs*. This intuition is visually confirmed (and will soon be mathematically proved) in Fig. 3.6, which depicts a signal and its spectra<sup>3</sup> when two different graphs are taken as domain. The reader may notice that, in Fig. 3.6c, the samples with highest values are adjacent to the ones with small values, what causes the highest concentration of energy to be in the highest frequencies. The opposed behaviour is observed in the signal defined over the undirected ring graph in Fig. 3.6a.

### 3.3.3 Graph inference

Some contexts in which GSP is applied do not provide clear information on *how the underlying graph is structured*. For example, let us suppose the temperature data (or any other data in fact) of some Brazilian Northeastern cities will be treated using GSP. How is one supposed to weight the graph edges, and before this, how does one decide which vertices to connect? Is the graph shown in Fig. 3.5c the only option? Clearly not. Although generally the problem of graph inference is complex, this type of geography-based graph has an adequate method of topology estimation.

The general idea is that, if there is a clear metric to evaluate the *expected* similarity between samples as a function of the available information regarding the respective vertices, then this metric may be used to generate the edge weight and a threshold is set so that any weight below this value causes the respective edge to

<sup>3</sup>These spectra were obtained using the Laplacian matrix as a graph shift operator, not the adjacency matrix. The reader should not bother with the differences for now, it suffices to know that the spectra can be read as usual: frequencies increase as you move to the right on the frequency axis. In other words, any low-pass signal will have most of its energy lying on the left-hand side of the spectrum.

be eliminated. In the case of vertices which have geographic location, the euclidian distance may be used as the metric because vertices that are closer together are (usually) *expected* to have similar signal samples, and therefore the adjacency matrix of the underlying graph may have entries given by

$$A_{ij} = \begin{cases} \exp\left(-\frac{\text{dist}^2(v_i, v_j)}{2\theta^2}\right) & \text{if } \text{dist}(v_i, v_j) < T \\ 0 & \text{otherwise,} \end{cases} \quad (3.18)$$

as used in (Shuman et al. 2013). The choice of the parameters  $T^4$  and  $\theta$  (standard deviation of the distribution), and of how to use the metric (in this case, inside a Gaussian distribution), are dictated by the application and by the analyst experience.

However, if there is an isolated vertex, far from the others, the use of (3.18) may lead to a compromise between keeping the graph connected and obtaining a sparse adjacency matrix, since imposing connectivity to the graph in this case implies increasing  $T$ , and therefore having many edges. To deal with this problem and still have a good representation of the underlying graph, one alternative is to connect a vertex to its  $K$  closest neighbours (setting  $K$  to an appropriate value, according to the context) and weight the edges using the Gaussian distribution in (3.18).

As previously discussed, these methods require an adequate metric to evaluate the expected similarity between samples in the graph vertex, but given the diverse areas in which graph signals may arise, estimating the topology of the underlying graph constitutes a challenge of its own (Mei and Moura 2016; Sardellitti et al. 2016).

### 3.4 Formulating GSP based on the graph adjacency matrix

In 2008, Püschel and Moura published their *algebraic signal processing* (ASP) theory (Püschel and Moura 2008b; Püschel and Moura 2008a), which expands DSP by looking at it from an algebraic point of view: each signal processing theory is studied as a triple  $(\mathcal{A}, \mathcal{M}, \Phi)$  consisting of an algebra  $\mathcal{A}$  (a vector space endowed with multiplication between vectors), an  $\mathcal{A}$ -module  $\mathcal{M}$  (a vector space over the

---

<sup>4</sup> $T$  indicates a distance threshold above which we set the edge weight to zero, effectively leaving the vertices unlinked. This means that the distance between them is assumed to be too high for any significant interdependence to exist.

same base field as  $\mathcal{A}$  which admits left-multiplication by elements of  $\mathcal{A}$ ) and a linear transformation  $\Phi$ .  $\mathcal{A}$  is called the filter space,  $\mathcal{M}$  is the signal space and  $\Phi$  is the Fourier transform (homomorphism over  $\mathcal{M}$ ) associated with the structure.

When these authors drew inspiration from ASP to develop their GSP theory, the starting point was necessarily to find (better, to define) the unit shift operator of graph signals, the reason being that such an operator in ASP is the building block of the algebra  $\mathcal{A}$  (as, for example, the unit delay  $z^{-1}$  is the building block for filters of discrete-time and finite-length signals  $\mathcal{A} = \{\sum_{\ell=0}^{N-1} h_\ell z^{-\ell} | h_\ell \in \mathbb{C}\}$ ). To do so, the shift of discrete-time signals, defined over directed ring graphs, was investigated.

By inspection of the *adjacency matrix* of the directed ring graph (Fig. 3.5a),

$$\mathbf{C} = \begin{bmatrix} & & & 1 \\ 1 & & & \\ & \ddots & & \\ & & & 1 \end{bmatrix}, \quad (3.19)$$

it was noticed that the unit (circular) shift of discrete-time signals is precisely the left-multiplication by  $\mathbf{C}$ , for a given discrete-time signal  $\mathbf{x} = (x_0 \ x_1 \ \dots \ x_{N-1})^T$ ,

$$\mathbf{Cx} = \begin{bmatrix} & & & 1 \\ 1 & & & \\ & \ddots & & \\ & & & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} = \begin{bmatrix} x_{N-1} \\ x_0 \\ \vdots \\ x_{N-2} \end{bmatrix} \triangleq \mathbf{x}^{(1)}, \quad (3.20)$$

and the generalization followed: *the graph unit shift was defined as the left-multiplication by the graph adjacency matrix*.

In other words, for a signal  $\mathbf{x}$  defined over the graph  $\mathcal{G} = \{\mathcal{V}, \mathbf{A}\}$ , the adjacency matrix  $\mathbf{A}$  acts as a *filter* which “delays” (i. e., translates in the vertex domain) the graph signal  $\mathbf{x}$  (represented here as a column vector) by one unit, producing the delayed version represented hereinafter by  $\mathbf{x}^{(1)} = \mathbf{Ax}$ .

### 3.4.1 Graph filters

Seeing the adjacency matrix as a filter suggested the general definition of *graph filter* as any matrix  $\mathbf{H} \in \mathbb{C}^{N \times N}$  (Sandryhaila and Moura 2013b), which preserves the necessary property that the output of a filter (i. e., the matrix-vector product) is a signal (i. e., a column vector). Such a definition implies that *linearity*

is always valid for graph filters, since the distributivity of matrix multiplication with respect to matrix addition guarantees that

$$\mathbf{H}(\alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2) = \alpha_1 \mathbf{Hx}_1 + \alpha_2 \mathbf{Hx}_2. \quad (3.21)$$

The next desirable property would be *shift invariance*, analogous to the classical time invariance of DSP, and this means that filtering and shifting should commute. In other words, for a graph filter  $\mathbf{H}$  to be *linear and shift invariant* (LSI) it is required that

$$\mathbf{AHx} = \mathbf{HAx}, \quad \forall \mathbf{x} \in \mathbb{C}^N \Rightarrow \mathbf{AH} = \mathbf{HA}. \quad (3.22)$$

The following theorem establishes an important property satisfied by every LSI filter (Sandryhaila and Moura 2013a).

**Theorem 5.** *Let  $\mathbf{A}$  be the adjacency matrix of a graph. Let us assume that the characteristic polynomial  $\text{char}_{\mathbf{A}}(x)$  of  $\mathbf{A}$  coincides with the respective minimal polynomial  $m_{\mathbf{A}}(x)$ . Therefore,  $\mathbf{H}$  is an LSI filter if and only if  $\mathbf{H}$  is a polynomial in  $\mathbf{A}$ , i. e.*

$$\mathbf{H} = h(\mathbf{A}) = \sum_{\ell=0}^L h_{\ell} \mathbf{A}^{\ell}, \quad (3.23)$$

where  $\mathbf{A}^0$  is the identity matrix and  $L < \deg(m_{\mathbf{A}})$ .

The assumption on  $\text{char}_{\mathbf{A}}(x)$  and  $m_{\mathbf{A}}(x)$  in Theorem 5 does not hold for all adjacency matrices  $\mathbf{A}$ . Nevertheless, the result in the referred theorem can be extended to all matrices using the concept of *equivalent graph filters*, as clearly explained in (Sandryhaila and Moura 2013b). As a consequence, for any graph  $\mathcal{G} = \{\mathbf{A}, \mathcal{V}\}$ , every LSI filter has polynomial representation in  $\mathbf{A}$ . In this sense, Theorem 5 suggests a convenient analogy with the classical DSP, since every filter for discrete-time signals can be represented as polynomials evaluated in  $z^{-1}$ , the unit delay, via the  $z$ -transform of its impulse response.

### 3.4.2 Graph Fourier transform

The topic of spectral analysis is key in signal processing. When studying how this would be implemented in the scope of GSP, the authors started by looking at the classical Fourier transform as a signal decomposition into a basis of eigenfunctions of the LTI filtering (Oppenheim et al. 1997), i. e., the basis of complex time exponentials. From that point, the generalization to GSP was clear: the *graph Fourier transform*

(GFT) was defined as the decomposition of a graph signal into a basis of eigenvectors of LSI filtering<sup>5</sup>.

Let us take the graph  $\mathcal{G} = \{\mathcal{V}, \mathbf{A}\}$ ,  $|\mathcal{V}| = N$ . If  $\mathbf{A}$  is diagonalizable<sup>6</sup>, then one may write

$$\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^{-1}, \quad (3.24)$$

in which  $\mathbf{V}$  contains the  $N$  eigenvectors of  $\mathbf{A}$  in its columns,

$$\mathbf{V} = (\mathbf{v}_0 \ \mathbf{v}_1 \ \dots \ \mathbf{v}_{N-1}). \quad (3.25)$$

Since LSI filters are polynomials in  $\mathbf{A}$ , and since a matrix and its powers share the same set of eigenvectors, the columns of  $\mathbf{V}$  form a basis of vectors invariant to LSI filtering. Besides, given that the subspaces generated by the linearly independent eigenvectors of a same eigenvalue of  $\mathbf{A}$  are irreducible, have null intersection and the dimensions of all subspaces add to  $N$  (Sandryhaila and Moura 2013c),  $\mathbf{V}$  provides a basis which is invariant to LSI filtering for the space of signals defined over  $\mathcal{G}$ .

Therefore, a signal  $\mathbf{x}$  may be decomposed into its components with respect to  $\mathbf{V}$  as

$$\begin{aligned} \mathbf{x} &= \hat{x}_0 \mathbf{v}_0 + \dots + \hat{x}_{N-1} \mathbf{v}_{N-1} \\ &= \mathbf{V}(\hat{x}_0 \ \hat{x}_1 \ \dots \ \hat{x}_{N-1})^T \\ &= \mathbf{V}\hat{\mathbf{x}} \stackrel{\Delta}{=} \text{GFT}\{\mathbf{x}\}, \end{aligned} \quad (3.26)$$

and this is the synthesis equation of the GFT. The analysis equation follows,

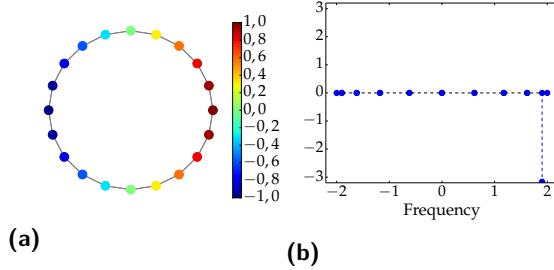
$$\hat{\mathbf{x}} = \mathbf{V}^{-1}\mathbf{x} \stackrel{\Delta}{=} \text{GFT}^{-1}\{\mathbf{x}\}. \quad (3.27)$$

It has been emphasized that the directed ring graph is the link between GSP and DSP, because it models the discrete-time domain. This provides a way of checking how consistent with the classical theory are the proposed GSP tools. When investigating how the GFT would act upon discrete-time signals, one should first diagonalize the adjacency matrix  $\mathbf{C}$  of the directed ring graph, given by (3.19). Since it is circulant, it is known to be diagonalized by the DFT matrix  $\mathbf{F}$ , with

---

<sup>5</sup>One may wish to distinguish between the different implementations of the GFT regarding the choice of graph shift operator. Unless clearly stated otherwise, this chapter considers it to be always the graph adjacency matrix, but other popular option is the graph Laplacian, as already mentioned. In this regard, the reader even may find the terms  $\text{GSP}_A$  and  $\text{GSP}_L$  in previous publications (Ribeiro and Lima 2018), making the distinction clear.

<sup>6</sup>If not, the reasoning may be replicated using the Jordan decomposition of  $\mathbf{A}$  and the set of generalized eigenvectors (Deri and Moura 2017).



**Fig. 3.7.:** (a) Signal defined over an undirected ring graph and (b) its spectrum. The frequencies are indicated by the eigenvalues of the adjacency matrix.

entries  $F_{n,k} = \exp\left(-j\frac{2\pi}{N}nk\right)$ , which contains in its *rows* the DFT eigenvectors. The calculation of the characteristic polynomial of  $\mathbf{C}$ ,

$$p_{\mathbf{C}}(\lambda) = \det(\lambda\mathbf{I} - \mathbf{C}) = \begin{vmatrix} \lambda & & & -1 \\ -1 & \lambda & & \\ \ddots & \ddots & \ddots & \\ & -1 & \lambda \end{vmatrix} = \lambda^N - 1, \quad (3.28)$$

shows that its eigenvalues are the  $N$  complex roots of unity. Setting these eigenvalues as the entries of a diagonal matrix  $\Lambda_{\mathbf{C}}$ , the eigendecomposition of  $\mathbf{C}$  may be written as

$$\mathbf{C} = \mathbf{F}^{-1}\Lambda_{\mathbf{C}}\mathbf{F}, \quad (3.29)$$

and one can see that, in the case of directed ring graphs, the GFT and the DFT matrices *coincide*, since  $\mathbf{V}^{-1} = \mathbf{F}$ . This equivalence indicates a desirable consistency with the classical theory.

### 3.4.3 The frequency domain

The way the GFT was defined naturally suggests the interpretation of the adjacency matrix eigenvectors  $\mathbf{v}_i$  as “frequency components” associated with the “graph frequencies” given by the eigenvalues  $\lambda_i$ , exactly as the Fourier component  $e^{-j\Omega t}$ , in the continuous time domain  $t$ , is associated with the frequency  $\Omega$ . This subsection aims to provide the mathematical justification used by Sandryhaila and Moura (Sandryhaila and Moura 2014b) to support this understanding, along with some authorial comments.

The reader may have noticed a curious consequence from what was previously stated: unless the minimal and characteristic polynomials of  $\mathbf{A}$  are equal, the same frequency may be associated with two or more linearly independent frequency

components, as indeed was the case in the example of Fig. 3.7. Furthermore, this figure shows that although the signal seems to be smooth, its frequency components are mostly associated with eigenvalues of high magnitude, what is counter-intuitive and provides a motivation to define a clear criterion to distinguish high and low graph frequencies.

The following mathematical reasoning consists of taking a metric which quantifies the expected signal smoothness, and use it to propose or confirm a notion of graph frequency. The metric used by Sandryhaila and Moura was the *total variation*, taken from classical real analysis and defined for differentiable functions as (Rudin 1987; Mallat 1999)

$$\|f\|_V = \int_{-\infty}^{\infty} |f'(t)| dt. \quad (3.30)$$

For discrete domain functions  $f_N[n]$ , the Riemann integral is replaced by first order differences,

$$\|f_N\|_V = \sum_p |f_N[n_p + 1] - f_N[n_p]|, \quad (3.31)$$

which clearly quantifies the dissimilarity between contiguous values of the function  $f_N$ . With this in mind, it was natural for Sandryhaila and Moura to use this metric in their attempt to quantify smoothness in GSP. Taking again the ring graph as a starting point, they looked at the total variation of a *finite-length* discrete-time signal  $\mathbf{x}$ :

$$TV(\mathbf{x}) = \sum_n |x_n - x_{(n-1) \bmod N}|. \quad (3.32)$$

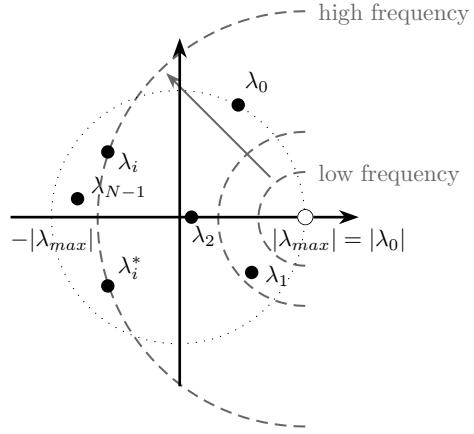
From (3.20), one can see that (3.32) may be written in terms of the  $\ell_1$ -norm<sup>7</sup> as  $TV(\mathbf{x}) = \|\mathbf{x} - \mathbf{Cx}\|_1$ , by using the directed ring graph adjacency matrix to perform the cyclic shift. From that point, the generalization consisted of using this expression and defining the *total variation on graphs* of a signal  $\mathbf{s}$  defined over the graph  $\mathcal{G} = \{\mathcal{V}, \mathbf{A}\}$  as

$$TV_G(\mathbf{s}) \stackrel{\Delta}{=} \|\mathbf{s} - \mathbf{A}^{\text{norm}}\mathbf{s}\|_1, \quad (3.33)$$

with  $\mathbf{A}^{\text{norm}} = |\lambda_{\max}|^{-1} \mathbf{A}$  and  $\lambda_{\max}$  being the eigenvalue of  $\mathbf{A}$  having the highest absolute value. The normalization of the adjacency matrix aims to avoid the excessive magnification of the shifted signal (Sandryhaila and Moura 2014b).

---

<sup>7</sup>Throughout this text, the concepts of  $\ell_1$ - and  $\ell_2$ -norm will be frequently used. They are particular cases of the  $\ell_n$ -norm of a vector  $\mathbf{x} \in \mathbb{C}^N$ , defined as  $\|\mathbf{x}\|_n \stackrel{\Delta}{=} \left( \sum_{k=0}^{N-1} |x_k|^n \right)^{1/n}$ .



**Fig. 3.8.:** Frequency ordering of graph signals, from low to high frequencies, in the complex plane (Sandryhaila and Moura 2014b).

Let  $\mathbf{A}$  be diagonalizable as in (3.24) with (possibly complex) eigenvalues ordered like so

$$|\lambda_0| \leq |\lambda_1| \leq \cdots \leq |\lambda_{N-1}| \stackrel{\Delta}{=} |\lambda_{\max}|, \quad (3.34)$$

associated with the eigenvectors  $(\mathbf{v}_i)_{i=0,\dots,N-1}$ , scaled so that  $\|\mathbf{v}_i\|_1 = 1 \forall i$ . Taking the total variation (on graphs) of the eigenvector  $\mathbf{v}_k$ , one has

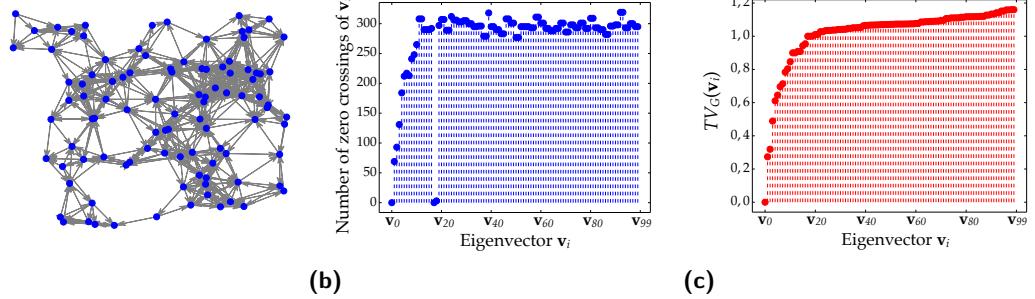
$$\begin{aligned} TV_G(\mathbf{v}_k) &= \|\mathbf{v}_k - \mathbf{A}\mathbf{v}_k\|_1 = \|\mathbf{v}_k - \frac{1}{|\lambda_{\max}|} \lambda_k \mathbf{v}_k\|_1 \\ &= \left| 1 - \frac{\lambda_k}{|\lambda_{\max}|} \right| \|\mathbf{v}_k\|_1 = |\lambda_k - |\lambda_{\max}|| \frac{\|\mathbf{v}_k\|_1}{|\lambda_{\max}|} \end{aligned}$$

so that, since  $\|\mathbf{v}_k\|_1 = 1$ ,

$$|\lambda_i - |\lambda_{\max}|| \leq |\lambda_j - |\lambda_{\max}|| \iff TV_G(\mathbf{v}_i) \leq TV_G(\mathbf{v}_j), \quad (3.35)$$

i. e., frequency components associated with eigenvalues closer to the real point  $|\lambda_{\max}|$  in the complex plane are *smoother* (because they have lower total variation), and therefore are said to be of *low frequency*. Fig. 3.8 illustrates this ordering for graph frequencies, what clarifies the spectrum of the signal in Fig. 3.7a (notice that since the graph is undirected, its adjacency matrix is symmetric and the eigenvalues are real-valued).

Let us take the directed graph in Fig. 3.9a to try to verify the consistency of the notion of frequency just derived. For this, along with the total variation on graphs, also the number of zero crossings (i. e., the number of edges connecting vertices with signal samples of different sign) will be used to *quantify* frequency. This quantity is also related to frequency in classical theory: the more a discrete signal



**Fig. 3.9.:** (a) Directed sensor graph, with  $N = 100$  vertices and no loops or multiple edges. (b) Number of zero crossings and (b) total variation of the eigenvectors  $(\mathbf{v}_i)_{i=0,\dots,N-1}$  of the adjacency matrix  $\mathbf{A}$  of the graph in (a), ordered so that the respective eigenvalues appear from the closest to the farthest from the real point  $|\lambda_{max}|$  in the complex plane. That is, according to (3.35) and Fig. 3.8, the eigenvectors are disposed in ascending order of frequency.

has contiguous samples with different sign, generally the higher are its frequency components. These two functions, the total variation on graphs and the number of zero crossings, were calculated for each of the adjacency matrix eigenvectors, and the result is shown in Fig. 3.9, in which the eigenvectors  $\mathbf{v}_k$  are ordered in such a way that the respective eigenvalues  $\lambda_k$  appear from the closest to the farthest from the real point  $|\lambda_{max}|$  in the complex plane. Both metrics have similar behaviour, but since the number of zero crossings is indifferent to graph signal variations which do not change sign, it was already expected to be less accurate as a figure of merit for frequency. It matters to highlight, however, how the adopted eigenvector ordering indeed implies an ascending frequency order, since both functions in Fig. 3.9b and 3.9c agree on the tendency of growth. More than that,  $TV_G(\mathbf{v}_k)$  grows *monotonically*, as it should do according to (3.35).

It is convenient to conclude this discussion on the graph frequency domain by referring to the frequency response of graph filters. The definition given in Subsection 3.4.1 considers the action of a matrix on a signal  $\mathbf{x}$  in the vertex domain of the graph  $\mathcal{G} = \{\mathcal{V}, \mathbf{A}\}$ . In order to understand how the filter acts in the GFT domain, hereinafter called frequency domain, one may use (3.24) and the polynomial representation of LSI filters. Let us take the filter  $\mathbf{H} = \sum_{\ell=0}^L h_\ell \mathbf{A}^\ell$  and its output to the input  $\mathbf{x}$  given by

$$\begin{aligned} \mathbf{Hx} &= \sum_{\ell=0}^L h_\ell \mathbf{A}^\ell \mathbf{x} = \sum_{\ell=0}^L h_\ell (\mathbf{V} \Lambda \mathbf{V}^{-1})^\ell \mathbf{x} \\ &= \mathbf{V} \left( \sum_{\ell=0}^L h_\ell \Lambda^\ell \right) \mathbf{V}^{-1} \mathbf{x}. \end{aligned} \quad (3.36)$$

Taking the GFT of both sides of the last equation yields

$$\mathbf{V}^{-1}\mathbf{H}\mathbf{x} = h(\Lambda)\hat{\mathbf{x}}, \quad (3.37)$$

which indicates that left-multiplication by  $\mathbf{H}$  (action of the filter in the vertex domain) is equivalent to the left-multiplication, in the frequency domain, by the matrix  $h(\Lambda)$ . In other words,  $h(\Lambda)$  represents the frequency response of  $\mathbf{H}$ .

## 3.5 Summary

This chapter guided the reader through the aspects regarding the fractional Fourier transform and graph signal processing most relevant to this work, closing the review of literature needed to have enough context to grasp and critique the chapters of original contributions that lie ahead. The key points to keep in mind are that

- the fractional Fourier transform performs a real-valued rotation in the time-frequency plane,
- the graph signal processing framework extends the usual discrete time domain to any network modeled by a graph, and as a consequence key basic concepts such as a unit shift must be redefined in terms of a graph matrix, called the graph shift operator,
- the graph Fourier transform is the projection of a graph signal (vector) onto the space of eigenvectors of the graph shift operator.



# The fractional graph shift operator and its applications

Over the last decade, theory and applications related to graph signal processing have been widely developed and attracted the attention of several scholars (Ortega et al. 2018; Djuric and Richard 2018; Ribeiro and Lima 2018). The reader may recall from Chapter 3 that GSP aims to extend concepts and operations of classical digital signal processing to scenarios in which the signals lie over irregular domains. Such scenarios include, for instance, sensors arbitrarily positioned in a geographic region and measuring some climatological variable, points of a three-dimensional cloud representing some virtual object and its attributes, people linked according to their interests and proximity relationships in a social network and so on (Chen et al. 2014b; Zhang et al. 2014; Benzi et al. 2016b; Huang et al. 2018; Ben Saad and Beferull-Lozano 2018; Jiang et al. 2021; Gama et al. 2019; Liu et al. 2019; Zhang et al. 2020; Ferreira and Lima 2020; Zhang et al. 2021; Xiao et al. 2021; Sun et al. 2021).

Among the research fronts active in GSP, the one that investigates alternatives to the shift operators, employed as building blocks to describe graph signals and systems, deserves to be highlighted (Girault et al. 2015; Gavili and Zhang 2017; Fan et al. 2019b; Fan et al. 2019a; Mollaebrahim and Beferull-Lozano 2021; Shafipour et al. 2018; Shafipour et al. 2019). In fact, when the purpose is to consider linear operators in this context, any matrix can be chosen to play the role of elementary building block; multiplying a matrix by a graph signal represented as a vector produces another signal whose samples result from a linear combination of the samples of the original signal. In this scope, the use of matrices other than the *standard* adjacency matrix and the Laplacian for the mentioned purpose may be more suitable in specific scenarios and to carry out specific (graph) signal processing tasks. Even when the focus is on designing other graph operators (e.g., the graph Fourier transform), the decision about which elementary operator to use has an impact on the expected results.

Regarding the issue discussed in the last paragraph, some works archived in the GSP literature can be brought to the fore. In (Girault et al. 2015), for example, the authors propose an isometric graph translation operator that is described in the

spectral domain as a phase shifting operator; this operator shares key properties with the time shift and behaves reasonably in the vertex domain. In (Gavili and Zhang 2017), the authors define an energy-preserving shift operator that satisfy many properties similar to their counterparts in classical signal processing; the GSP framework based on the referred operator enables the signal analysis along a correlation structure defined by a graph shift manifold. In (Fan et al. 2019b) and (Fan et al. 2019a), the authors employ different features associated with a graph to generate a series of shift operators and design a graph-filter-based classifier. Although the proposed method produces better results than those achieved using conventional graph-filter-based classifiers, it requires dealing with a non-convex optimization problem whose solution involves a relatively high computational cost. In (Mollaebrahim and Beferull-Lozano 2021), motivated by the typical scenario of asymmetric communications in wireless sensor networks, the authors study the optimal design of graph shift operators to perform decentralized subspace projection for asymmetric topologies. Obtaining the referred operators can be performed either by solving an optimization problem or by employing a decentralized algorithm based on an Alternating Direction Method of Multipliers (ADMM). In (Shafipour et al. 2018) and (Shafipour et al. 2019), the goal is to construct a graph Fourier transform for directed graphs (digraphs), such that the corresponding orthonormal frequency components are as spread as possible in the graph spectral domain. The method uses the Laplacian of an undirected version of the digraph and involves non-convex, orthonormality-constrained optimization problems.

This chapter brings contributions related to the above mentioned works by discussing the possibility (and potentially useful consequences) of computing a non-integer power  $\mathbf{A}^a$ ,  $a \in \mathbb{R}$ , of the adjacency matrix  $\mathbf{A}$ , which is taken as the (unit) graph shift operator (Sandryhaila and Moura 2014a). As a consequence, the notion of fractional shift (or delay) of signals on graphs is presented, which, to the best of the author's knowledge, has not yet been addressed in the literature. Differently from the referred papers, in which new operators are created or standard operators are adjusted using strategies potentially expensive from the computational point of view, a relatively simple generalization is proposed, which fills a theoretical gap concerning the extension to the GSP framework of a well-established concept in the classical signal processing.

This chapter contains mostly the content of the paper (Ribeiro et al. 2022), published on the IEEE Access journal, and presents the contributions listed below.

- The introduction of the fractional graph shift operator  $\mathbf{A}^a$  and the discussion of its several aspects. More specifically, the demonstration that  $\mathbf{A}^a$  can be

computed by using the theory of matrix functions, considering the Jordan decomposition of  $\mathbf{A}$ .

- The demonstration that  $\mathbf{A}^a$  acts as a graph filter, furthermore giving its frequency response and discussing issues related to fractionally shifting graph signals containing discontinuities (Gibbs phenomenon).
- An analogy between the proposed graph fractional operator and that considered in the classical discrete-time case; suggesting that, when a directed ring graph with  $N$  vertices is considered, the response of the corresponding graph filter related to  $\mathbf{A}^a$  converges to that of the classical fractional delay filter as  $N$  grows.
- The determination of the polynomial representation of  $\mathbf{A}^a$  and, with that, the demonstration that, for any graph, such an operator can be implemented as a linear and shift-invariant (LSI) graph filter.

This chapter is organized as follows. In Section 4.1, it is introduced the concept of fractional shift on graphs and the main contributions are developed: the computation of  $\mathbf{A}^a$  in Subsection 4.1.1, discussion of its interpretation in Subsection 4.1.2, demonstration of its consistency with the ideal fractional delay filter in Subsection 4.1.3 and determination of its polynomial representation in Subsection 4.1.4. Section 4.2 is devoted to numerical results related to the developed theory: it is firstly presented a small example regarding the polynomial representation of  $\mathbf{A}^a$  in Subsection 4.2.1; then a real-world graph signal is considered (temperature measured by weather stations) and it is demonstrated that, using  $\mathbf{A}^a$ , one can obtain filters that approximate an ideal filter (in the least-squares sense) better than those designed using  $\mathbf{A}$  (Subsections 4.2.2 and 4.2.3); finally, this possibility is illustrated by means of an example involving the noise removal from the same graph signal (Subsection 4.2.4). The chapter closes with concluding remarks in Section 4.3.

## 4.1 Fractional Shift on Graphs

Since the unit shift of a graph signal can be defined as the product by the adjacency matrix of the graph on which it lies, in this work, the proposed definition of a graph fractional shift as the product by a non-integer power of  $\mathbf{A}$ . Precisely, the signal  $\mathbf{x}$  over the graph  $\mathcal{G} = \{\mathbf{A}, \mathcal{V}\}$ , after being shifted by  $a \in [0, 1]$ , is given by

$$\tilde{\mathbf{x}}_a = \mathbf{A}^a \mathbf{x}. \quad (4.1)$$

The following sections discuss aspects related to the computation of  $\mathbf{A}^a$ , the interpretation of its application to a graph signal and the consistency of the proposed operator with the classical DSP approach (ideal fractional delay filter).

### 4.1.1 Computation of $\mathbf{A}^a$

The computation of  $\mathbf{A}^a$  can be well established by employing results from the theory of matrix functions (Higham 2008). In this context, it suffices to evaluate the originally scalar function  $f(t) = t^a$ ,  $a \in \mathbb{R}$  having  $\mathbf{A}$  as argument. The most direct way to formally define a function like this uses the Jordan canonical form. With this purpose, let us reconsider (3.24), replacing the diagonalization of  $\mathbf{A}$  with the most general decomposition

$$\mathbf{A} = \mathbf{V}\mathbf{J}\mathbf{V}^{-1}, \quad (4.2)$$

using the Jordan block diagonal matrix  $\mathbf{J}$ ,

$$\mathbf{J} = \text{diag}(\mathbf{J}_1, \mathbf{J}_2, \dots, \mathbf{J}_p), \quad (4.3)$$

and the matrix  $\mathbf{V}$  of generalized eigenvectors. Each  $k$ -th Jordan block  $\mathbf{J}_k$  is

$$\mathbf{J}_k = \mathbf{J}_k(\lambda_k) = \begin{bmatrix} \lambda_k & 1 & & \\ & \ddots & & \\ & & \ddots & 1 \\ & & & \lambda_k \end{bmatrix} \in \mathbb{C}^{m_k \times m_k} \quad (4.4)$$

and  $m_1 + m_2 + \dots + m_p = N$ . Denote by  $\lambda_1, \dots, \lambda_s$  the distinct eigenvalues of  $\mathbf{A}$  and by  $n_i$  the *index* of  $\lambda_i$  (the order of the largest Jordan block in which  $\lambda_i$  appears). The function  $f$  is said to be defined on the spectrum of  $\mathbf{A}$  if the values

$$f^{(j)}(\lambda_i), \quad j = 0, 1, \dots, n_i - 1, \quad i = 1, 2, \dots, s, \quad (4.5)$$

exist, where  $f^{(j)}$  denotes the  $j^{\text{th}}$  derivative of  $f$ <sup>1</sup>. That holds true for  $f(t) = t^a$ . The computation of  $f(\mathbf{A}) = \mathbf{A}^a$  can then be carried out as follows.

---

<sup>1</sup>As usual, the notation  $f'$  will be used interchangeably with  $f^{(1)}$ .

**Definition 2.** Let  $f$  be defined on the spectrum of  $\mathbf{A} \in \mathbb{C}^{N \times N}$  and let  $\mathbf{A}$  have the Jordan decomposition (4.2). Then

$$f(\mathbf{A}) \triangleq \mathbf{V} f(\mathbf{J}) \mathbf{V}^{-1} = \mathbf{V} \operatorname{diag}(f(\mathbf{J}_k)) \mathbf{V}^{-1}, \quad (4.6)$$

where

$$f(\mathbf{J}_k) \triangleq \begin{bmatrix} f(\lambda_k) & f'(\lambda_k) & \cdots & \frac{f^{(m_k-1)}(\lambda_k)}{(m_k-1)!} \\ & f(\lambda_k) & \ddots & \vdots \\ & & \ddots & f'(\lambda_k) \\ & & & f(\lambda_k) \end{bmatrix}. \quad (4.7)$$

In the present context, the last definition constitutes a practical way to calculate  $\mathbf{A}^a$ , because the Jordan form of the adjacency matrix, being necessary for the definition of the corresponding GFT, may already have been computed and thus be available to be used in (4.6) and (4.7).

#### 4.1.2 Interpreting the graph fractional shift

In order to perform a meaningful interpretation of the graph fractional shift, we consider (4.1) and the case in which  $\mathbf{A}$  is diagonalizable. Using the matrix factorization (3.24), the GFT analysis equation (3.27) and the computation strategy described in the last subsection, we can write

$$\begin{aligned} \mathbf{A}^a \mathbf{x} &= \mathbf{V} \Lambda^a \mathbf{V}^{-1} \mathbf{x} = \mathbf{V} \begin{bmatrix} \lambda_1^a & & \\ & \ddots & \\ & & \lambda_N^a \end{bmatrix} \hat{\mathbf{x}} \\ &= \mathbf{V} (\hat{\mathbf{h}}_a \odot \hat{\mathbf{x}}) = \text{GFT}^{-1}\{\hat{\mathbf{h}}_a \odot \hat{\mathbf{x}}\}, \end{aligned} \quad (4.8)$$

where  $\hat{\mathbf{h}}_a \triangleq (\lambda_1^a \dots \lambda_N^a)^T$  and  $\odot$  represents the point-wise vector product.

Equation (4.8) shows that  $\mathbf{A}^a$  is a graph filter with frequency response  $\operatorname{diag}(\hat{\mathbf{h}}_a)$ ; moreover, if  $\mathbf{x}$  is an  $N$ -point discrete-time signal (case in which the GFT coincides with the DFT), one observes that the filter in the DFT domain is the vector  $\hat{\mathbf{h}}_a$  itself. In this case, it has been discussed that the adjacency matrix of the respective graph is diagonalized according with (3.29), where  $\Lambda_C$  has as entries the  $N$  roots of unity. The fact that the matrix of eigenvectors of  $C$  is the Fourier matrix imposes a specific order of the eigenvalues in  $\Lambda_C$ , so that the vector  $\hat{\mathbf{h}}_a$  is

$$\hat{\mathbf{h}}_a = (1 \ W_N^a \ W_N^{2a} \ \dots \ W_N^{aR} \ W_N^{-aR'} \ W_N^{a(-R'+1)} \ \dots \ W_N^{-a}),$$

$$h_a[n] = \begin{cases} \frac{1}{N} \frac{\sin(\pi(n-a))}{\sin(\frac{\pi}{N}(n-a))}, & \text{if } N \text{ is odd,} \\ \frac{1}{N} \cot\left(\frac{\pi}{N}(n-a)\right) \sin(\pi(n-a)) + \frac{j}{N} (-1)^n \sin(\pi a), & \text{if } N \text{ is even.} \end{cases} \quad (4.10)$$

where  $W_N = e^{-j\frac{2\pi}{N}}$  and

$$\begin{cases} R = \frac{N-1}{2} \text{ and } R' = R, & \text{if } N \text{ is odd,} \\ R = \frac{N}{2} - 1 \text{ and } R' = R + 1, & \text{if } N \text{ is even,} \end{cases} \quad (4.9)$$

$n = 0, 1, \dots, N-1$ . It can be shown that the inverse DFT of  $\hat{h}_a$  has components given by (4.10).

The product by a fractional power of the adjacency matrix produces the effect illustrated in Fig. 4.1, for a directed ring graph; it can be seen, for example, how the 5<sup>th</sup> sample of the signal shifted by  $a = 0.3$  coincides with the value of the continuous-time signal at the same position. On the other hand, the analysis we can perform by observing the *irregular* graph in Fig. 4.2 is mostly visual; as we vary the fractional parameter from 0 (original signal) to 1, we see in the intermediate snapshots how the signal gradually spreads out from the vertices where, originally, there were already non-zero samples. In this scope, although we employ terms such as delay and shift, which are inherited from classical signal processing, the process observed in the figure looks more like a kind of (fractional) diffusion. In fact, diffusion on graphs have been widely studied (Zhang and Hancock 2008; Thanou et al. 2017; Benzi and Simunec 2021); it is usually described in terms of a system of ordinary differential equations in time, with the Laplacian matrix of the graph as the coefficient matrix. Fractional diffusion has been used to model certain phenomena that allow long-range interactions and are non-local in nature (Ilic et al. 2005; Riascos and Mateos 2014; Estrada 2021; Antil et al. 2021).

Finally, it matters to highlight the fact that the signal to be shifted has to be bandlimited (see Fig. 4.3a). If the signal has abrupt changes in its sample values, this can be viewed as a kind of discontinuity and represents high frequency components, when compared to the predominantly smooth behavior of the signal (see Fig. 4.3b). As a consequence, we can observe considerable fluctuations around the disparate samples when the signal is fractionally delayed, an effect similar to the Gibbs phenomenon.

### 4.1.3 Consistency with classical approach: the ideal fractional delay filter

In the classical approach to the problem of fractionally shifting a discrete-time signal, the continuous-time version of the signal can be reconstructed, shifted and then resampled with the same sampling period (Oppenheim and Schafer 2013; Välimäki 1995). Due to the Nyquist-Shannon Theorem, this procedure requires that the signal is bandlimited. In this context, it can be shown that, if a discrete-time signal  $x$  is bandlimited, its version shifted by  $a \in [0, 1]$  is

$$x[n - a] = \sum_k x[k] \text{sinc}(n - k - a),$$

so that the (ideal low-pass) filter used to perform the referred shift has components

$$h_{L_{PF}}[n] = \text{sinc}(n - a). \quad (4.11)$$

The filter  $\mathbf{h}_{L_{PF}}$  is non-causal and unstable (it is not BIBO – *bounded input, bounded output*, because its impulse response has infinite energy) and, therefore, it is not physically realizable. In this way, fractional delay filter implementations should just *approximate*  $\mathbf{h}_{L_{PF}}$  as much as possible.

In order to evaluate how close to  $h_{L_{PF}}[n] = \text{sinc}(n - a)$ ,  $0 \leq a \leq 1$ , is  $h_a[n]$ , for odd  $N$  (see the first row of (4.10)), the point-wise difference between these signals has been computed for different values of  $N \in [10^1, 10^6]$ . Fig. 4.4, shows the relative error (ratio between the energy of the error ( $\mathbf{h}_a - \mathbf{h}_{L_{PF}}$ ) and that of  $\mathbf{h}_{L_{PF}}$ ), in terms of  $N$  and  $a$ .

The result suggests that, in fact,  $\mathbf{h}_a$  converges in the mean in  $\ell^2$  to  $\mathbf{h}_{L_{PF}}$  as  $N$  grows, with relative error less than 5% for  $N \approx 30$ . Moreover, the error is greater when  $a$  is close to 0.5, being negligible or null when  $a$  is an integer. In fact, the error is exactly zero for  $a = 0$  (or  $a = 1$ ) and  $n = a$ , since

$$\lim_{n \rightarrow a} h_a[n] = 1 \Rightarrow \lim_{n \rightarrow a} (h_a[n] - \text{sinc}(n - a)) = 0. \quad (4.12)$$

The same result is obtained for even  $N$ , starting from the second row of (4.10). When  $a$  is non-integer,  $h_a[n]$  is complex, with imaginary part of constant modulus for a fixed  $a$ . Considering the corresponding real part only, the error was smaller than that taking into account also the contribution of the imaginary part. Fig.

4.5 and Fig. 4.6 show that the errors with and without the imaginary part equally decay as  $N$  grows, but, using the real part only, the results are significantly better.

#### 4.1.4 Polynomial representation

The fractional shift matrix  $\mathbf{A}^a$  necessarily commutes with  $\mathbf{A}$ , because  $\mathbf{A}^a \mathbf{A} = \mathbf{A}^{1+a} = \mathbf{A} \mathbf{A}^a$ , so that  $\mathbf{A}^a$  is an LSI filter for signals on graphs having  $\mathbf{A}$  as adjacency matrix (see (3.22)). Therefore, according to Theorem 5,  $\mathbf{A}^a$  admits a polynomial representation like the one given in (3.23). In what follows, we evaluate such a possibility for directed ring graphs and for arbitrary graphs.

**Directed ring graphs.** The adjacency matrix  $\mathbf{C}$  in (3.19) of the directed ring graph with unitary weights satisfies  $\text{char}_{\mathbf{C}} = m_{\mathbf{C}}$  (due to the fact that the eigenvalues of  $\mathbf{C}$  are distinct). Therefore  $\mathbf{H} = \mathbf{C}^a$  can be directly expressed as a polynomial of degree up to  $(N - 1)$  in  $\mathbf{C}$ . In order to do this, we consider (3.29) and the fact that  $\mathbf{F}^{-1} = \mathbf{F}^H$ , with  $H$  indicating the conjugate transpose. This allows to show that  $\mathbf{C}^a = \mathbf{F}^H \mathbf{\Lambda}_{\mathbf{C}}^a \mathbf{F}$  is a circulant matrix with the first column given by  $\mathbf{h}_a$  in (4.10). Moreover, since the left product of a matrix by  $\mathbf{C}$  produces a circular down-shift in each column of the matrix, the  $N$  powers of  $\mathbf{C}$  form a basis for the space of  $N \times N$  circulant matrices (note that  $\mathbf{C}^N = \mathbf{C}^0$  is the identity matrix). From the above, we conclude that the coefficients of the polynomial representation of  $\mathbf{C}^a$  are the entries of  $\mathbf{h}_a$ , i. e.

$$\mathbf{H} = \mathbf{C}^a = \sum_{\ell=0}^{N-1} h_a[\ell] \mathbf{C}^\ell. \quad (4.13)$$

**Arbitrary graphs.** In order to demonstrate how to obtain the polynomial representation of  $\mathbf{H} = \mathbf{A}^a$  for arbitrary graphs, let us consider another strategy to compute matrix functions. By definition, the minimal polynomial  $m_{\mathbf{A}}(t)$  of  $\mathbf{A}$  is the unique monic polynomial of lowest degree such that  $m_{\mathbf{A}}(\mathbf{A}) = \mathbf{0}$ . By considering the Jordan canonical form of  $\mathbf{A}$ , it can be seen that

$$m_{\mathbf{A}}(t) = \prod_{i=1}^s (t - \lambda_i)^{n_i}. \quad (4.14)$$

It follows immediately that  $m_{\mathbf{A}}$  is zero on the spectrum of  $\mathbf{A}$ , that is, the values computed in (4.5) are all zero for  $f(t) = m_{\mathbf{A}}(t)$ . Given any polynomial  $p$  and any matrix  $\mathbf{A} \in \mathbb{C}^{N \times N}$ ,  $p$  is clearly defined on the spectrum of  $\mathbf{A}$  and  $p(\mathbf{A})$  can be defined by substitution. For polynomials  $p$  and  $q$ ,  $p(\mathbf{A}) = q(\mathbf{A})$  if and only if  $p$  and  $q$  take the same values on the spectrum. Thus the matrix  $p(\mathbf{A})$  is completely

determined by the values of  $p$  on the spectrum of  $\mathbf{A}$ . The following definition can then be established.

**Definition 3.** Let  $f$  be defined on the spectrum of  $\mathbf{A} \in \mathbb{C}^{N \times N}$ . Then  $f(\mathbf{A}) \stackrel{\Delta}{=} p(\mathbf{A})$ , where  $p$  is the unique polynomial of degree less than  $\sum_{i=1}^s n_i$  (which is the degree of the minimal polynomial) that satisfies the interpolation conditions

$$p^{(j)}(\lambda_i) = f^{(j)}(\lambda_i), \quad j = 0 : n_i - 1, \quad i = 1 : s. \quad (4.15)$$

The polynomial  $p$  above is known as the Hermite interpolating polynomial. In particular, if  $n_i = 1$ ,  $i = 1, \dots, s$ ,  $p$  corresponds to the Lagrange interpolating polynomial

$$p(t) = \sum_{i=1}^s f(\lambda_i) l_i(t), \quad l_i(t) = \prod_{j=1, j \neq i}^s \left( \frac{t - \lambda_j}{\lambda_i - \lambda_j} \right). \quad (4.16)$$

In any case, the results briefly presented above lead us to conclude that  $\mathbf{A}^a$  can be expressed as a polynomial in  $\mathbf{A}$  and, therefore, according to Theorem 5, the fractional shift of a graph signal can be implemented as an LSI graph filter.

## 4.2 Numerical Results

The last section discussed the effect of applying a fractional shift to a graph signal and demonstrated that  $\mathbf{A}^a$  admits a polynomial representation. In the first part of this section, a brief numerical example is presented to illustrate how the referred representation can be obtained. Then, the proposed fractional operator is applied in a context which brings to the front its practical advantage, namely the design of graph filters. Naturally, the resulting filter, being a polynomial in  $\mathbf{A}^a$ , could also be expressed as a polynomial in  $\mathbf{A}$  and, therefore, it is a LSI filter.

### 4.2.1 Example: Polynomial Representation of $\mathbf{A}^a$

The graph considered in this example is shown in Fig. 4.7 and has adjacency matrix

$$\mathbf{A} = \begin{bmatrix} 5 & 4 & 2 & 1 \\ 0 & 1 & -1 & -1 \\ -1 & -1 & 3 & 0 \\ 1 & 1 & -1 & 2 \end{bmatrix}. \quad (4.17)$$

The entries of  $\mathbf{A}$  in (4.17) were chosen so that the Jordan decomposition of such a matrix had integer entries only. The referred decomposition is written using matrices

$$\mathbf{V} = \begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 9 & -1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \quad \mathbf{V}^{-1} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

and

$$\mathbf{J} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 1 \\ 0 & 0 & 0 & 4 \end{bmatrix}. \quad (4.18)$$

Considering  $f(t) = t^{0.3}$  and Definition 2,  $f(\mathbf{A}) = \mathbf{A}^{0.3}$  can be computed according to

$$\mathbf{A}^{0.3} = \mathbf{V} \begin{bmatrix} f(1) & 0 & 0 & 0 \\ 0 & f(2) & 0 & 0 \\ 0 & 0 & f(4) & f'(4) \\ 0 & 0 & 0 & f(4) \end{bmatrix} \mathbf{V}^{-1},$$

which gives

$$\mathbf{A}^{0.3} = \begin{bmatrix} 1.6294 & 0.6294 & 0.3448 & 0.2311 \\ 0 & 1.0000 & -0.2311 & -0.2311 \\ -0.1137 & -0.1137 & 1.4020 & 0 \\ 0.1137 & 0.1137 & -0.1709 & 1.2311 \end{bmatrix}.$$

The same result can be achieved by using Definition 3, which gives

$$\begin{aligned} p(\mathbf{A}) &= f(\mathbf{A}) = \mathbf{A}^{0.3} \\ &= 0.6688\mathbf{I} + 0.3915\mathbf{A} - 0.0654\mathbf{A}^2 + 0.0051\mathbf{A}^3, \end{aligned}$$

the polynomial representation of  $\mathbf{A}^{0.3}$ .

#### 4.2.2 Least-square approximation of LSI filters

Before developing a numerical example illustrating the use of  $\mathbf{A}^a$  to filter graph signals, let us first review a simple design technique that are least-squares approximations of ideal LSI filters (Sandryhaila and Moura 2014b). Such a method consists of defining the (ideal) filter by specifying the values of  $h(\lambda_i)$  (filter response

in each eigenvalue of the shift operator), instead of determining the values of  $h_\ell$  (filter coefficients). Describing the frequency response of the filter for each eigenvalue  $\lambda_i$  yields the linear system of equations

$$\begin{aligned} h(\lambda_0) &= \alpha_0, \\ h(\lambda_1) &= \alpha_1, \\ &\vdots \\ h(\lambda_{N-1}) &= \alpha_{N-1}, \end{aligned} \tag{4.19}$$

or, since  $h(\cdot)$  is a polynomial of degree  $L$ ,

$$\begin{aligned} h_0 + h_1\lambda_0 + \cdots + h_L\lambda_0^L &= \alpha_0, \\ h_0 + h_1\lambda_1 + \cdots + h_L\lambda_1^L &= \alpha_1, \\ &\vdots \\ h_0 + h_1\lambda_{N-1} + \cdots + h_L\lambda_{N-1}^L &= \alpha_{N-1}. \end{aligned} \tag{4.20}$$

Using a Vandermonde matrix constructed from the eigenvalues  $\lambda_i$ , the system (4.20) can be written in matrix form as

$$\begin{bmatrix} 1 & \lambda_0 & \lambda_0^2 & \dots & \lambda_0^L \\ 1 & \lambda_1 & \lambda_1^2 & \dots & \lambda_1^L \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \lambda_{N-1} & \lambda_{N-1}^2 & \dots & \lambda_{N-1}^L \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_L \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{N-1} \end{bmatrix}. \tag{4.21}$$

More specifically, if one desires to design a low-pass filter (LPF) whose cutoff frequency is  $\lambda_{i_{\text{cut}}}$ , one could set

$$\alpha_j = \begin{cases} 1, & \text{for } j = 0, \dots, i_{\text{cut}}, \\ 0, & \text{for } j = i_{\text{cut}} + 1, \dots, N - 1. \end{cases} \tag{4.22}$$

Since one generally has  $N \geq L + 1$ , the system of equations (4.21) is *overdetermined* and does not have an exact solution. A possible strategy is to find coefficients  $h_\ell$ ,  $\ell = 0, \dots, L$ , that minimize, in the least-squares sense, the deviation from the ideal filter response. This corresponds to solve the optimization problem

$$\min_{\{h_\ell\}_{0,\dots,L}} \sum_{n=0}^{N-1} (h(\lambda_n) - \alpha_n)^2. \tag{4.23}$$

**Proposed application of the fractional shift.** The proposed change to the method above, which includes the fractional shift in the filter design, consists of replacing  $\mathbf{A}$  with  $\mathbf{A}^a$  in (3.23), representing the LSI filter as

$$h(\mathbf{A}^a) = \sum_{\ell=0}^L h_\ell \mathbf{A}^{a \cdot \ell}. \quad (4.24)$$

If this is performed, the only adjustment needed in the technique described above consists of replacing the eigenvalues  $\lambda_i$  with their  $a^{\text{th}}$  powers  $\lambda_i^a$  in (4.20). The effect of such a substitution is illustrated and evaluated in what follows.

### 4.2.3 Example: LS Approximation using $\mathbf{A}^a$

In this example, we consider a network formed by 230 weather stations that measure daily temperature across the United States (*National Climatic Data Center* n.d.). Such stations are represented by the vertices of an undirected graph whose edges have been established by using the 8-nearest neighbor criterion. The edge connecting vertices  $v_n$  and  $v_m$  is weighted according to

$$\mathbf{A}_{n,m} = \frac{e^{-d_{n,m}^2}}{\sqrt{\sum_{k \in \mathcal{N}_n} e^{-d_{n,k}^2} \sum_{\ell \in \mathcal{N}_m} e^{-d_{n,\ell}^2}}}, \quad (4.25)$$

where  $d_{n,m}$  denotes the geodesical distance between the  $n^{\text{th}}$  and the  $m^{\text{th}}$  sensors. The snapshot of all measurements taken on February 1<sup>st</sup>, 2003 forms the signal indexed by the referred graph, which is shown in Fig. 4.8. From the GFT of the signal, which is plotted in Fig. 4.9, it can be seen that its spectral content is concentrated in the low graph frequencies. Note that such frequencies correspond to the eigenvalues of  $\mathbf{A}$ , which are marked along the horizontal axis of the figure; additionally, the referred marking accompanies the fact that low (resp. high) graph frequencies are associated with higher (resp. lower) eigenvalues (Sandryhaila and Moura 2014b).

We then use the strategy explained in Subsection 4.2.2 to design a filter that approximates an ideal low-pass filter with  $\lambda_{i_{\text{cut}}} = 0.2$ . In this case,  $i_{\text{cut}} = 39$  so that the 40 lowest graph frequencies are (ideally) preserved after the signal is filtered. We considered approximations with  $L$  ranging from 1 to 19, that is, filters with 2 to 20 coefficients. For each of these values, we varied the fractional parameter  $a$  from 0 to 1 and, in (4.21), after replacing  $\lambda_i$  with  $\lambda_i^a$ ,  $i = 0, 1, \dots, N - 1$ , and solving (6.20)<sup>2</sup>, we registered the value of  $a$  providing the minimum error between

---

<sup>2</sup>The optimization problem 6.20 has been solved using the Linear Algebra module *linalg* for Scipy, a free and open-source Python library used for scientific and technical computing. In all experiments

the designed filter and the ideal filter. At the end of this procedure, the plot shown in Fig. 4.10 was produced. Observing the figure, we verify that, for any value of  $L$ , the best approximation is provided when  $a \neq 1$ . This is enough to conclude that, for the graph considered in the example, the use of a fractional version  $\mathbf{A}^a$ ,  $a \neq 1$ , of  $\mathbf{A}$  always provides a better result than the one obtained with the non-fractional matrix. A visual comparison between these alternatives can be performed from Fig. 4.11, where we show the (minimum) errors we have just referred to together with the errors when the original (non-fractional) matrix  $\mathbf{A}$  is employed.

In Fig. 4.12, we can observe the ideal filter response superimposed on the responses obtained when  $\mathbf{A}$  and  $\mathbf{A}^a$  are used to design a filter with  $L + 1 = 10$  coefficients. In this case, the fractional parameter providing the minimum error is  $a = 0.855$ . In the figure, we notice that the filter designed with  $\mathbf{A}^a$  has fluctuations that deviate less from the ideal filter, when compared to those related to the filter designed using  $\mathbf{A}$ . This can be observed mainly in the passband and constitutes a visual result coherent with the obtained approximation errors. Plots with similar behaviour are obtained for other values  $L$ .

#### 4.2.4 Example: noise removal via LSI low-pass filtering

In this example, we start from the same graph signal considered in Subsection 4.2.3. We add to the samples of the referred signal random uniformly-distributed values whose amplitude corresponds to a percentage of the range of the signal itself. Such a synthetic noise addition is intended to simulate what happens in many practical scenarios, in which measurements performed on a sensor network are subject to different sources of distortion. The resulting noisy signal is then filtered by the filters shown in Fig. 4.12, as an attempt to reduce the influence of the noise and recover the original signal.

During the experiment, the aforementioned percentage was varied from 1% to 50% and, for each of these values, 100 noisy signals were generated. After low-pass filtering, the mean squared error between the original and filtered graph signals were computed. The results show that the filter designed using  $\mathbf{A}^{0.855}$  allows to recover the signal with average reconstruction error always smaller than that with the filter designed using  $\mathbf{A}$ ; see the data in Fig. 4.13.

In this context, it is relevant to remark that the (best) fractional parameter  $a = 0.855$  has been found using the strategy described in the second paragraph of

---

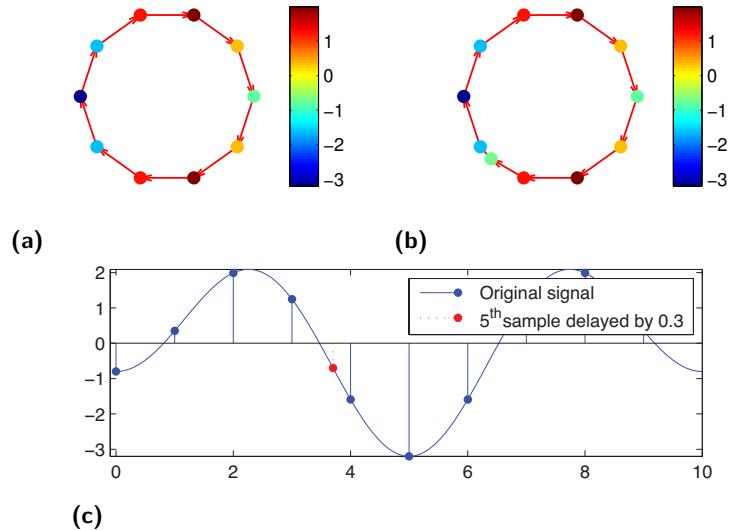
performed, the least mean squares algorithm converged and the time required for this was negligible, considering the addressed application scenario.

Subsection 4.2.3, which depends on the error between the designed filter and the ideal filter only. Therefore, the referred choice does not require prior knowledge of the original signal beforehand, which is usually not available in real-world problems. This illustrates the potential gain that can be achieved, in this application scenario, when considering the possibility of fractionalization of the graph shift operator.

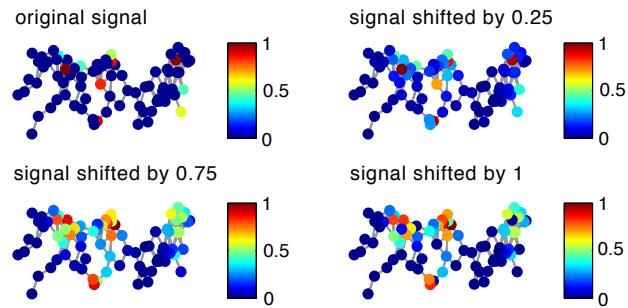
Finally, it is also interesting to mention that only one or a few nodes could have had their measurements corrupted by noise or changed due to other factors; this would represent, for instance, a scenario in which certain sensors would be malfunctioning. In order to obtain some preliminary results taking into account the above described assumption, additional simulations were carried out. The previous tests were repeated, but assuming that only a number from 1 to 12 nodes had their values nullified or increased by 20 times. We then performed a low-pass filtering, expecting that the high-frequency component associated with the referred measurement changes would be attenuated and that the smooth behavior of the signal would be recovered. In general, the results obtained using the proposed fractional operator were better or at least equivalent to those obtained with the corresponding ordinary operator. Future works may address this issue in more detail.

## 4.3 Concluding remarks

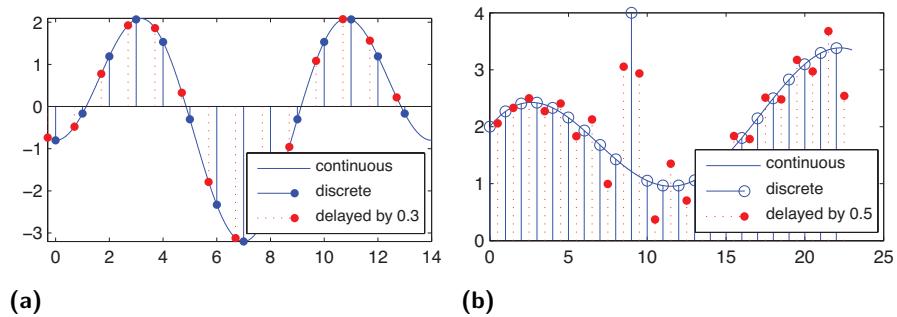
This chapter proposed and discussed the fractional shift operator of graph signals. The core idea for the contribution was that, in the GSP theory, the unit shift is defined from the adjacency matrix of a graph. Interpreting the fractional shift as a filtering operation, it was demonstrated that, for ring graphs, its application produces the expected effect of approximating the classical ideal interpolating filter, exhibiting satisfactory results for bandlimited signals. Moreover, it was also shown that the referred fractional operator can be implemented as an LSI graph filter for arbitrary graphs and real-world examples were presented, illustrating the benefits of using  $\mathbf{A}^a$  to design graph filters for noise removal.



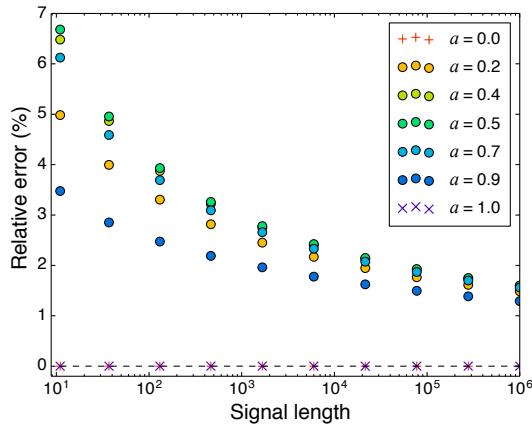
**Fig. 4.1.:** Fractional shift by  $a = 0.3$  of a sample of a signal on a directed ring graph with unit weights. (a) Original signal on a directed ring graph. (b) Graph in which the 5<sup>th</sup> sample delayed by  $a = 0.3$  appears as an interpolated sample between the 4<sup>th</sup> and the 5<sup>th</sup> samples of the original signal. (c) Original discrete signal and the delayed sample.



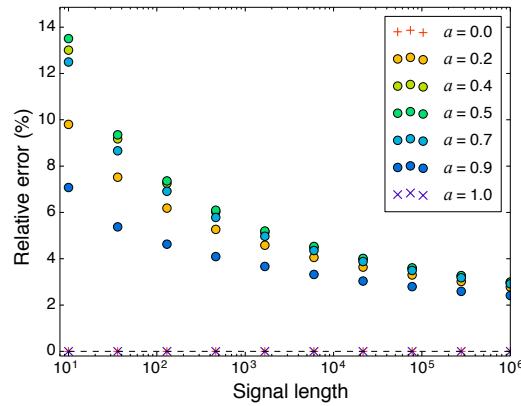
**Fig. 4.2.:** Fractional shift of a signal, (originally) with 10 non-zero samples, defined on a graph formed by 80 cities of Pernambuco state, Brazil. Note that the shifted signal is similar to the original signal, if  $a$  is close to 0, and similar to the unit-shifted signal, if  $a$  is close to 1.



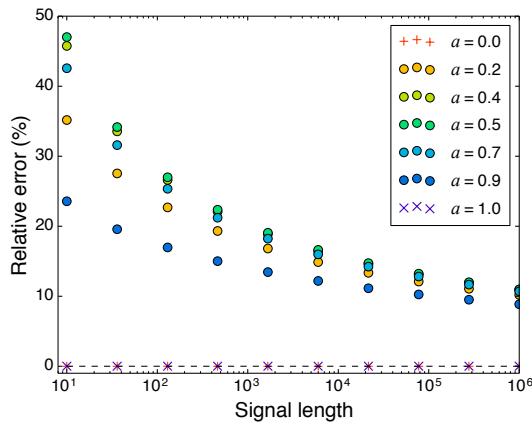
**Fig. 4.3.:** Fractional shift for a signal (a) without and (b) with abrupt variations (descontinuities).



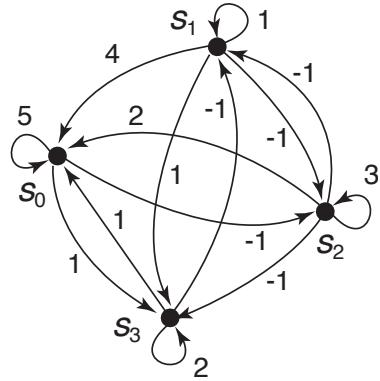
**Fig. 4.4.:** Percent error (normalized by the energy of  $\mathbf{h}_{LPP}$ ) of  $\mathbf{h}_a$  related to  $\mathbf{h}_{LPP}$ , for different (odd) values of  $N$  and the fractional shift parameter  $a$ .



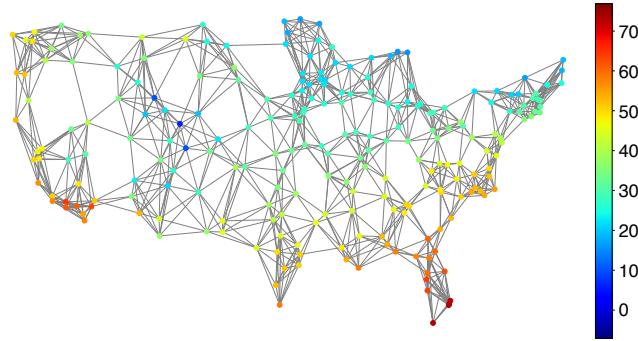
**Fig. 4.5.:** Relative mean error between  $\mathcal{R}e\{\mathbf{h}_a\}$  and  $\mathbf{h}_{LPP}$  for  $N$  even, in terms of the fractional shift parameter  $a$ .



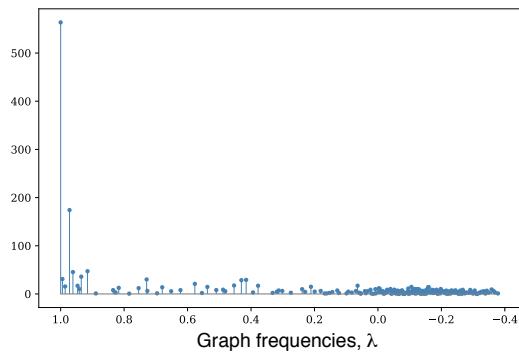
**Fig. 4.6.:** Modulus of the relative mean error between  $\mathbf{h}_a$  and  $\mathbf{h}_{LPP}$  for  $N$  even, in terms of the fractional shift parameter  $a$ .



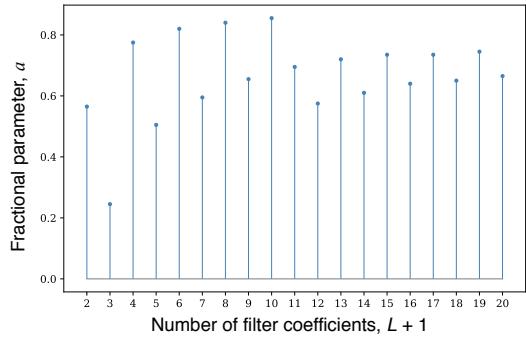
**Fig. 4.7.:** Directed graph used to illustrate how the corresponding fractional shift operator can be computed and represented in polynomial form.



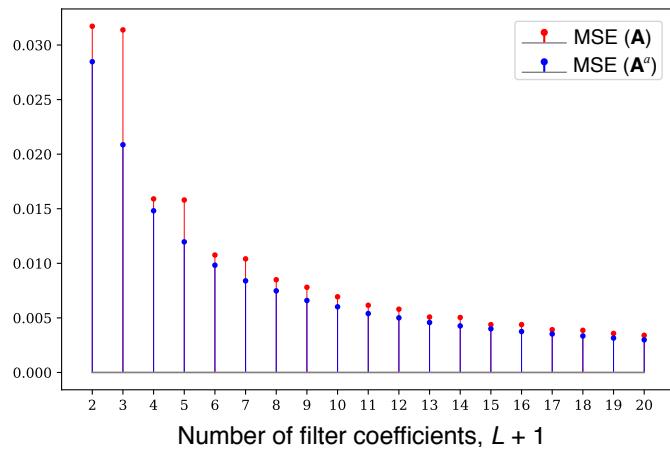
**Fig. 4.8.:** Graph of a network formed by 230 weather stations measuring the temperature across the United States. The snapshot of all measurements taken on February 1<sup>st</sup>, 2003 is the corresponding graph signal.



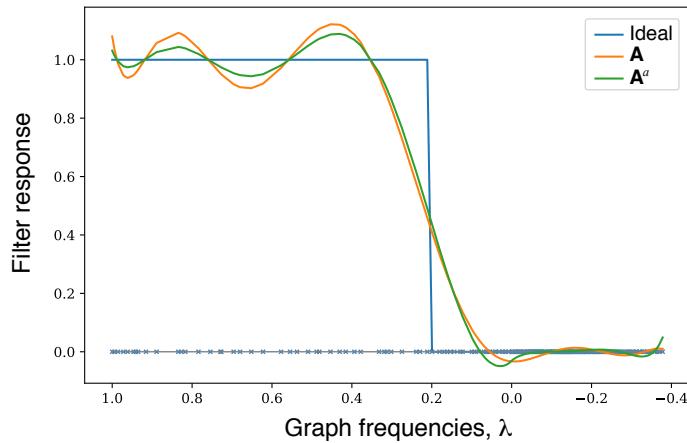
**Fig. 4.9.:** Magnitude of the graph Fourier transform of the signal in Fig 4.8. The graph frequencies correspond to the eigenvalues of  $\mathbf{A}$ ; low (resp. high) graph frequencies are associated with higher (resp. lower) eigenvalues.



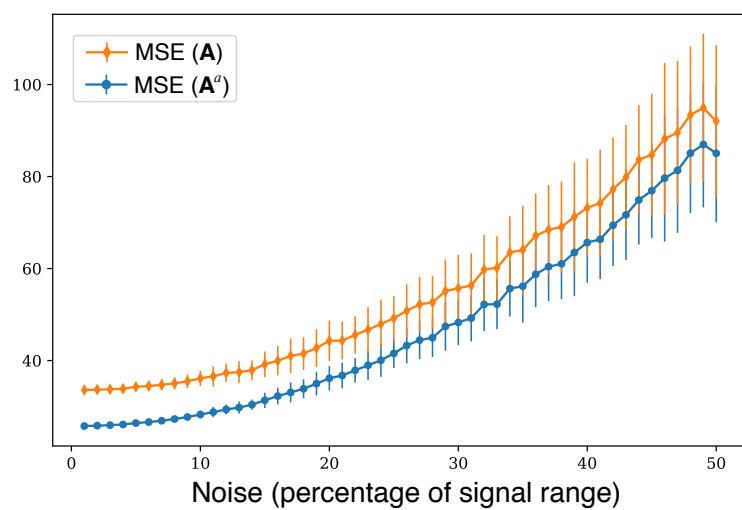
**Fig. 4.10.:** Fractional parameters providing the minimum approximation errors, for different values of  $L$ , between the ideal LPF and the filter designed by using the fractional graph shift operator  $\mathbf{A}^a$ .



**Fig. 4.11.:** Minimum approximation (mean squared) errors, for different values of  $L$ , between the ideal LPF and the filters designed by using the fractional graph shift operator  $\mathbf{A}^a$  and the non-fractional operator  $\mathbf{A}$ .



**Fig. 4.12.:** Ideal filter response superimposed on the responses obtained when  $\mathbf{A}$  and  $\mathbf{A}^a$ ,  $a = 0.855$ , are used to design a filter with  $L + 1 = 10$  coefficients.



**Fig. 4.13.:** Reconstruction (mean squared) errors after a noise removal procedure is performed by using graph filters with  $L + 1 = 10$  coefficients and designed from  $\mathbf{A}$  and  $\mathbf{A}^a$ ,  $a = 0.855$ .



# The fractional quaternion discrete Fourier transform and its applications

Linear discrete transforms are building blocks for a multitude of techniques in the field of signal processing, being almost as important as they are diverse. Among the factors which distinguish one from the others, there is the algebraic structure over which they are defined, e. g. a finite field and its extensions (such as in number-theoretic (Blahut 2010; Pedrouzo-Ulla et al. 2017; Chandra and Svalbe 2014; Lima and Campello de Souza 2013) and arithmetic (Knockaert 1994; Rajapaksha et al. 2014) transforms), or the real and complex fields (as in the usual discrete Fourier transform). Let us recall an important point that has already been made: as the algebra is extended (for instance, from  $\mathbb{R}$  to  $\mathbb{C}$ ), it is possible to encode more information into each signal sample. Such was the motivation behind Sangwine's definition (Sangwine 1996) of discrete two-dimensional quaternion transforms, based on their continuous counterparts previously defined by Ell (Ell 1993): to apply these four-dimensional numbers — the quaternions, an extension of the complex field — to color image processing. Since then, quaternion transforms have been useful not only to image processing (Ell and Sangwine 2007; Chen et al. 2018b; Li 2013; Evans et al. 2000; Silva Jr. and Lima 2018), but also to other fields, such as bivariate signal analysis (Flamant et al. 2017b; Flamant et al. 2017a; Flamant et al. 2018).

Fractional transforms are yet another class of tools of great use, the reader may remember the review presented in Chapter 3. Regarding quaternion transforms, a couple of competent works have already addressed their fractionalization (Guanlei et al. 2008; Wei and Li 2013; Roopkumar 2016) and some applications have been proposed (Chen et al. 2018b). However, the author is not aware of papers approaching fractional quaternion discrete transforms *from an eigenstructure analysis point of view*. This reasoning may unfold new theoretical insights and implementation techniques, and such is the motivation for this work.

In this chapter, the eigenstructure of the quaternion discrete Fourier transform (QDFT) matrix is investigated and shown to be closely related to that of the

unitary discrete Fourier transform (DFT). This result offers an approach for defining the fractional version of the QDFT (referred to as FrQDFT). Following the central goal of defining the transform through eigendecomposition theory, a generalization is proposed in the form of a multiparametric fractional quaternion discrete Fourier transform (MFrQDFT). For illustrative purposes, this work proposes and briefly explores an encryption scheme for color images with opacity layer, fully harnessing the holistic processing of 4-channels 2D signals through the MFrQDFT.

The proposed method for image encryption fits in the class of schemes that use linear transforms alongside non-linear blocks (Hsue 2018), to implement confusion and diffusion. The currently available methods in the state-of-the-art literature respond to a diverse range of needs, as fast implementation through parallel computing (Wang et al. 2019b), increased safety and robustness by using matrix semi-tensor product (Wang and Gao 2020) or one-time keys (Liu and Wang 2010), just to name a few examples. The main scope of this chapter is not the encryption scheme *per se*, but rather the FrQDFT eigenscructure and the MFrQDFT definition, the latter having the image encryption as a framework to showcase some of its possibilities. Nevertheless, the proposed encryption algorithm is described and evaluated to some extent. Although being an illustrative scenario, it stands on tools adopted currently by the literature. For example, the chosen method for key generation involved the use of chaotic maps, known to help achieving high bit sensibility and large key space. It follows works such as the one by Liu and Wang (Liu and Wang 2011), which proposed a color image encryption scheme using two chaotic maps and one-time keying, aiming to achieve large key space and cycle lengths. Another work by Liu and Wang (Liu, Wang, et al. 2012) employed a piecewise linear chaotic map and DNA encoding for ensuring the initial conditions of the encryption change according to the image, whereas Wang *et al.* (Wang et al. 2010) used a Lorentz chaotic map and a perceptron model applied to image encryption. The proposed scheme uses a chaotic tent map to generate a pseudo-random sequence, from which the secret parameters are extracted.

This chapter is structured as follows. Section 5.1 presents the usual definition of the QDFT and proves the central theorem of this work, regarding how the DFT and QDFT share symmetric eigenvectors; it closes with comments on the quaternion representation of color images with opacity layer. Section 5.2 discusses the fractionalization of the QDFT from an eigenstructure point of view and demonstrates some properties purely based on matrix algebra. Section 5.3 proposes a multiparametric extension of the fractional transform and presents an application involving encryption of color images with opacity layer. The main results are summarized in Section 5.4.

## 5.1 Eigenstructure of the QDFT

Let us recall the definition of the 1D left QDFT of axis  $\mu$  – a unit pure quaternion, as presented in (Ell et al. 2014, Sec. 3.3.1) and in (5.1). The  $m$ -th entry of the QDFT of vector  $\mathbf{v}$  is

$$\hat{v}_m = \text{QDFT}\{\mathbf{v}\}_m \triangleq \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \exp\left(-\mu \frac{2\pi}{N} nm\right) v_n \in \mathbb{C}_\mu, \quad (5.1)$$

in which  $\mathbb{C}_\mu$  denotes the set of numbers  $a + \mu b$ ,  $a, b \in \mathbb{R}$ , isomorphic to the complex set. It matters to notice that, due to the lack of commutativity in quaternion multiplication, the position of the kernel relative to the operand in (5.1) is relevant and must be kept consistent along all computations. Hence, the inverse transformation must be computed with the kernel on the same relative position,

$$v_n = \text{QDFT}^{-1}\{\hat{\mathbf{v}}\}_n = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} \exp\left(\mu \frac{2\pi}{N} nm\right) \hat{v}_m. \quad (5.2)$$

The synthesis and analysis equations may be written in matrix form as

$$\hat{\mathbf{v}} = \text{QDFT}\{\mathbf{v}\} = \mathbf{F}\mathbf{v}, \quad (5.3)$$

$$\mathbf{v} = \text{QDFT}^{-1}\{\hat{\mathbf{v}}\} = \mathbf{F}^{-1}\hat{\mathbf{v}}, \quad (5.4)$$

where  $\mathbf{F}$  is the unitary QDFT matrix, with entries  $\{\mathbf{F}\}_{n,m} = \sqrt{N}^{-1} \exp\left(-\mu \frac{2\pi}{N} nm\right)$ . Since  $\exp\left(-\mu \frac{2\pi}{N}\right)$  is an  $N$ -th root of unity, such as  $\exp\left(-i \frac{2\pi}{N}\right)$ , it follows that  $\mathbf{F}$  shares many properties of the DFT matrix, such as invertibility (simple calculations show that  $\mathbf{F}^{-1} = \mathbf{F}^H$ ), what guarantees validity of the inversion formulae in (5.2) and (5.4).

The two-dimensional QDFT can be defined in a similar fashion, although more options regarding kernel positioning are available, since for any pure quaternions  $\mu \neq \nu$ , one may verify that generally  $e^{\nu\alpha} e^{\mu\beta} \neq e^{\nu\alpha+\mu\beta}$ . Ell and Sangwine (Ell et al. 2014) presented the *eight* distinct ways of building a 2D-QFT, which translate directly into options for 2D-QDFT. One of such possibilities is to transform the quaternion-valued matrix  $\mathbf{X} \in \mathbb{H}^{N \times M}$  according to the equation

$$\hat{X}_{u,k} = \text{2D-QDFT}\{\mathbf{X}\}_{u,k} \triangleq \frac{1}{\sqrt{MN}} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \exp\left(-\mu \frac{2\pi}{N} nu\right) X_{n,m} \exp\left(-\nu \frac{2\pi}{M} mk\right). \quad (5.5)$$

This formulation of the 2D-QDFT translates into the following matrix equation,

$$\hat{\mathbf{X}} = \mathbf{F}^{(\mu)} \mathbf{X} \mathbf{F}^{(\nu)}, \quad (5.6)$$

where the Fourier matrices are similar to the one used in (5.3), except from the pure quaternion which serves as transform axis (shown in the parenthesis). This is a consequence of the *separability* of the 2D-QDFT, by which this transform may be conceived as the successive application of two 1D-QDFTs: once in the rows of  $\mathbf{X}$ , once in the columns. Therefore, some results and properties derived for the 1D-QDFT may naturally extend to the two-dimensional case.

The similarities between the QDFT and the DFT matrices hugely aid the investigation of the eigenstructure of matrix  $\mathbf{F}$ . As a result of Theorem 6, which is a contribution of this thesis, one is able to deduce the QDFT eigenstructure out of even and odd DFT eigenvectors, by using a variation of Pei's reasoning regarding the 2D-QDFT (Pei et al. 2010).

**Theorem 6.** *Let  $\mathbf{v}$  be an eigenvector of the unitary DFT with eigenvalue  $\lambda$ .*

- (a) *If  $\mathbf{v}$  has even symmetry (in which case  $\lambda = \pm 1$ ), then it is also an eigenvector of the QDFT with eigenvalue  $\lambda$ .*
- (b) *If  $\mathbf{v}$  has odd symmetry (in which case  $\lambda = \pm i$ ), then it is also an eigenvector of the QDFT (of axis, let us say,  $\mu$ ) with eigenvalue  $-\lambda i\mu$ , i. e.,  $\pm\mu$ .*

*Proof.* (a) If  $\mathbf{v}$  has even symmetry, i. e.,  $v_n = v_{N-n}$  for  $n = 1, \dots, N-1$ , then

$$\sum_{n=0}^{N-1} v_n \sin \frac{2\pi}{N} nm = 0, \quad (5.7)$$

therefore,

$$\begin{aligned} \sqrt{N} \text{QDFT}\{\mathbf{v}\}_m &= \sum_{n=0}^{N-1} v_n e^{-\mu \frac{2\pi}{n} nm} \\ &= \sum_{n=0}^{N-1} v_n \left( \cos \frac{2\pi}{n} nm - \mu \sin \frac{2\pi}{n} nm \right) \\ &= \left( \sum_{n=0}^{N-1} v_n \cos \frac{2\pi}{n} nm \right) - \underbrace{\left( \sum_{n=0}^{N-1} v_n \sin \frac{2\pi}{n} nm \right)}_{=0} \mu \\ &= \left( \sum_{n=0}^{N-1} v_n \cos \frac{2\pi}{n} nm \right) - \underbrace{\left( \sum_{n=0}^{N-1} v_n \sin \frac{2\pi}{n} nm \right)}_{=0} i \end{aligned}$$

$$= \sqrt{N} \text{DFT}\{\mathbf{v}\}_m = \sqrt{N} \lambda v_m \\ \Rightarrow \text{QDFT}\{\mathbf{v}\} = \lambda \mathbf{v}.$$

(b) If  $\mathbf{v}$  has odd symmetry, i. e.,  $v_n = -v_{N-n}$  for  $n = 1, \dots, N-1$  and  $v_0 = 0$ , then

$$\sum_{n=0}^{N-1} v_n \cos \frac{2\pi}{N} nm = 0, \quad (5.8)$$

hence,

$$\begin{aligned} \sqrt{N} \text{QDFT}\{\mathbf{v}\}_m &= \sum_{n=0}^{N-1} v_n e^{-i \frac{2\pi}{N} nm} \\ &= \sum_{n=0}^{N-1} v_n \left( \cos \frac{2\pi}{N} nm - i \sin \frac{2\pi}{N} nm \right) \\ &= \underbrace{\left( \sum_{n=0}^{N-1} v_n \cos \frac{2\pi}{N} nm \right)}_{=0} - \left( \sum_{n=0}^{N-1} v_n \sin \frac{2\pi}{N} nm \right) i \mu \\ &= - \left( \sum_{n=0}^{N-1} v_n \sin \frac{2\pi}{N} nm \right) i \mu. \end{aligned} \quad (5.9)$$

But, from the odd symmetry assumption,

$$\sqrt{N} \lambda v_m = \sqrt{N} \text{DFT}\{\mathbf{v}\}_m = \sum_{n=0}^{N-1} v_n e^{-i \frac{2\pi}{N} nm} = - \left( \sum_{n=0}^{N-1} v_n \sin \frac{2\pi}{N} nm \right) i, \quad (5.10)$$

therefore (remember that  $\lambda$  and  $v_m$  commute)

$$\sum_{n=0}^{N-1} v_n \sin \frac{2\pi}{N} nm = \sqrt{N} v_m \lambda i. \quad (5.11)$$

From (5.9) and (5.11),

$$\begin{aligned} \sqrt{N} \text{QDFT}\{\mathbf{v}\}_m &= - \sqrt{N} \text{DFT}\{\mathbf{v}\}_m i \mu \\ &= - \sqrt{N} v_m \lambda i \mu \\ \Rightarrow \text{QDFT}\{\mathbf{v}\} &= - \lambda i \mu \mathbf{v}. \end{aligned} \quad (5.12)$$

□

The results of Theorem 6 are summarized in Table 5.1. This analysis can immediately be extended to the two-dimensional case if one considers the separability of the 2D-QDFT, mentioned after (5.6). If  $(\mathbf{e}^{(\mu)}, \lambda_\mu)$  and  $(\mathbf{e}^{(\nu)}, \lambda_\nu)$  are

(column-)eigenvector-eigenvalue pairs of the matrices  $\mathbf{F}^{(\mu)}$  and  $\mathbf{F}^{(\nu)}$ , respectively, then the 2D signal  $\mathbf{e}^{(\mu)}\mathbf{e}^{(\nu)^T}$  is a 2D-QDFT eigenvector with eigenvalue  $\lambda_\mu\lambda_\nu$  (Candan 2011). Therefore, this 2D-QDFT has eight (possibly) distinct eigenvalues:  $\pm 1, \pm \mu, \pm \nu, \pm \mu\nu$ .

### 5.1.1 2D-QDFT and color images with opacity layer

The 2D-QDFT has been commonly used to process color images, which are represented as matrices of pure quaternions (Lu et al. 2007; Ell and Sangwine 2006; Chen et al. 2018a). In this representation, each color channel of a certain pixel corresponds to an imaginary component of the pure quaternion. Although this mapping is useful and adequate, it neglects the quaternion scalar part (always set to zero), causing a difference in dimensionality between input and output of the 2D-QDFT: while the image is a 2D signal with three-dimensional components, its spectrum has four-dimensional entries. It surely is not a problem *per se*, rather is an inconvenience, also found when processing real signals with the DFT.

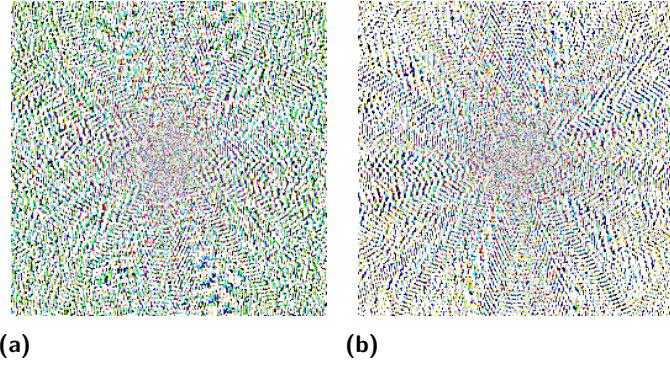
An application free from this inconvenience is the analysis of color images with opacity (or alpha) layer, as in files in portable network graphics format (PNG). In this case, each pixel  $(R, G, B, \alpha)$  is mapped into  $q = \alpha + Ri + Gj + Bk$ , forming the quaternion-valued matrix  $\mathbf{X}$ . Let us compute the 2D-QDFT by separately transforming the rows and columns using (5.6), with the same transform axis on both sides,

$$\text{2D-QDFT}\{\mathbf{X}\} = \mathbf{F}\mathbf{X}\mathbf{F}^T. \quad (5.13)$$

One should notice that the transformation in (5.13) does *not* consist on the successive application of the *same* QDFT to the rows and columns of matrix  $\mathbf{X}$ . Due to the noncommutative nature of quaternion multiplication, as it was previously mentioned, different transforms are obtained when choosing between left- or right-multiplications, even though the transform axis is kept unchanged. The operation in (5.13) performs a *left* QDFT on the columns and a *right* QDFT on the rows, as it is

**Tab. 5.1.:** DFT and QDFT eigenvectors.

| Eigenvector symmetry | Eigenvalue (DFT) | Eigenvalue (QDFT)       |
|----------------------|------------------|-------------------------|
| Even                 | $\pm 1$          | $\pm 1$                 |
| Odd                  | $\pm i$          | $-(\pm i)i\mu = \pm\mu$ |



**Fig. 5.1.:** (a) 2D-QDFT of the PNG image in Fig. 5.2a, according to (5.13), with axis  $\mu = \frac{1}{\sqrt{3}}(i + j + k)$ . (b) 2D-QDFT of the same image, same transform axis, following the approach in (5.14).

clear from (5.5). A 2D-QDFT consisting of the *same* transformation applied in both dimensions must use multiplications with the same orientation, e. g.

$$\mathbf{F} (\mathbf{F} \mathbf{X}^T)^T. \quad (5.14)$$

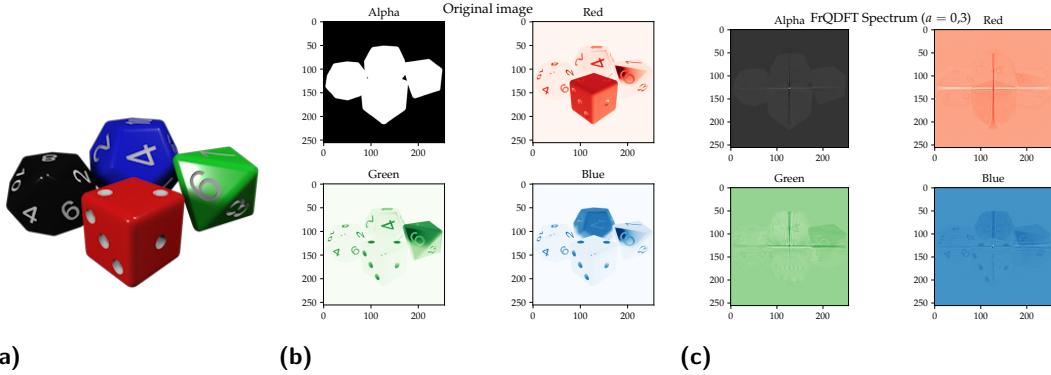
Fig. 5.1a shows the QDFT – according to (5.13) – of Fig. 5.2a, as another PNG image. As an illustration of the difference in using (5.13) or (5.14), caused by the lack of commutativity in quaternion multiplication, the mean squared error (MSE) between the two spectra was computed. The result was approximately 1070.

## 5.2 The fractional quaternion discrete Fourier transform

The proposed fractionalization method for the QDFT explores the eigenvector sharing between the DFT and the QDFT: as long as one possesses an orthogonal eigenvector matrix  $\mathbf{E}$  for the DFT, it can be used for the QDFT matrix diagonalization and its subsequent fractionalization. For instance, the eigendecomposition of the DFT matrix allows to find its fractional counterpart by raising each eigenvalue to a non-integer parameter  $a$ , i. e.

$$\mathbf{F}_{\text{DFT}}^a = \mathbf{E} \Lambda^a \mathbf{E}^T, \quad (5.15)$$

where  $\Lambda$  is the diagonal matrix containing the DFT eigenvalues.



**Fig. 5.2.:** (a) Test PNG image. Visualization of each layer in the (b) test image and (c) in its FrQDFT spectrum, computed with transform axis  $\mu = \frac{1}{\sqrt{3}}(i + j + k)$  and  $a = 0.3$ .

Oliveira Neto and Lima (Oliveira Neto and Lima 2017) stress that a FrDFT expressed as in (5.15) will numerically approximate its continuous version if and only if the columns of  $\mathbf{E}$  approximate samples of continuous Hermite-Gaussian functions. In (Oliveira Neto and Lima 2017), the authors present two methods to generate such an orthogonal eigenbasis, one of which (the generating matrix method) is the one adopted in this work.

Once one is able to compute an orthogonal eigenbasis  $\mathbf{E}$  containing Hermite-Gaussian-like DFT eigenvectors, for instance by means of the generating matrix method, Theorem 6 assures that matrix  $\mathbf{E}$  is also an eigenvector matrix for the QDFT. As a consequence, the transform matrix  $\mathbf{F}$  may be decomposed and written as

$$\mathbf{F} = \mathbf{E}\boldsymbol{\Gamma}\mathbf{E}^T, \quad (5.16)$$

in which the diagonal matrix  $\boldsymbol{\Gamma}$  is obtained by replacing  $i$  with  $\mu$  in  $\Lambda$  (Table 5.1). Therefore, the *fractional* quaternion Fourier transform, or simply FrQDFT, is obtained by raising each eigenvalue in  $\boldsymbol{\Gamma}$  to a so-called fractional order  $a \in \mathbb{R}$ , so that

$$\text{FrQDFT}_a\{\mathbf{v}\} \triangleq \mathbf{F}^a \mathbf{v}, \quad (5.17)$$

where

$$\mathbf{F}^a = \mathbf{E}\boldsymbol{\Gamma}^a\mathbf{E}^T. \quad (5.18)$$

The FrQDFT, as defined in (5.17) and (5.18), possesses all the classical properties of a fractional Fourier transform:

- *Reduction to the ordinary quaternion transform:* if  $a = 1$ , then the synthesis equation in (5.17) equals (5.3), coinciding with the QDFT. The proof is immediate.
- *Reduction to the identity:* if  $a = 0$ , the FrQDFT reduces to the identity operator, represented as  $\mathbb{I}$ .

*Proof.* From the orthogonality of the eigenvector matrix  $\mathbf{E}$ ,

$$\mathbf{F}^0 = \mathbf{E}\Gamma^0\mathbf{E}^T = \mathbf{E}\mathbf{E}^T = \mathbb{I}. \quad (5.19)$$

□

- *Index additivity:* applying the FrQDFT twice, using  $a$  and  $b$  as fractional orders, equals applying a single FrQDFT with fractional order  $a + b$ . Equivalently,  $\mathbf{F}^a\mathbf{F}^b = \mathbf{F}^{a+b}$ .

*Proof.*

$$\begin{aligned} \mathbf{F}^a\mathbf{F}^b &= \mathbf{E}\Gamma^a\mathbf{E}^T\mathbf{E}\Gamma^b\mathbf{E}^T \\ &= \mathbf{E}\Gamma^a\Gamma^b\mathbf{E}^T, \end{aligned} \quad (5.20)$$

but, since  $\Gamma$  is a diagonal matrix,  $\Gamma^a\Gamma^b = \Gamma^{a+b}$ , hence

$$\mathbf{F}^a\mathbf{F}^b = \mathbf{E}\Gamma^a\Gamma^b\mathbf{E}^T = \mathbf{E}\Gamma^{a+b}\mathbf{E}^T = \mathbf{F}^{a+b}. \quad (5.21)$$

□

- *Unitary matrix:* the matrix  $\mathbf{F}^a$  is unitary, i. e.

$$\mathbf{F}^a(\mathbf{F}^a)^H = \mathbb{I}, \quad (5.22)$$

in which  $(\cdot)^H$  denotes the Hermitian (conjugate transpose) operator.

*Proof.* Since all FrQDFT eigenvalues  $\gamma_n$  are fourth roots of unity in the  $1-\mu$  plane (i. e.,  $\gamma_n = \pm 1, \pm \mu$  and, therefore, it has unit module), they can be written as  $\gamma_n = \exp(\mu\theta)$  (in which  $\theta = 0, \pm\frac{\pi}{2}, \pi$ ). Hence

$$\overline{\gamma_n^a} = \gamma_n = \exp(-a\mu\theta) = \gamma_n^{-a}, \quad (5.23)$$

consequently

$$(\mathbf{F}^a)^H = \mathbf{F}^{-a}. \quad (5.24)$$

From (5.24) and (5.18),

$$(\mathbf{F}^a)^H = \left( (\mathbf{E}\Gamma^a)\mathbf{E}^T \right)^H = \mathbf{E}(\mathbf{E}\Gamma^a)^H = \mathbf{E}(\Gamma^a)^H\mathbf{E}^T = \mathbf{E}\Gamma^{-a}\mathbf{E}^T = \mathbf{F}^{-a}, \quad (5.25)$$

and, following the index additivity and the reduction to identity properties,

$$\mathbf{F}^a(\mathbf{F}^a)^H = \mathbf{F}^a\mathbf{F}^{-a} = \mathbf{F}^0 = \mathbb{I}. \quad (5.26)$$

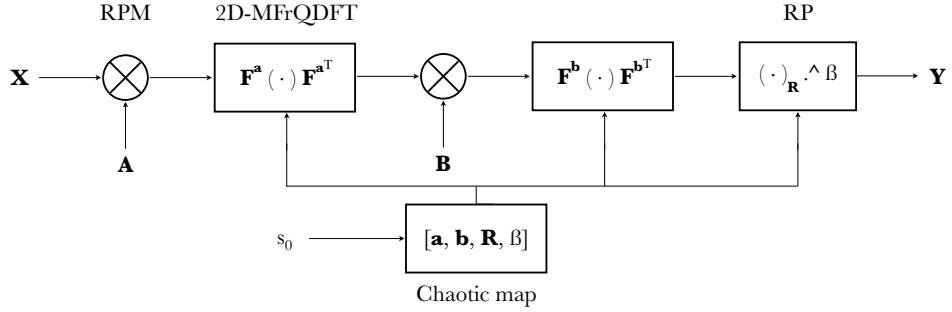
□

Before proceeding, it matters to notice that other approaches have been used to define fractional transforms especially suited for image encryption. Lima *et al.* (Lima et al. 2018) applied the generating matrix method to obtain eigenvectors — in a fashion similar to this work — and define multiorder reality-preserving discrete fractional transforms. Roopkumar (Roopkumar 2016), on the other hand, defined a continuous one-dimensional quaternion fractional Fourier transform by using a reasoning similar to the symplectic decomposition of quaternions: adding together two traditional fractional Fourier operators with certain imaginary unit, with one of them multiplied by an orthogonal pure quaternion. None of the approaches, to the best of the author’s knowledge, made use of eigenstructure analysis to define the FrQDFT and prove some of its properties.

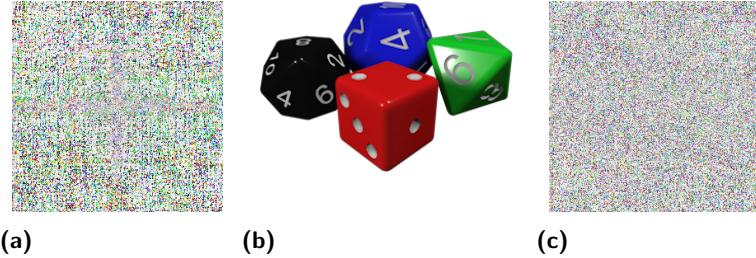
## 5.3 The multiparametric FrQDFT with application to color image encryption

Frequently, fractional transforms are employed in both grayscale (Tao et al. 2010) and color image (Kang et al. 2018; Kang and Tao 2018) encryption. By setting the secret key to be the transform fractional order, alongside the use of multiple encryption or multiparametric transforms, one is able to create ciphers with sufficiently large key spaces and highly sensible to small key changes. This section presents an illustrative application of the FrQDFT, creating a holistic encryption scheme for PNG images based on the proposition of a multiparametric FrQDFT.

The definition of the multiparametric FrQDFT, referred to as MFrQDFT, consists of employing a different fractional order for each eigenvalue in  $\Gamma$ . The



**Fig. 5.3.:** Proposed encryption scheme, exploring the 2D-MFrQDFT.



**Fig. 5.4.:** (a) Encrypted image. (b) Image decrypted with the correct key  $s_0$ . (c) Image decrypted with the wrong key  $\tilde{s}_0 = s_0 + \epsilon$ , with  $\epsilon = -1,6 \cdot 10^{-80}$ .

vector of fractional orders is represented by  $\mathbf{a} = [a_0, a_1, \dots, a_{N-1}]$ . The MFrQDFT of a column vector  $\mathbf{v}$  is

$$\text{MFrQDFT}\{\mathbf{v}\} = \mathbf{E}\Gamma^{\mathbf{a}}\mathbf{E}^T\mathbf{v} = \mathbf{F}^{\mathbf{a}}\mathbf{v}. \quad (5.27)$$

The symbols  $\Gamma^{\mathbf{a}}$  and  $\mathbf{F}^{\mathbf{a}}$  in (5.27) are abuses of notation. One must comprehend  $\Gamma^{\mathbf{a}}$  as the diagonal matrix obtained after raising the  $n$ -th entry in the diagonal of  $\Gamma$  to the  $n$ -th component in  $\mathbf{a}$ . On the other hand,  $\mathbf{F}^{\mathbf{a}}$  indicates the matrix  $\mathbf{E}\Gamma^{\mathbf{a}}\mathbf{E}^T$ . As it was done in Subsection 5.1.1, the 2D-MFrQDFT of a quaternion matrix  $\mathbf{X}$ , e. g. representing a PNG matrix, is written as

$$\text{2D-MFrQDFT}\{\mathbf{X}\} \stackrel{\Delta}{=} \widehat{\mathbf{X}} = \mathbf{F}^{\mathbf{a}}\mathbf{X}\mathbf{F}^{\mathbf{a}T}, \quad (5.28)$$

with inverse transform obtained from the properties listed in Section 5.2

$$\text{2D-MFrQDFT}^{-1}\{\widehat{\mathbf{X}}\} = (\mathbf{F}^{\mathbf{a}H})\widehat{\mathbf{X}}(\mathbf{F}^{\mathbf{a}}). \quad (5.29)$$

### 5.3.1 Encryption scheme using MFrQDFT

The chosen implementation for the PNG image encryption algorithm mixed a block of random-phase modulation (RPM, also called phase mask in (Chen et al. 2018a) and (Singh and Sinha 2008)) and multiparametric transform. As suggested by (Hsue 2018), a non-linear step of random power is also used. Fig. 5.3 depicts the system building blocks.

The plaintext image is initially converted into a quaternion matrix  $\mathbf{X}$ , which is input to a 2-round processing with RPM + 2D-MFrQDFT. The random-phase modulation consists of element-wise multiplication by a matrix of random unit quaternions. This role is fulfilled by matrices  $\mathbf{A}$  and  $\mathbf{B}$  in Fig. 5.3. The *inverse* RPM (in decryption) is achieved using the *conjugate* of the matrix used during encryption.

The random-power block (RP) is the final step in the encryption method. It creates non-linearity by raising randomly selected entries of the matrix  $\mathbf{X}$  to a random parameter  $\beta \in ]0, 1[$ . This selection is performed by a binary matrix  $\mathbf{R}$ , so that the output of the RP block to an input quaternions matrix  $\mathbf{M}$  is

$$\text{RP}(\mathbf{M})_{i,j} = \begin{cases} (\mathbf{M}_{i,j})^\beta & \text{if } \mathbf{R}_{i,j} = 1, \\ \mathbf{M}_{i,j} & \text{if } \mathbf{R}_{i,j} = 0. \end{cases} \quad (5.30)$$

During decryption, one must use the same matrix  $\mathbf{R}$  and an exponent  $\beta^{-1}$ .

A chaotic map was used to generate a pseudorandom sequence  $\mathbf{s}$  of numbers between 0 and 1, from which all of the encryption parameters were drawn: the fractional order vectors  $\mathbf{a}$  and  $\mathbf{b}$  of the two 2D-MFrQDFTs, the parameter  $\beta$  and the matrix  $\mathbf{R}$ <sup>1</sup> for the RP block. The random-phase modulation matrices were produced beforehand and could be left public in the scheme documentation. The encryption of a  $256 \times 256$ -pixels image, therefore, requires a sequence of length  $256 + 256 + 256^2 + 1$ . The secret key consists of the seed  $s_0$  of the chaotic map, a floating point variable between 0 and 1. As a consequence, the key space is determined by the smallest deviation  $\epsilon$  from  $s_0$  so that, using the wrong key  $\tilde{s}_0 = s_0 \pm \epsilon$  in decryption, it still leads to a noisy image without any detectable trace of original information.

The key space dimension, denoted by  $[K]$ , is the ratio between the range of all possible keys and the range of wrong keys *which still lead to partial image*

---

<sup>1</sup>For each entry  $\mathbf{R}_{i,j}$ , a corresponding element  $s_k$  of  $\mathbf{s}$  was taken and  $\mathbf{R}_{i,j}$  was set to 1 if and only if  $s_k > 0,5$ ,  $\mathbf{R}_{i,j} = 0$  otherwise.

*reconstruction*. The latter is  $[s_0 - \epsilon, s_0 + \epsilon]$ , following the previous definition of  $\epsilon$ . Since  $s_0 \in ]0, 1[$ ,

$$[K] = \frac{1 - 0}{s_0 + \epsilon - (s_0 - \epsilon)} = \frac{1}{2\epsilon}. \quad (5.31)$$

When testing the encryption scheme, the *tent map* (Singh and Sinha 2008) was chosen as tool for generating the  $s$  sequence; it is recursively defined as

$$s_{n+1} = \begin{cases} \gamma s_n & \text{if } 0 \leq s_n < 0.5, \\ \gamma(1 - s_n) & \text{if } 0.5 \leq s_n \leq 1. \end{cases} \quad (5.32)$$

with  $0 < \gamma \leq 2$ .

The tent map parameters were set to  $s_0 = 0.3$  and  $\gamma = 1.8$ , for no particular reason other than obeying the range of values for chaotic behaviour. The proposed encryption algorithm was used on the image in Fig. 5.2a, what yielded the ciphered output in Fig. 5.4a.

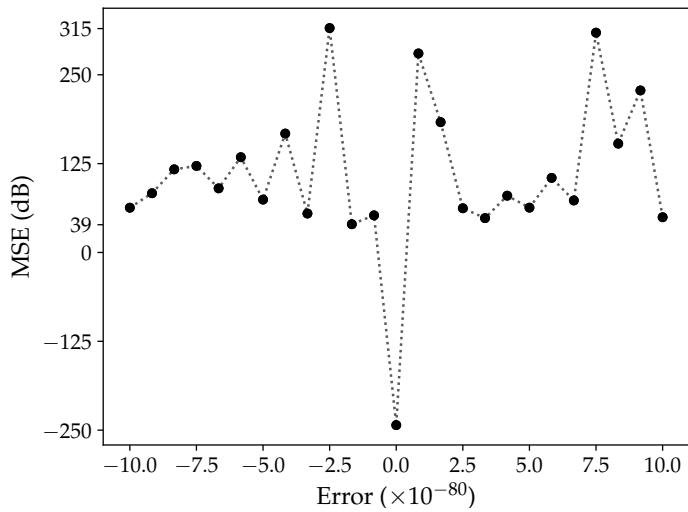
In order to evaluate the key space dimension, the ciphered image was decrypted using keys  $\tilde{s}_0 = s_0 \pm \epsilon$  for different values of  $\epsilon \in [-10 \times 10^{-80}; 10 \times 10^{-80}]$ , with the aid of multiple precision tools provided by the *RealField* class in the SageMath software. By the end of each decryption, it was computed the MSE between the supposedly recovered image and the original one, what is plotted in Fig. 5.5. The graph does not change smoothly because each tweak in  $s_0$  propagates along the whole sequence  $s$ , what affects randomly all the decryption parameters. This is, of course, consequence of the nature of a chaotic map. As shown in the graph, the smallest MSE (still using wrong keys, with  $\epsilon \neq 0$ ) was 39dB, obtained with  $\epsilon = -1,6 \times 10^{-80}$ .

Fig. 5.4b and 5.4c show the decrypted images using  $\epsilon = 0$  and  $\epsilon = -1,6 \times 10^{-80}$ , respectively. It can be seen that the wrong key which caused the smallest MSE still produced a seemingly random PNG image, a desirable property for an encryption scheme. Since the smallest deviation used in this test was  $0,83 \times 10^{-80}$ , it follows that the key space dimension is, at least, equal to

$$[K] = \frac{1}{2 \times 0,83 \times 10^{-80}} \approx 6,0 \times 10^{79}, \quad (5.33)$$

what means the key length must be

$$\lceil 79 \log_2 10 \rceil = 263 \text{ bits}, \quad (5.34)$$



**Fig. 5.5.:** Mean squared error, in dB, between the original and the decrypted images as function of the key error  $\epsilon$ .

a value greater than 256 bits, assumed to be appropriate for symmetric encryption schemes, by information security reports such as ECRYPT (Smart et al. 2018). It is also clear from these computations and Fig. 5.4c how sensitive the scheme is to small key changes: the smallest modification within the precision of  $10^{-80}$  in the seed  $s_0$  was enough to provide a completely noisy decrypted image (cf. Fig. 5.4c). The high key sensitivity is also indicated by the sharp dip in the MSE  $\times$  Error graph in Fig. 5.5.

Such an encryption scheme would not only be safe against brute-force attacks, given its key space dimension, but also known-plaintext attack. The reason for this is the non-linearity provided by the random-power block, as argued intensely by Hsue (Hsue 2018). Resistance to chosen-plaintext attacks, however, is not guaranteed, although simple adjustments could be done in future works to address this problem: the vector of fractional orders  $\mathbf{a} = \{a_0, a_1, \dots, a_{N-1}\}$  could be made dependent on both the plain image and external parameters (such as the secret seed  $s_0$ ). Such tweak in the algorithm would block the extraction of information from comparing chosen plain images with their encrypted counterparts (Chai et al. 2019; Hu et al. 2017; Murugan and Gounder 2016), making the scheme resistant to chosen-plaintext and chosen-ciphertext attacks (Wang et al. 2012); with resistance also to the weaker attacks of brute-force and known-plaintext.

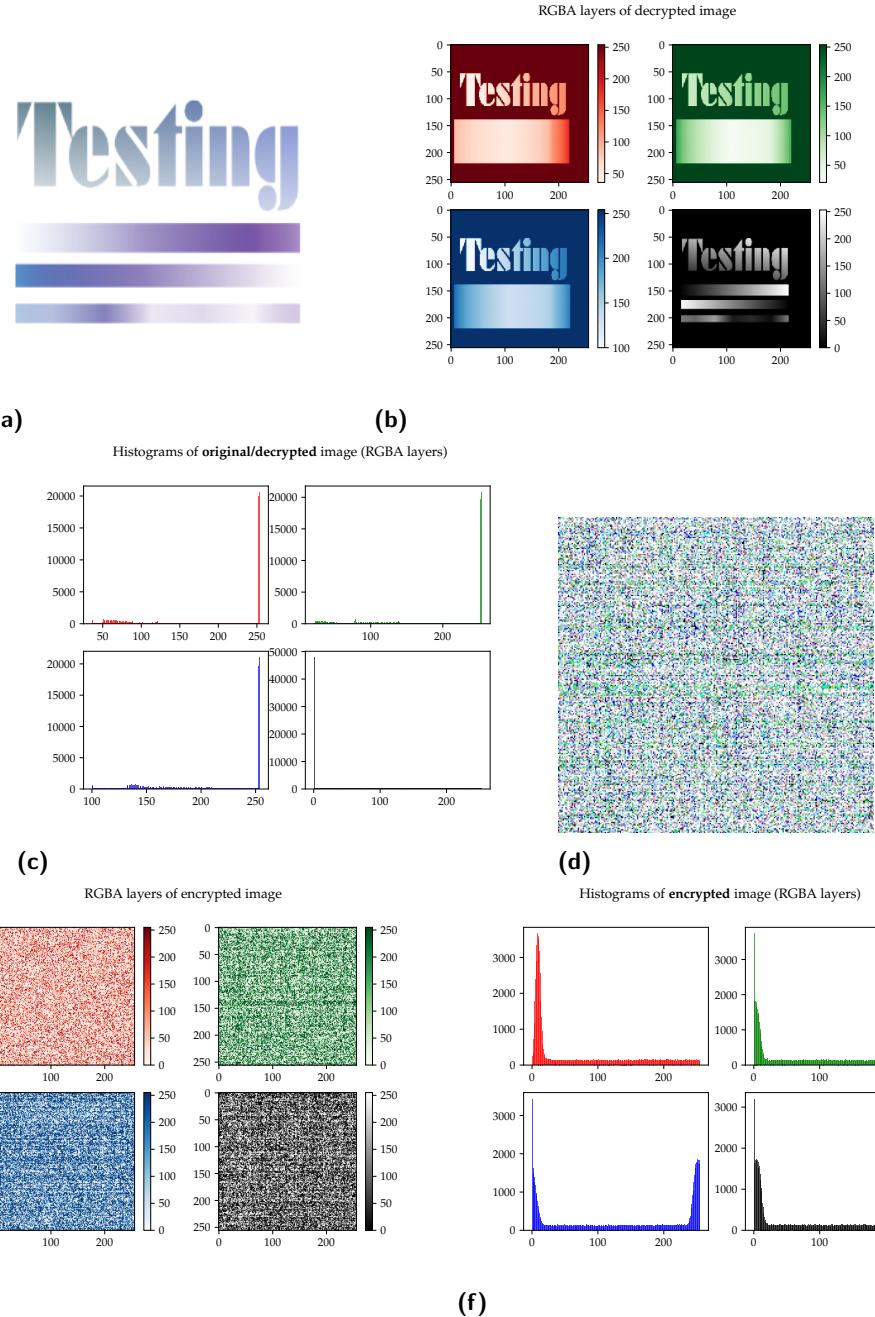
Alongside the investigation of key space dimension and sensitivity, histogram analysis play an important role in describing the performance of encryption schemes.

A practical cipher should present an output symbol distribution ideally independent from the cleartext, so that no information is leaked through histogram visualization (Zhang and Wang 2014). In the case of a PNG image, four histograms are drawn, one for each layer. Fig. 5.6 depicts a test PNG image split into its color and opacity components and the corresponding histograms, both prior and after encryption. A group of 13 256 × 256-pixels PNG images were encrypted using the proposed method and the histograms of the ciphered output were collected, what is shown in Fig. 5.8. No matter how diverse the input images may be, the histograms present continuously the same profile, what indicates decoupling of information between plain and ciphered images. Although a deeper study on the properties and safety standards of the proposed encryption scheme is needed for full description of its capacity, the analysis presented fit in the scope of this work and fulfill the goal of presenting the MFrQDFT with a clear and illustrative application.

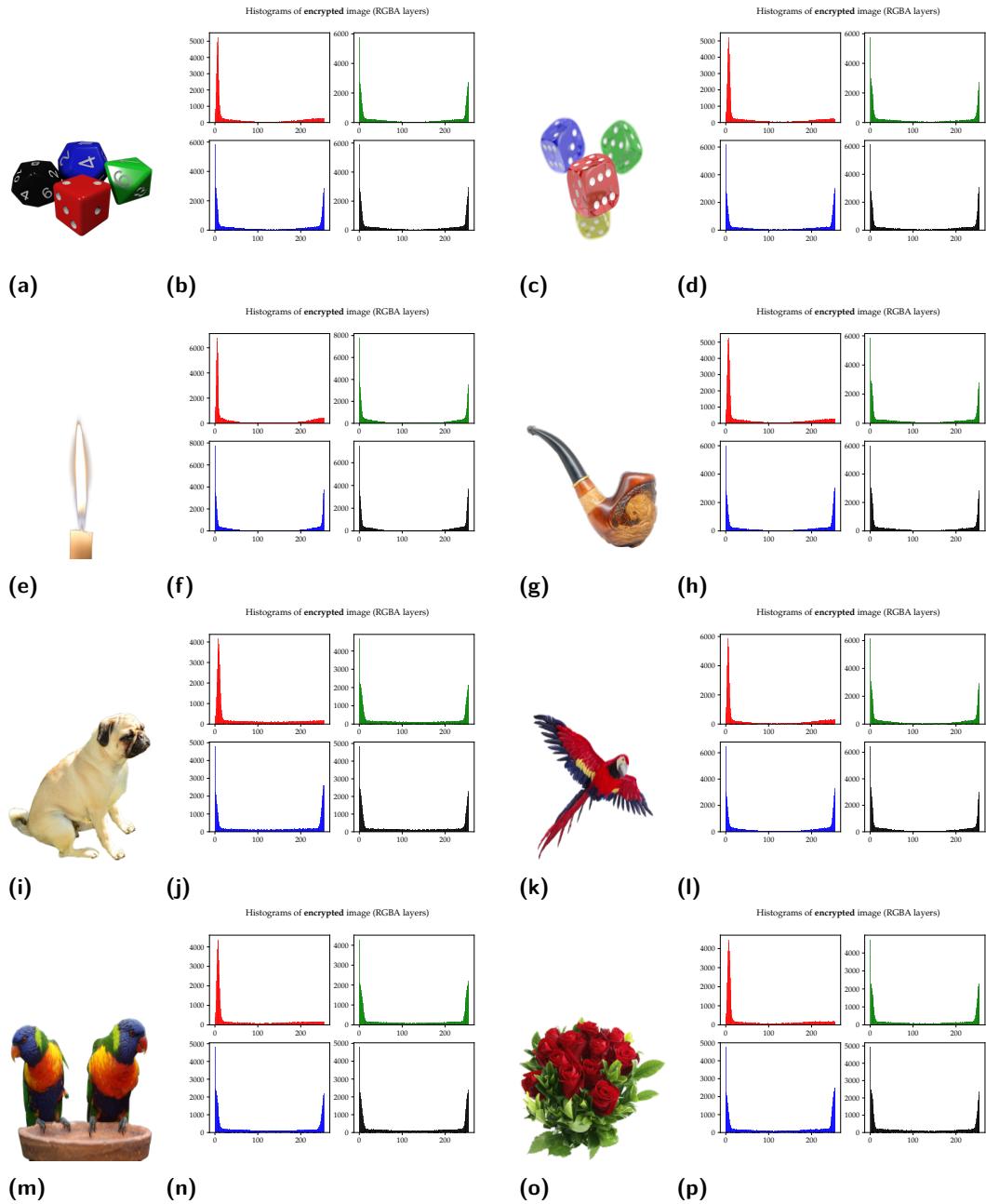
Finally, as stated at the beginning of the chapter, there are a multitude of studies on image encryption, each addressing particular aspects and problems. Therefore, it is adequate to acknowledge the position of this proposed encryption algorithm in the literature landscape. Without providing an extensive and complete analysis, for lack of space, some remarks can be done. The key space dimension in related works vary from circa  $10^{54}$ , around 160 bits, in a work by Wang *et al.* with flexible key space (Wang *et al.* 2015b), to more than 400 bits, in a paper by Zhang *et al.* (Zhang and Wang 2015), an interval which includes the 260 bits of the proposed algorithm. Regarding time of encryption, Wang *et al.* (Wang *et al.* 2015c) used Arnold cat map and dynamic random growth to obtain large running speed of the encryption and decryption algorithm. This is an aspect our proposed scheme is not optimized for, although great improvement in processing time can be achieved if the eigenvector and eigenvalue matrices of the QDFT are stored in-memory, taking the eigendecomposition out of the encryption process. As a final comment, the main advantages of the proposed scheme are its modularization, the holistic processing of color images with opacity layer, the lack of long iterations and the ease to describe, comprehend and implement.

## 5.4 Concluding remarks

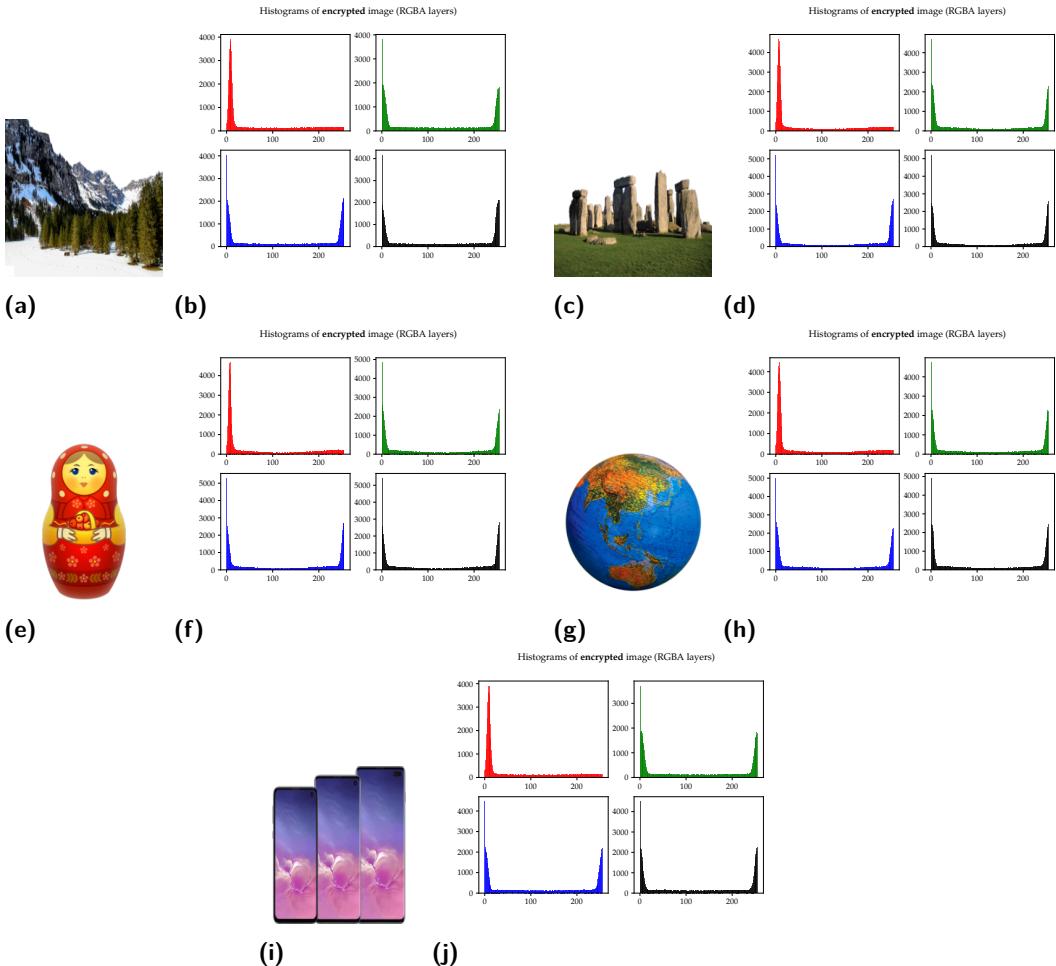
This chapter investigated the eigenstructure of the quaternion discrete Fourier transform. Although quaternion matrix decomposition is a challenging topic, the problem for the QDFT was solved by proving that this transform and the DFT share symmetric eigenvectors, what allowed for the construction of an



**Fig. 5.6.:** (a) Test image in PNG format, with  $256 \times 256$  pixels. (b) Color and opacity layers of the *decrypted* image (which coincides with the original in (a)). (c) Histograms of each layer of the *decrypted* image. (d) Encrypted image, in PNG format. (e) Color and opacity layers of the *encrypted* image. (f) Histograms of each layer of the *encrypted* image.



(Part of Fig. 5.8, it continues on the next page).



**Fig. 5.8.:** Set of 13 PNG test images,  $256 \times 256$  pixels, alongside the histograms of each layer (colors and opacity) of their encrypted version. Images downloaded from the online free database <https://purepng.com/>, under CC0 license.

orthogonal eigenbasis of the QDFT, using Hermite–Gaussian-like DFT eigenvectors (Oliveira Neto and Lima 2017). This result led to the definition of a fractional QDFT, which was proven to hold properties similar to those of the FrDFT, its complex-valued counterpart.

The FrQDFT was further generalized by introducing the MFrQDFT, a multi-parametric version. Exploring the 4D nature of quaternions, a holistic encryption scheme for color images with opacity layer was proposed, as an illustrative application of the 2D-MFrQDFT, and shown to provide satisfactorily large key space and key sensitivity, resistance to known-plaintext attack and ease of description and implementation. Future works could possibly expand this analysis and address whether some hypercomplex image moments, such as ternary radial harmonic Fourier moments and quaternion polar harmonic (Wang et al. 2019a; Wang et al. 2018), could be used for image encryption, eventually in specific scenarios and applications.



# Quaternion graph signal processing

Graph signal processing is a highly flexible tool for analysing and transforming graph-based data. However, as it happens with so many engineering tasks, often it appears to mix mathematics and art: the experienced GSP user knows that quite a few characteristics of the underlying graph are open to be determined by the very problem at hand and by the user's intuition. That is, whether one must consider real or complex edge weights, directed or undirected graphs, whether loops or multiple edges shall be allowed, all of these modeling choices arise from the data and from the way one considers the most appropriate to represent the relations within. We make hereby the argument that changing the algebra over which the graph signal samples (and edge weights) are defined, although rarely regarded among modeling choices, may reveal powerful new tools.

Let us take an example of how the signal algebra may affect the processing possibilities, in the current state of GSP. Undirected graphs with real-valued weights are a common option for those used to model data defined over geographic spaces, since it is reasonable to consider the adjacency relations to be non-directional and weighted by positive real numbers (Shuman et al. 2013). However, the same topology would be arguably inappropriate should one deal with complex-valued signals: a directed graph or a graph with complex-valued edge weights would be more reasonable options to evaluate, since they would produce graph Fourier transform matrices with complex entries. Although this scenario may seem to lack practical purpose, defining a graph signal with samples lying in extensions of the real field (e.g. complex numbers) would be justified by the increase in the amount of information stored within each sample (from a single real number, to twice as much in the complex case).

Electrical engineers for many decades have exploited the benefits of handling more than one real-valued information (e.g. magnitude and phase) encoded in a single signal sample, and a similar motivation led to Sangwine's (Sangwine 1996) discrete version of a family of bidimensional transforms over the *quaternions* (a skew-field which extends the complex numbers by having four real valued components). As the reader may recall from Chapter 1, these transforms used this class of

hypercomplex numbers to perform holistic color image processing, handling all three color channels at once. In fact, *quaternion signal processing* was created to embed three- or four-dimensional data into one-dimensional signals (Took and Mandic 2008). Ever since, quaternion transforms have been employed not only on color image processing (Ell and Sangwine 2007; Chen et al. 2018b; Li 2013; Evans et al. 2000), but also on other tasks such as bivariate signal analysis (Flamant et al. 2017b; Flamant et al. 2017a; Flamant et al. 2018).

In Chapter 3 it was presented the motivation and fundamentals of graph signal processing, while Chapter 4 delved into the exploration of a fractional graph shift. This chapter attempts to further extend the borders of GSP by pondering the problem of signal processing on graphs with quaternion-valued edge weights, referred to as *quaternion graph signal processing*.

## 6.1 Laying the foundations for QGSP

Since QGSP is the development and application of GSP tools to the context of quaternion-valued signals and graph edge weights, it starts by extending the usual definition of graph signal in (3.17), replacing the complex-valued numbers with quaternions. As such, the quaternion graph signal  $s$ , defined over the graph  $\mathcal{G} = \{\mathcal{V}, \mathbf{A}\} \mid \mathbf{A} \in \mathbb{H}^{n \times n}$ , is defined as

$$s : \mathcal{V} \rightarrow \mathbb{H} \mid s(v_i) = s_i. \quad (6.1)$$

This definition unfolds some challenges: what does it mean to have a smooth quaternion graph signal? Does the well known benefit of quaternion signal processing, namely holistic manipulation of multiple channels of information, transfer to graph signal processing? How is one able to design LSI filters in QGSP?

In order to properly address these questions and establish the foundations of QGSP, a few milestones were set to be conquered. Firstly, an algorithm to compute the direct and inverse (quaternion) graph Fourier transforms should be proposed, to allow for basic spectral analysis. This includes revisiting the definition of frequency ordering, which is well understood only for real and complex graph signals. Secondly, QGSP should ideally provide a class of graphs for which the Fourier transform is known to exist (and hopefully easier to compute than outside this class). The reasoning here is to find a case similar to that of usual GSP, where it is known that undirected graphs with real-valued edge weights always have diagonalizable GSOs.

Finally, QGSP should ideally provide a method to design graph FIR filters, tailored to approximate a given frequency response. The Subsection 6.1.1 gives the first step towards QGSP, properly defining the QGFT and studying the eigendecomposition of the quaternion graph shift operator (QGSO).

### 6.1.1 Eigendecomposing the shift operator

At the very center of vertex-frequency analysis in GSP lies the definition that a basis of eigenvectors of the chosen graph shift operator act as Fourier components for the space of graph signals. This can be directly translated to QGSP, after taking into account the results presented in Chapter 2, regarding diagonalization of quaternion matrices.

**Definition 4.** *Given the graph  $\mathcal{G} = \{\mathcal{V}, \mathbf{A}\}$ , with adjacency matrix  $\mathbf{A} \in \mathbb{H}^{n \times n}$ , the quaternion graph Fourier transform (QGFT) is the projection of a graph signal onto the eigenspace of  $\mathbf{A}$ .*

The first step towards the QGFT is, therefore, the generation of a basis of (possibly generalized) eigenvectors of the adjacency matrix. This work has dealt only with the case of diagonalizable graph matrices, but the use of the Jordan canonical form can be investigated for non-diagonalizable QGSOs (Longxuan et al. 1996).

On the one hand, from the Corollary 1 it is known that the diagonalizability of an adjacency matrix  $\mathbf{A}$  both implies and requires that of its complex adjoint matrix  $\chi_A$ . Let us write  $\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^{-1}$ . On the other hand, Theorem 3 guarantees that the eigenvalues of the adjacency matrix can be taken as half the ones of its complex adjoint matrix. More specifically, they can be taken as the union of the set of eigenvalues with positive imaginary part (recall that they are complex-valued) and the set of *distinct* eigenvalues with null imaginary part (recall that every real-valued eigenvalue appears twice).

However, once the desired eigenvalues from  $\chi_A$  have been defined, how does one assemble  $\mathbf{V}$  out of the eigenvectors of  $\chi_A$ ? That is answered by Equation (2.44), which is rewritten below for convenience:

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ -\overline{\mathbf{A}}_2 & \overline{\mathbf{A}}_1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1 \\ -\overline{\mathbf{v}}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{v}_1 \\ -\overline{\mathbf{v}}_2 \end{pmatrix} \lambda.$$

On the left-hand side of the equation, we find  $\chi_A$  and one of its eigenvectors, which is written in terms of the symplectic decomposition of a quaternionic column vector

$\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2 j$ . As it turns out,  $\mathbf{v}$  is precisely the eigenvector of  $\mathbf{A}$  associated with the eigenvalue  $\lambda$ . This is proven in Appendix A and constitutes a minor contribution of this thesis. The answer to the question raised a few lines above, to sum up, is that once one has the chosen eigenvalues of  $\chi_A$ , it is possible to assemble  $\mathbf{V} \in \mathbb{H}^{n \times n}$  out of the respective eigenvectors of  $\chi_A$  by simply splitting these  $2n$ -long eigenvectors into halves: the first half corresponds to  $\mathbf{v}_1$  (the simplex part of the eigenvector  $\mathbf{v}$ ), while the second half is  $-\bar{\mathbf{v}}_2$  (with  $\mathbf{v}_2$  being the perplex part of  $\mathbf{v}$ ).

So far we are able to obtain  $\mathbf{V}$  from  $\mathbf{A}$ , bypassing the direct eigendecomposition of a quaternion matrix by using its complex adjoint. However, the question of how to get  $\mathbf{V}^{-1}$ , if it exists, poses itself immediately, after all  $\mathbf{V}$  corresponds to the *inverse* of the QGFT. Let us address this point.

### 6.1.2 Inversion of the eigenvector matrix and frequency ordering

The inversion of quaternion matrices is certainly a problem tackled by many researchers, but is not featured prominently in literature, at least not as far as the author is aware. There is, however, an algorithm proposed by (Cohen and De Leo 1999) during studies on the determinant of quaternion matrices, when they highlighted the utility of Schur complements on computing the inversion of any matrix  $\mathbf{M} \in \mathcal{R}^{n \times n}$  over the ring  $\mathcal{R}$ . Let us go quickly through their idea. Given that  $\mathbf{M}$  is written in block form,

$$\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}, \quad (6.2)$$

and under the requisite that  $\mathbf{A} \in \mathcal{R}^{k \times k}$  is invertible, the Schur complement of  $\mathbf{A}$  in  $\mathbf{M}$  is defined as

$$\mathbf{A}_s \stackrel{\Delta}{=} \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}. \quad (6.3)$$

Given that  $\mathbf{A}_s$  is also invertible, the closed formula for the inverse of  $\mathbf{M}$  is

$$\mathbf{M}^{-1} = \begin{pmatrix} \mathbf{I}_k & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I}_{n-k} \end{pmatrix} \begin{pmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_s^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{I}_k & \mathbf{0} \\ -\mathbf{C}\mathbf{A}^{-1} & \mathbf{I}_{n-k} \end{pmatrix}, \quad (6.4)$$

so that the inversion of  $\mathbf{M}$  is now reduced to inversion of the two smaller matrices  $\mathbf{A}$  and  $\mathbf{A}_s$ .

Although this method proposes a closed formula for quaternion matrix inversion, it requires an exhaustive search for a submatrix in the upper left corner that, simultaneously, is *invertible* and *has invertible Schur complement*. Probably the idea can be explored in the future and used to find an efficient way to compute  $\mathbf{V}^{-1}$ ,

but in Algorithm 1 we propose a more reasonable compromise between processing speed, implementation time and broad applicability.

The reasoning goes as follows. According to Theorem 2, a necessary and sufficient condition for the invertibility of a matrix  $\mathbf{V} \in \mathbb{H}^{n \times n}$  is the existence of  $\chi_V^{-1}$ . Moreover, from Theorem 1, if  $\chi_V^{-1}$  exists and has the form of a complex adjoint matrix, let us say  $\chi_V^{-1} = \chi_M$ , then it follows that  $\mathbf{M} = \mathbf{V}^{-1}$ , since the theorem guarantees that

$$\chi_V \chi_M = \chi_V \chi_V^{-1} = \mathbf{I}_{2n \times 2n} = \chi_{\mathbf{I}_{n \times n}} \Rightarrow \mathbf{V}\mathbf{M} = \mathbf{I}_{n \times n}, \quad (6.5)$$

letting  $\mathbf{I}_{m \times m}$  be the identity matrix of order  $m^1$ . That is, once the complex adjoint of  $\mathbf{V}$  is computed, one simply needs to compute its inverse, if it exists, and verify if it follows the format in Definition 1.

**Algorithm 1.** *Compute the inverse of a quaternion-valued matrix, if a sufficient condition for its existence is met:*

**Input:**  $\mathbf{V} \in \mathbb{H}^{n \times n}$ . **Output:**  $\mathbf{V}^{-1}$  or None.

```

1:  $\chi_V \leftarrow \text{to\_complex\_adjoint}(\mathbf{V})$ 
2: if  $\det(\chi_V) = 0$  then
3:   return None
4: else
5:    $\mathbf{U} \leftarrow \text{inverse}(\chi_V)$ 
6:   if not  $\text{has\_complex\_adjoint\_form}(\mathbf{U})$  then
7:     return None
8:   else
9:      $\mathbf{V}^{-1} \leftarrow \text{from\_complex\_adjoint}(\mathbf{U})$ 
10:    return  $\mathbf{V}^{-1}$ 
11:   end if
12: end if
```

These functions were used in the algorithm to improve readability:

- `to_complex_adjoint`: converts a quaternion matrix to its complex adjoint form.
- `from_complex_adjoint`: converts a complex adjoint matrix to its quaternion-valued form.
- `inverse`: computes the inverse of a complex-valued matrix.
- `has_complex_adjoint_form`: checks if the matrix has the complex adjoint form, i. e., is a block matrix as the one in Definition 1.

---

<sup>1</sup>For simplicity, this notation is slightly loose, representing both a complex- and a quaternion-valued identity matrix, since in both cases their entries have zero-valued or zero-normed imaginary parts.

At this point, all steps required to generate the QGFT, assuming the graph adjacency matrix is diagonalizable, have been addressed. However, the transform still remains of little use until a clear definition of frequency ordering is presented, otherwise the lack of sense of low and high frequencies make the signal spectrum meaningless.

Since the quaternions form a normed algebra — and the reader may refer back to (2.16) — it is natural to borrow from classical GSP the definition of the graph total variation as a *metric* for frequency. In fact, even the elegant property of frequency ordering in the complex plane follows from GSP to QGSP. Let us see why.

Let  $\mathbf{A} \in \mathbb{H}^{n \times n}$  be diagonalizable, with standard eigenvalues ordered like so

$$|\lambda_0| \leq |\lambda_1| \leq \cdots \leq |\lambda_{N-1}| \stackrel{\Delta}{=} |\lambda_{max}|, \quad (6.6)$$

associated with the eigenvectors  $(\mathbf{v}_i)_{i=0,\dots,n-1}$ . Now, let us notice that the graph total variation, defined in (3.33), does not have to use  $\ell_1$ -norm for it to quantify the notion of frequency. In fact, let us use the general  $\ell_p$ -norm for a moment, with  $p \geq 1$ , represented as  $\|\mathbf{v}\|_p \stackrel{\Delta}{=} \left( \sum_{k=0}^{n-1} |v_k|^p \right)^{1/p}$ , for  $\mathbf{v} \in \mathbb{H}^n$ , and define the graph total variation of the graph signal  $\mathbf{s}$  as

$$TV_{G,p}(\mathbf{s}) \stackrel{\Delta}{=} \left\| \mathbf{s} - \frac{1}{|\lambda_{max}|} \mathbf{A} \mathbf{s} \right\|_p. \quad (6.7)$$

If we take the important step of scaling the eigenvectors  $\mathbf{v}_i$  so they have unit  $\ell_p$ -norm, i. e., making  $\|\mathbf{v}_i\|_p = 1$ , then the associative and distributive properties of quaternion multiplication allow us to do

$$\begin{aligned} TV_{G,p}(\mathbf{v}_k) &= \left\| \mathbf{v}_k - \frac{1}{|\lambda_{max}|} \mathbf{A} \mathbf{v}_k \right\|_p = \left\| \mathbf{v}_k - \mathbf{v}_k \frac{1}{|\lambda_{max}|} \lambda_k \right\|_p \\ &= \left\| \mathbf{v}_k \left( 1 - \frac{1}{|\lambda_{max}|} \lambda_k \right) \right\|_p \\ &= \underbrace{\|\mathbf{v}_k\|_p}_{=1} \left| 1 - \frac{\lambda_k}{|\lambda_{max}|} \right| = \left| \lambda_k - |\lambda_{max}| \right| \frac{1}{|\lambda_{max}|}. \end{aligned} \quad (6.8)$$

What leads to exactly the same frequency ordering obtained in the classical GSP derivation,

$$\left| \lambda_i - |\lambda_{max}| \right| \leq \left| \lambda_j - |\lambda_{max}| \right| \iff TV_{G,p}(\mathbf{v}_i) \leq TV_{G,p}(\mathbf{v}_j). \quad (6.9)$$

In other words, once the eigenvectors are individually *normalized*, i. e., scaled to have unit  $\ell_p$ -norm, then the farther an eigenvalue lies from the point  $|\lambda_{max}|$  in the real line of the complex plane, the greater is its eigenvector graph total variation  $TV_{G,p}$ . By definition, it means the greater is the frequency it represents. Besides, notice how the scaling implies that the value of  $TV_{G,p}$  depends solely on the eigenvalues, not on  $p$ .

In order to interpret the spectrum domain, the eigenvalues are sorted in ascending order of their respective eigenvector total variation: this sorts the frequency from lowest to highest. When not explicitly mentioned, it is assumed a value of  $p = 1$  for all occurrences of  $\ell_p$ -norm.

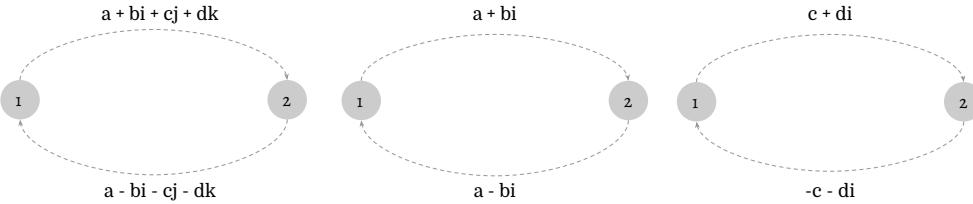
### 6.1.3 On the existence of a class of graphs with diagonalizable adjacency matrix

A relevant topic of discussion, when laying the foundations for QGSP, is to ponder the question on the existence of a class of graphs with diagonalizable QGSOs. It is a reasonable point of investigation, since in classical GSP one can confidently take advantage of some classes of graphs to make sure the shift operator is diagonalizable beforehand, which saves time and effort. Some examples are undirected real-weighted graphs and directed ring graphs with equal edge weights.

The reader may notice that undirected quaternion graphs do not possess necessarily a diagonalizable adjacency matrix. Indeed, the fact that  $\mathbf{A}$  is symmetric has no effect on proving the diagonalizability of its complex adjoint,

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ -\overline{\mathbf{A}}_2 & \overline{\mathbf{A}}_1 \end{pmatrix}. \quad (6.10)$$

The subjacent reason as to why undirected real-weighted graphs have diagonalizable adjacency matrices is that these real-valued matrices are symmetric, or more generally *normal*, i. e., they commute with (because are equal to) their transpose. In the case of complex-valued matrices, being normal (i. e., commuting with their Hermitian transpose) also suffices for proving their diagonalizability. So, in order to find quaternion graphs analogous to undirected real-weighted graphs, in the sense of having a sufficient condition for diagonalizable shift operators, their adjacency matrix must possess *normal* complex adjoint matrices. Fortunately, Theorem 1 states the condition:  $\chi_A$  is normal ( $\chi_A^H \chi_A = \chi_A \chi_A^H$ ) or Hermitian ( $\chi_A^H = \chi_A$ ) if and only if so is  $\mathbf{A}$ .



**Fig. 6.1.:** Illustration of two connected vertices in a graph with Hermitian adjacency matrix (left-hand side). In the center and on the right, respectively, are displayed the edges created by the simplex and perplex parts of the adjacency matrix.

Zhang claims Theorem 1 is proved by direct verification (Zhang 1997), but let us go through a way to visualize part of the proof within the context of GSP.

*Proof.* The part of the Theorem about to be verified is: *is it true that Hermitian quaternion-valued matrices have Hermitian complex adjoints?* Equivalently,

$$\mathbf{A}^H = \mathbf{A}, \quad \mathbf{A} \in \mathbb{H}^{n \times n} \Rightarrow \chi_A^H = \chi_A, \quad \chi_A \in \mathbb{C}^{2n \times 2n} \quad (6.11)$$

Look at  $\mathbf{A}$  as a graph adjacency matrix and let  $A_{i,j} = a + bi + cj + dk$  be the weight of the edge going from  $v_j$  to  $v_i$ . The hypothesis of a Hermitian quaternion-valued matrix implies  $A_{j,i} = \overline{A_{i,j}}$ , or  $A_{j,i} = a - bi - cj - dk$ . As an example, a pair of connected vertices in this graph is represented in Fig. 6.1, in which one can see the edges in opposite directions and with conjugate edge weights.

Fig. 6.1 depicts also the edges that would arise separately from the simplex and perplex parts of the edge weights (see the center and right diagrams). For instance, the edge weight  $A_{j,i} = a - bi - cj - dk$  satisfies<sup>2</sup>

$$A_{j,i} = A_{j,i}^{(s)} + A_{j,i}^{(p)} j, \quad \text{with} \quad \begin{cases} A_{j,i}^{(s)} = a - bi \\ A_{j,i}^{(p)} = -c - di \end{cases} . \quad (6.12)$$

This (along with Fig. 6.1) sheds light on the symplectic decomposition of  $\mathbf{A}$ : its symplex part satisfies  $\mathbf{A}_1^T = \overline{\mathbf{A}_1}$  (complex Hermitian matrix) while its perplex satisfies  $\mathbf{A}_2^T = -\mathbf{A}_2$ . With this in mind, the Hermitian of the complex adjoint yields

$$\chi_A^H = \begin{pmatrix} \mathbf{A}_1^H & -\mathbf{A}_2^T \\ \mathbf{A}_2^H & \mathbf{A}_1^T \end{pmatrix} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ -\overline{\mathbf{A}}_2 & \overline{\mathbf{A}}_1 \end{pmatrix}, \quad (6.13)$$

<sup>2</sup>Notice how the notation will mix once more in this thesis the superscripts <sup>(s)</sup> and <sup>(p)</sup> along with the subscripts 1 and 2 to indicate the symplectic decomposition parts.

precisely the matrix  $\chi_A$ , as intended to be proved.  $\square$

### 6.1.4 Does a Hermitian QGSO have unitary eigenvector matrix?

It is known that every Hermitian complex matrix is diagonalizable by a unitary matrix and has only real-valued eigenvalues. In other words, if  $\mathbf{M} \in \mathbb{C}^{n \times n}$  and  $\mathbf{M} = \mathbf{M}^H$ , then there exists a complex matrix  $\mathbf{U}$  and a diagonal real matrix  $\mathbf{\Gamma}$  so that  $\mathbf{M} = \mathbf{U}\mathbf{\Gamma}\mathbf{U}^{-1}$  and  $\mathbf{U}^{-1} = \mathbf{U}^H$ . However, despite the fact one can guarantee that a Hermitian QGSO has a Hermitian complex adjoint matrix, one could ponder the question whether this implies its eigenvector matrix is necessarily unitary. Let us verify this.

Let  $\mathbf{A}$  be a Hermitian (hence diagonalizable) quaternion adjacency matrix, therefore  $\mathbf{AV} = \mathbf{V}\Lambda$ . Since  $\chi_A$  is complex-valued and Hermitian, its eigenvector matrix (let us say,  $\Phi$ ) is unitary and we may write  $\chi_A = \Phi\mathbf{\Gamma}\Phi^H$ . Each column  $\phi$  in  $\Phi$ , the reader may recall, is written in terms of the simplex and perplex parts (which are complex-valued) of the respective column  $v$  in  $\mathbf{V}$ :

$$\phi^{(n)} = \begin{pmatrix} \mathbf{v}_1^{(n)} \\ -\bar{\mathbf{v}}_2^{(n)} \end{pmatrix}. \quad (6.14)$$

Since  $\Phi$  is unitary,

$$\begin{aligned} \phi^{(n)H}\phi^{(n)} &= 1 \\ \begin{pmatrix} \mathbf{v}_1^{(n)H} & -\mathbf{v}_2^{(n)T} \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^{(n)} \\ -\bar{\mathbf{v}}_2^{(n)} \end{pmatrix} &= 1 \\ \mathbf{v}_1^{(n)H}\mathbf{v}_1^{(n)} + \mathbf{v}_2^{(n)T}\bar{\mathbf{v}}_2^{(n)} &= 1. \end{aligned} \quad (6.15)$$

Now, does the final equality in (6.15) imply that  $\mathbf{v}^{(n)H}\mathbf{v}^{(n)} = 1$ ? If so, this would make the quaternion-valued eigenvector matrix  $\mathbf{V}$  unitary. Let us drop the superscript  $n$  temporarily, to clean the notation during the following lines. Since  $\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2\mathbf{j}$ ,

$$\begin{aligned} \mathbf{v}^H\mathbf{v} &= (\mathbf{v}_1^H - \mathbf{v}_2^T\mathbf{j})(\mathbf{v}_1 + \mathbf{v}_2\mathbf{j}) \\ &= \mathbf{v}_1^H\mathbf{v}_1 + \mathbf{v}_1^H\mathbf{v}_2\mathbf{j} - \underbrace{\mathbf{v}_2^T\mathbf{j}\mathbf{v}_1}_{\mathbf{v}_2^T\bar{\mathbf{v}}_1\mathbf{j}} - \underbrace{\mathbf{v}_2^T\mathbf{j}\mathbf{v}_2\mathbf{j}}_{-\mathbf{v}_2^T\bar{\mathbf{v}}_2} \\ &= \underbrace{\mathbf{v}_1^H\mathbf{v}_1 + \mathbf{v}_2^T\bar{\mathbf{v}}_2}_{=1} + (\mathbf{v}_1^H\mathbf{v}_2 - \mathbf{v}_2^T\bar{\mathbf{v}}_1)\mathbf{j} \\ &= 1 + (\mathbf{v}_1^H\mathbf{v}_2 - \mathbf{v}_2^T\bar{\mathbf{v}}_1)\mathbf{j}. \end{aligned} \quad (6.16)$$

Since the eigenvector  $\phi^{(n)} = (\mathbf{v}_1^{(n)} \quad -\bar{\mathbf{v}}_2^{(n)})^T$  is complex-valued, then it satisfies the additional constraint  $\mathbf{v}_1^{(n)H} \mathbf{v}_2^{(n)} - \mathbf{v}_2^{(n)T} \bar{\mathbf{v}}_1^{(n)} = 0$ , because the product  $\mathbf{v}_2^{(n)T} \bar{\mathbf{v}}_1^{(n)}$  commutes. Therefore, a Hermitian QGSO *does have* a unitary eigenvector matrix. As a practical consequence, the inverse eigenvector matrix can always be computed simply by taking its complex conjugate.

### 6.1.5 Numerical example: computing the QGFT matrix

Let the graph in Fig. 6.3 have the following adjacency matrix,

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 1 - 7i - 5j - 1k & 6 + 3i + 7j + 4k \\ 0 & 0 & 0 & 0 & 6 + 9i + 2j + 6k \\ 0 & 0 & 0 & 0 & 7 + 4i + 3j + 7k \\ 1 + 7i + 5j + 1k & 0 & 0 & 0 & 7 + 2i + 5j + 4k \\ 6 - 3i - 7j - 4k & 6 - 9i - 2j - 6k & 7 - 4i - 3j - 7k & 7 - 2i - 5j - 4k & 0 \end{pmatrix}.$$

As expected, since  $\mathbf{A}$  is Hermitian, so it is its complex adjoint:

$$\chi_A = \begin{pmatrix} 0 & 0 & 0 & 1 - 7i & 6 + 3i & 0 & 0 & 0 & -5 - 1i & 7 + 4i \\ 0 & 0 & 0 & 0 & 6 + 9i & 0 & 0 & 0 & 0 & 2 + 6i \\ 0 & 0 & 0 & 0 & 7 + 4i & 0 & 0 & 0 & 0 & 3 + 7i \\ 1 + 7i & 0 & 0 & 0 & 7 + 2i & 5 + 1i & 0 & 0 & 0 & 5 + 4i \\ 6 - 3i & 6 - 9i & 7 - 4i & 7 - 2i & 0 & -7 - 4i & -2 - 6i & -3 - 7i & -5 - 4i & 0 \\ 0 & 0 & 0 & 5 - 1i & -7 + 4i & 0 & 0 & 0 & 1 + 7i & 6 - 3i \\ 0 & 0 & 0 & 0 & -2 + 6i & 0 & 0 & 0 & 0 & 6 - 9i \\ 0 & 0 & 0 & 0 & -3 + 7i & 0 & 0 & 0 & 0 & 7 - 4i \\ -5 + 1i & 0 & 0 & 0 & -5 + 4i & 1 - 7i & 0 & 0 & 0 & 7 - 2i \\ 7 - 4i & 2 - 6i & 3 - 7i & 5 - 4i & 0 & 6 + 3i & 6 + 9i & 7 + 4i & 7 + 2i & 0 \end{pmatrix}.$$

Since the matrix  $\chi_A$  is complex-valued and Hermitian, it is diagonalizable by a unitary matrix  $\Phi$ . The eigenvalues of  $\chi_A$  are all real-valued and appear in pairs:

$$\text{diag}(\mathbf{\Gamma}) = \begin{pmatrix} -22.83570384 \\ -22.83570384 \\ -6.34920255 \\ -6.34920255 \\ 0 \\ 0 \\ 6.45799327 \\ 6.45799327 \\ 22.72691312 \\ 22.72691312 \end{pmatrix}.$$

The next step is to take half the eigenvalues of  $\chi_A$  (the ones with positive imaginary part, plus half the real-valued ones) to obtain the standard eigenvalues of  $\mathbf{A}$  and the respective eigenvectors. Since the real-valued eigenvalues appear in pairs, it is important to take one from each pair, to avoid picking eigenvectors from the same eigenspace. Therefore,

$$\text{diag}(\mathbf{A}) = (-22.83570384, -6.34920255, 0, 6.45799327, 22.72691312)^T.$$

The chosen eigenvalues will determine the eigenvectors of  $\chi_A$  used to assemble those from A. The resulting eigenvector matrix (with rounded decimals, to fit the page) is

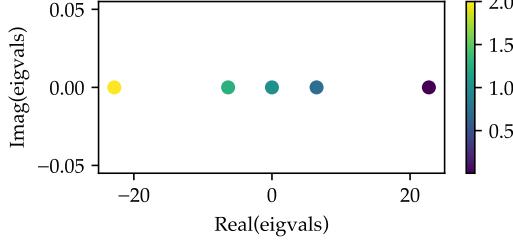
$$\mathbf{V} = \begin{pmatrix} -0.24 - 0.27j + 0.16k & 0.59 - 0.02j + 0.04k & 0 & -0.02 - 0.52j - 0.28k & -0.040 + 0.38j + 0.03k \\ -0.17 - 0.19i - 0.22j + 0.17k & -0.31 + 0.13i + 0.09j + 0.16k & -0.410 + 0.24i + 0.39j - 0.23k & -0.05 + 0.01i - 0.12j + 0.35k & -0.010 + 0.05i + 0.20j + 0.31k \\ -0.25 - 0.12i - 0.11j + 0.13k & -0.21 + 0.14i + 0.20j + 0.09k & 0.190 - 0.40i - 0.48j + 0.37k & -0.1 - 0.04i + 0.01j + 0.31k & 0.010 - 0.05i + 0.26j + 0.21k \\ -0.26 + 0.01i + 0.04j + 0.25k & -0.25 - 0.48i - 0.25j - 0.06k & 0 & 0.36 - 0.13i + 0.30j - 0.36k & -0.110 - 0.03i + 0.27j + 0.21k \\ 0.31 + 0.18i - 0.25j - 0.52k & -0.02 - 0.15i - 0.07j - 0.09k & 0 & 0.07 - 0.04i + 0.10j + 0.14k & 0.380 + 0.14i + 0.55j + 0.04k \end{pmatrix}$$

This matrix passes the invertibility test, since its determinant is  $\det(\mathbf{V}) = \det(\chi_V) = 1 \neq 0$ . Not only that, the matrix is also unitary, since the product  $\mathbf{V}\mathbf{V}^H$  yields an identity matrix. Therefore, the QGFT matrix is promptly determined.

Fig. 6.2 closes the example, depicting the graph total variation  $TV_{G,p}$  for each normalized eigenvector  $\mathbf{v}_i$ . The values are

$$TV_{G,p}(\mathbf{v}_i) \in \{2, 1.2780384, 1, 0.71719754, 0.00476406\}, \quad (6.17)$$

and remain the same as  $p$  is changed, following the commentary after (6.9).



**Fig. 6.2.:** Graph total variation (pseudocolor scale) of the normalized eigenvectors, as a function of their respective eigenvalues.

## 6.2 Filter design via QLMS optimization

We have seen so far the establishment of the basis of QGSP, borrowing concepts and tools from GSP and finding the particular consequences of having a quaternion-valued graph shift operator. The last milestone set in the beginning of the last section was to find a method for designing finite impulse response (FIR) LSI filters in QGSP and, fortunately, we can benefit from the large field of adaptive filtering with quaternion signals (Ortolani et al. 2017). In this section, we will go through how the quaternion least mean square (QLMS) algorithm can be used for FIR LSI filter design in QGSP, mimicking the method applied in in Subsection 4.2.2.

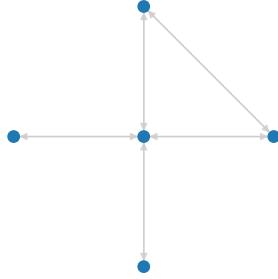
As the reader may recall, the desired frequency response of the LSI filter with filter taps  $\{h_\ell\}_{\ell=0,\dots,L}$  may be defined through a system of linear equations with the format  $h(\lambda_i) = \alpha_i$ . Using the matrix and vectors

$$\underbrace{\begin{bmatrix} 1 & \lambda_0 & \lambda_0^2 & \dots & \lambda_0^L \\ 1 & \lambda_1 & \lambda_1^2 & \dots & \lambda_1^L \\ \vdots & & \vdots & & \\ 1 & \lambda_{N-1} & \lambda_{N-1}^2 & \dots & \lambda_{N-1}^L \end{bmatrix}}_{\triangleq \mathbf{X}}, \quad \underbrace{\begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_L \end{bmatrix}}_{\triangleq \mathbf{h}}, \quad \underbrace{\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{N-1} \end{bmatrix}}_{\triangleq \mathbf{y}} \quad (6.18)$$

this system of equations can be written in matrix form as

$$\mathbf{h}^T \mathbf{X}^T = \mathbf{y}^T, \quad (6.19)$$

very similar to (4.21). The odd use of transposition in (6.19) is justified by the non-commutativity of quaternion multiplication: the following discussion requires that



**Fig. 6.3.:** Graph used in the numerical example. The edges are represented by double-headed arrows to indicate the adjacency matrix is not symmetrical.

the filter taps left-multiply the eigenvalues' powers, otherwise the QLMS equations would have different formulations.

Back to the methodology of filter design, after choosing the cutoff frequency (or frequencies) of the ideal filter one wishes to approximate and setting to 1 the values  $\alpha_i$  within the passband (zero otherwise), the intended FIR filter is designed by means of the QLMS algorithm, which was proposed by Took in 2009 (Took and Mandic 2008). Although many different versions have been formulated since then (Ogunfunmi 2015; Almeida Neto and Nascimento 2014; Ortolani and Uncini 2016), all consist in solving the optimization problem

$$\min_{\{\tilde{h}_\ell\}_{0,\dots,L}} J(\tilde{\mathbf{h}}), \quad (6.20)$$

with  $\tilde{\mathbf{h}}$  being an estimation of the desired filter, by means of the LMS-like update rule

$$\tilde{\mathbf{h}}^{(k+1)} = \tilde{\mathbf{h}}^{(k)} + \mu \nabla_{\tilde{\mathbf{h}}^{(k)}} J(\tilde{\mathbf{h}}^{(k)}). \quad (6.21)$$

The core difference between QLMS approaches stem from the subjacent definition of  $\nabla_{\tilde{\mathbf{h}}} J(\tilde{\mathbf{h}})$ , the gradient of the real-valued mean squared error cost function  $J$  with respect to the quaternion-valued vector  $\tilde{\mathbf{h}}$  (which depends on the concept of quaternion derivatives, a subject matter of many research papers, e.g. (Xu et al. 2015; Jahanchahi et al. 2012)). The approach used in this thesis was the original one, from (Took and Mandic 2008), which considers the following formulation – adapted to be in matrix form – for the estimation error and cost function,

$$\mathbf{e}(\tilde{\mathbf{h}}^{(k)}) \triangleq \mathbf{y} - (\tilde{\mathbf{h}}^{(k)T} \mathbf{X}^T)^T, \quad (6.22)$$

$$J(\tilde{\mathbf{h}}^{(k)}) \triangleq \mathbf{e}^T \bar{\mathbf{e}}, \quad (6.23)$$

in which the notation  $\mathbf{e} = \mathbf{e}(\tilde{\mathbf{h}}^{(k)})$  was used to improve readability. After the gradient derivation in (Took and Mandic 2008), one arrives at the gradient expression in matrix form

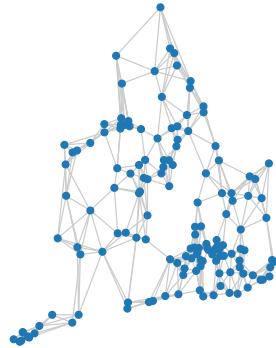
$$\nabla_{\tilde{\mathbf{h}}^{(k)}} J(\tilde{\mathbf{h}}^{(k)}) = 2 \left( \mathbf{e}^T \bar{\mathbf{X}} \right)^T - \mathbf{X}^H \bar{\mathbf{e}} \quad (6.24)$$

providing all that is needed to implement the update rule in (6.21). The reader is encouraged to refer to (Took and Mandic 2008) and verify the equivalence between the original equations and their matrix forms, just presented. Equations (6.21) to (6.24) were used exactly as they were written in the computations in Subsection 6.3.1.

As a last and more practical note, it is important to mention that, because of the structure of matrix  $\mathbf{X}$ , it is indispensable to add the extra step of normalizing the columns of  $\mathbf{X}$  before running the QLMS, otherwise convergence is slow, if not even prevented. It is a well known practice for least-mean squares regression and other algorithms based on gradient descent, which do not work well if the features have highly disproportionate scales. Let us recall that this is precisely the current case: the columns in  $\mathbf{X}$  represent variables in a regression problem (with target given by the ideal frequency response  $\mathbf{y}$ ) and they differ massively in scale. The normalization here is understood in the sense of bringing the mean and standard deviation of the feature distribution to 0 and 1, respectively. That is, for each vector  $\mathbf{x}_i = (\lambda_0^i, \lambda_1^i, \dots, \lambda_{N-1}^i)^T$  with mean  $\mu$  and standard deviation  $\sigma$ , the normalization takes place by replacing it with  $\frac{\mathbf{x}_i - \mu}{\sigma}$ .

## 6.3 Examples

This section is devoted to experimenting with QGSP in real-world datasets, tackling topics such as graph inference, spectral analysis, compression and denoising. All computations in this chapter were performed using `gspx`, an open-source Python package dedicated to implement the core concepts and tools of QGSP. It constitutes a contribution of this thesis and is available at <https://github.com/gboaviagem/gspx>.



**Fig. 6.4.:** Graph created using the 145 towns in England and Wales.

### 6.3.1 Denoising a quaternion graph signal via QLMS low-pass filtering

Let us walk through an example on spectral analysis and denoising. The graph signal will be generated using UK towns' weather data, specifically setting the humidity, atmospheric pressure, temperature and wind speed data to the  $1, i, j$  and  $k$  components of each signal sample. The data was fetched from the Open Weather Map API<sup>3</sup>, in April 20th 2022, at approximately 13:00 GMT, and a sample is shown in Table 6.1. The latitude and longitude values of UK towns originate from the LatLong.Net database<sup>4</sup>.

The graph was created using a nearest-neighbors approach, as in (3.18), since the nodes possess a clear relation to geographic location. The edge weights were defined using the following steps:

- Compute a real-valued nearest-neighbors adjacency matrix based on the latitude and longitude of UK towns (graph nodes), using the seven closest neighbors to each graph node. At this point, the real-valued edge weights correspond to the euclidian distances between nearby towns  $i$  and  $j$ , represented by  $d(i, j)$ .
- For each pair of connected nodes from the real-weighted graph in the previous step, compute the **absolute difference** in *humidity*, *atmospheric pressure*, *temperature* and *wind speed*, represented respectively by  $h(i, j)$ ,  $p(i, j)$ ,  $t(i, j)$

---

<sup>3</sup>The API is accessible, as of August 2022, at the address <https://home.openweathermap.org/>.

<sup>4</sup>Accessible, as of August 2022, at the address <https://www.latlong.net/category/towns-235-55.html>.

**Tab. 6.1.:** Sample of the dataset containing UK towns' weather data.

|   | town                                  | latitude | longitude        | humidity (%) |
|---|---------------------------------------|----------|------------------|--------------|
| 1 | St.Asaph                              | 53.257   | -3.442           | 59           |
| 2 | Welling, Bexley                       | 51.45    | 0.1056           | 39           |
| 3 | Soham, Ely                            | 52.33    | 0.3375           | 43           |
| 4 | Boulmer, Northumberland               | 55.422   | -1.536           | 79           |
| 5 | Higham Ferrers, East Northamptonshire | 52.303   | -0.592841        | 53           |
|   | pressure (hPa)                        | temp (K) | wind speed (m/s) | wind degrees |
| 1 | 1017                                  | 288.49   | 3.71             | 99           |
| 2 | 1014                                  | 290.09   | 8.23             | 80           |
| 3 | 1016                                  | 290.07   | 6.69             | 60           |
| 4 | 1020                                  | 282.77   | 4.07             | 115          |
| 5 | 1017                                  | 289.78   | 3.13             | 53           |

and  $w(i, j)$ . Then, create the quaternion  $q(i, j) = h(i, j) + ip(i, j) + jt(i, j) + kw(i, j)$  and compute the edge weight

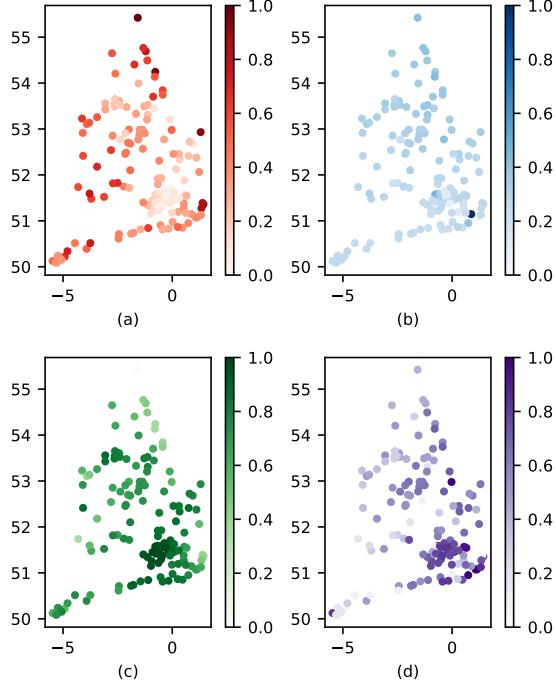
$$\mathbf{A}_{i,j} = \exp^{-1} \left( \frac{d(i, j)}{\theta} \cdot \frac{q(i, j)}{\| q(i, j) \|} \right). \quad (6.25)$$

Finally, in order to exploit the diagonalizability of the adjacency matrix, it was made Hermitian, i. e.,  $\mathbf{A}_{i,j} = \overline{\mathbf{A}}_{j,i}$ .

The reasoning behind the choice of (6.25) is to have edge weights whose magnitude reflect the expected similarity between adjacent signal samples, as in (3.18), and whose phase are aligned with the difference between such samples. The choice of  $\theta$  for the current graph, after some experimentation, was  $\theta = 2$ .

As a way to remove the scale disparity between the quaternion components (the reader may compare, for instance, pressure and wind speed), each of the 4 columns were linearly scaled down to be within the range  $[0, 1]$ . A way of visualizing the graph signal is to plot each quaternion dimension separately, as it was done in Fig. 6.5.

The QGFT of the graph signal is computed, as defined earlier on this chapter, by the projection of the graph signal on the basis of eigenvectors of the graph adjacency matrix (which is diagonalizable, since it is Hermitian). The signal spectrum is depicted in Fig. 6.6, in which the frequency components are ordered from lowest to highest frequency, using the total variation rule in (6.9). After the computation of the total variation for each eigenvector, in order to perform the frequency ordering, Fig. 6.8 was generated, demonstrating two previously discussed facts: the eigen-



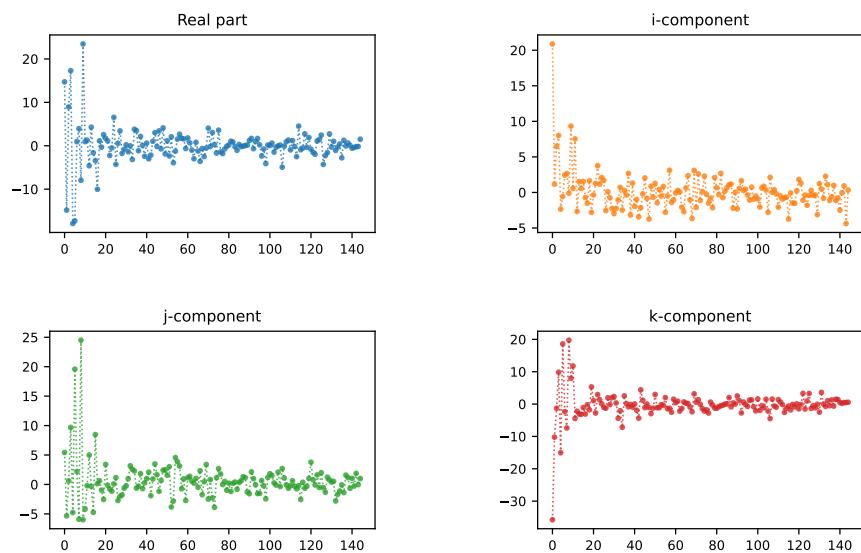
**Fig. 6.5.:** Quaternion graph signal, visualized in the vertex domain. The panels (a) to (d) depict, respectively, the real,  $i$ ,  $j$  and  $k$  components of each quaternion-valued sample.

values of the adjacency matrix are all real-valued and the total variation increases monotonically the further the eigenvalue is from the point  $|\lambda_{max}|$  in the complex plane.

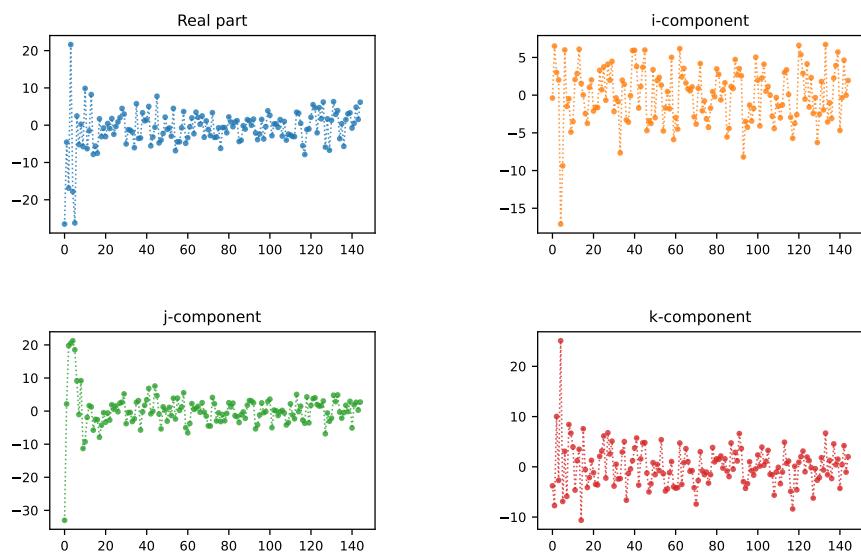
Let us now add a white noise on the graph signal, obtained by adding to the original spectrum samples from a uniform distribution with amplitude 5. The result is shown in Fig. 6.7. The goal of this exercise is to recover as perfectly as possible the original signal out of the noisy one, by using a FIR LSI low-pass filter of length  $L + 1 = 7$ , designed using QLMS.

The ideal low-pass filter was created by setting the passband to have 20% of the frequency support, i. e., the values  $\alpha_i$  in  $h(\lambda_i) = \alpha_i$  were set to 0 except for the eigenvalues corresponding to the 20% eigenvectors with smallest total variation, in which case  $\alpha_i = 1$ .

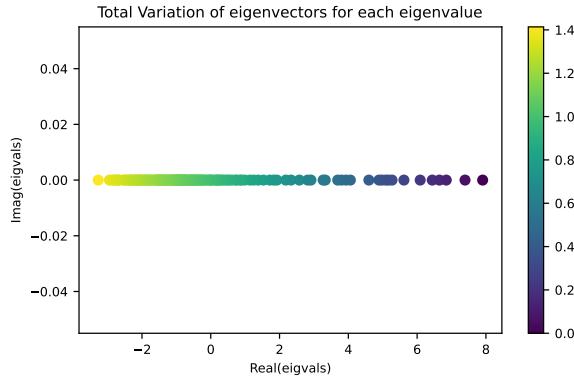
The QLMS algorithm was executed using the following values of step sizes: 0.0001, 0.0005, 0.001 and 0.002, with a maximum of 100 iterations for each step size. To avoid incurring in computational memory overflow, a “patience” of 3 iterations were considered, meaning that after 3 iterations of monotonically increasing cost function, the process for that step size was aborted. Any step size



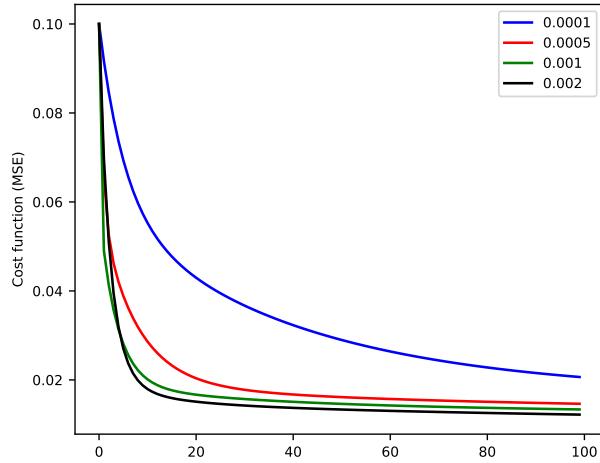
**Fig. 6.6.:** Spectrum of the quaternion graph signal, defined over the vertices of the graph with Hermitian adjacency matrix.



**Fig. 6.7.:** Spectrum of the noisy signal.



**Fig. 6.8.:** Total variation of the eigenvectors of the Hermitian graph adjacency matrix, for each respective eigenvalue.



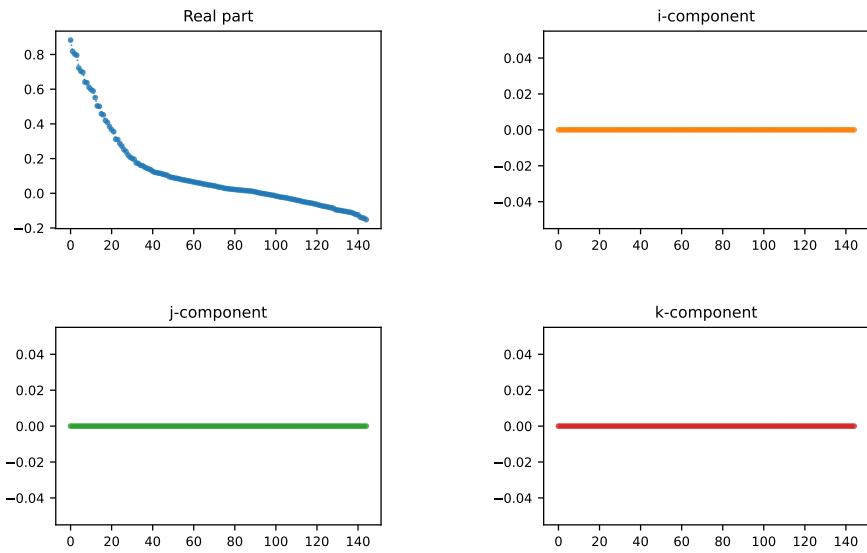
**Fig. 6.9.:** QLMS cost function for each iteration and for each step size.

greater than 0.002 was found to diverge quickly. Fig. 6.9 shows the evolution of the cost function during all 100 iterations for each of the mentioned step sizes. The optimal vector  $\tilde{\mathbf{h}}$  was found with  $\mu = 0.002$ .

The filter taps obtained by the end of the optimization were all real-valued, because the matrix  $\mathbf{X}$ , the ideal frequency response  $\mathbf{y}$  and the vector of initial filter taps (a null vector) are all real-valued:  $\tilde{\mathbf{h}}_{best} = (0.199, 0.241, 0.304, -0.177, -0.088, -0.072, 0.014)^T$  (rounded up to 3 decimal places).

The frequency response of the FIR low-pass filter which best approximated the ideal one is obtained as in (6.19),

$$\mathbf{y}_{best} = (\tilde{\mathbf{h}}_{best}^T \mathbf{X}^T)^T. \quad (6.26)$$



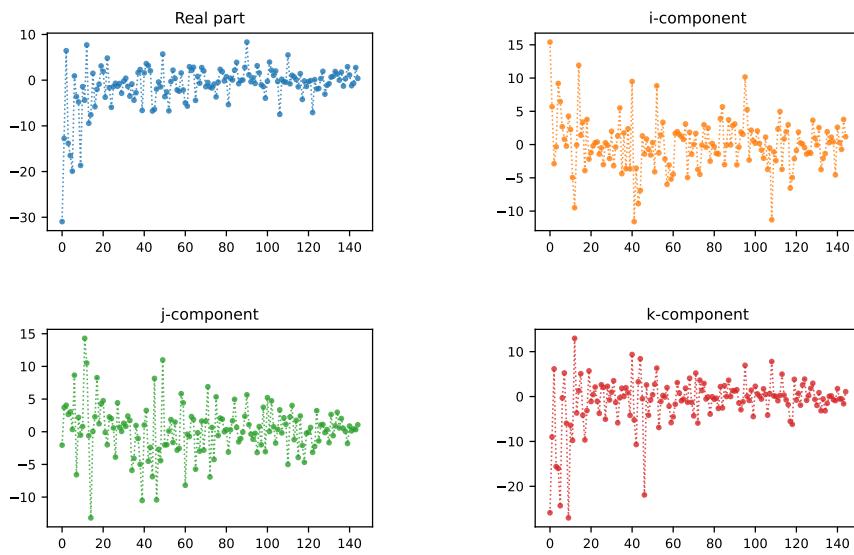
**Fig. 6.10.:** Frequency response of the FIR low-pass filter after running the QLMS.

Fig. 6.10 shows the filter frequency response.

Finally, let us evaluate how well the QLMS filter realizes the denoising in the corrupted signal. As a figure of merit, let us consider the mean squared error between the original  $s$  and filtered  $s_f$  signals, normalized by the total average energy per sample of the original signal. This so called normalized MSE (NMSE) can be written as

$$\text{NMSE}\{s_f\} \triangleq \frac{\|s - s_f\|}{N} \frac{N}{\|s\|} = \frac{\|s - s_f\|}{\|s\|}. \quad (6.27)$$

Trying out firstly the *ideal* low-pass filter, as a benchmark, we see that it was able to produce a filtered signal with  $\text{NMSE}\{s_{\text{ideal}}\} = 0.166$ , dropping by 42% the error verified in the corrupted signal  $s_n$ , which was  $\text{NMSE}\{s_n\} = 0.28991$ . The FIR low-pass filter, designed via QLMS optimization and having only  $L + 1 = 7$  filter taps, performed better, reaching an error of  $\text{NMSE}\{s_{\text{qlms}}\} = 0.15579$ , a 46% drop. The reason is justified, since the original signal was not null outside the passband, so that the FIR filter was able to preserve more of its energy in that frequency range than the ideal filter.



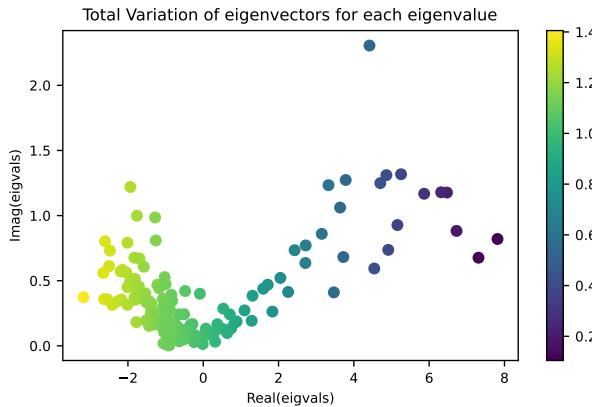
**Fig. 6.11.:** Spectrum of the original signal in the symmetric graph.

### Comparison between signal smoothness in symmetric and Hermitian graphs

The reader may have risen the question of which differences would have appeared if a symmetric adjacency matrix was used, instead of a Hermitian one. Although the graph shift operator would lose the *guaranteed* diagonalizability, the undirect edges would more naturally represent the context at hand, in which similarity between weather in close towns has no directional preference. Let us go briefly through the consequences found when switching from Hermitian to symmetric adjacency matrices.

First of all, it was somehow harder to find a value of  $\theta$  in (6.25) which led to a low-pass signal. After some trials, the most satisfactory choice was also  $\theta = 2$ , but the spectrum profile did not show an energy concentration on low frequencies as good as with the Hermitian case. See Fig. 6.11.

Another difference was the fact that the symmetric quaternion adjacency matrix does not have *only* real-valued eigenvalues. Instead, the standard eigenvalues follow the rule explained in Subsections 2.2.2 and 6.1.1, being placed either *on* or *above* the real line in the complex plane. See the eigenvalues and the total variation of eigenvectors in Fig. 6.12.



**Fig. 6.12.:** Total variation of eigenvectors of the symmetric graph adjacency matrix.

As a consequence of the eigenvalues being complex-valued, the matrix  $\mathbf{X}$  was also complex and the QLMS filter presented a complex frequency response, see Fig. 6.13. The QLMS algorithm did not suffer, given that the step sizes were conveniently chosen, and converged to the filter taps

$$\tilde{\mathbf{h}}_{best} = \begin{bmatrix} 0.199 \\ 0.227 - 0.025i \\ 0.148 - 0.065i \\ -0.062 + 0.002i \\ -0.031 + 0.049i \\ -0.030 + 0.037i \\ -0.016 + 0.020i \end{bmatrix}. \quad (6.28)$$

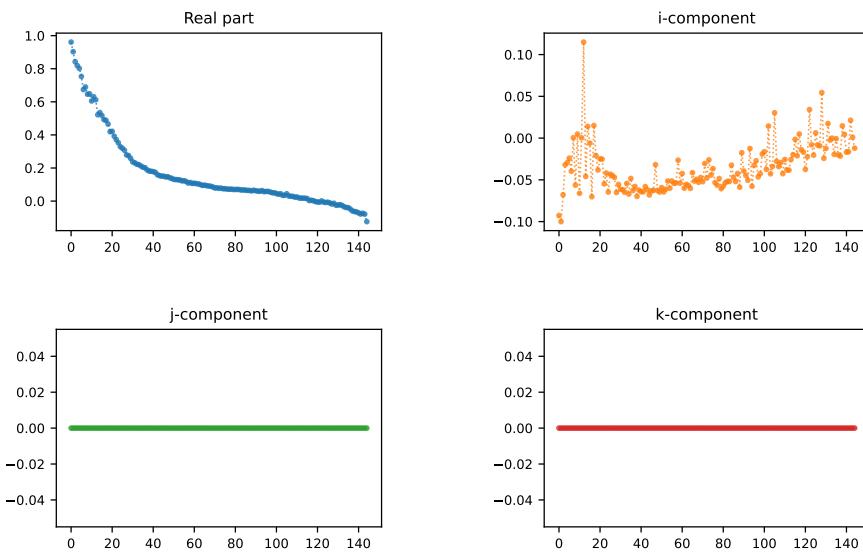
See the cost function *versus* iterations in Fig. 6.14.

### 6.3.2 Spectral analysis and compression of quaternion graph signal

For this example, let us move away from the UK graph and consider a problem with a larger network. Let us select 1000 of the United States (US) counties and create a simple nearest-neighbors graph out of their geographic coordinates. The resulting undirected graph is depicted in Fig. 6.15. It consists of 1000 vertices and 4158 edges. In order to create the quaternion graph signal, four sociodemographic variables will be used. The full source dataset<sup>5</sup> comprises 188 variables for each of the 3142 US counties, but only the following indicators were used:

---

<sup>5</sup>Extracted from OpenIntro, a non-profit organization focused on spreading high-quality open-source publications, and available at [https://www.openintro.org/data/?data=county\\_complete](https://www.openintro.org/data/?data=county_complete), accessed in October 2022.



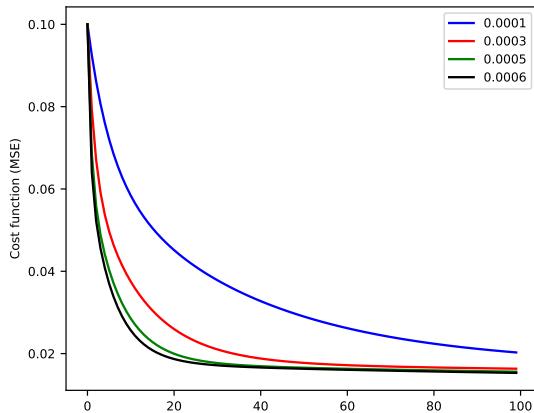
**Fig. 6.13.:** Frequency response of the FIR low-pass filter after running the QLMS in the scenario with symmetric graph adjacency matrix.

- **bachelors\_2017:** percent of population that earned a bachelor's degree in 2017.
- **median\_household\_income\_2017:** median household income as of 2017.
- **unemployment\_rate\_2017:** unemployment rate in 2017.
- **uninsured\_2017:** percent of population who were uninsured in 2017.

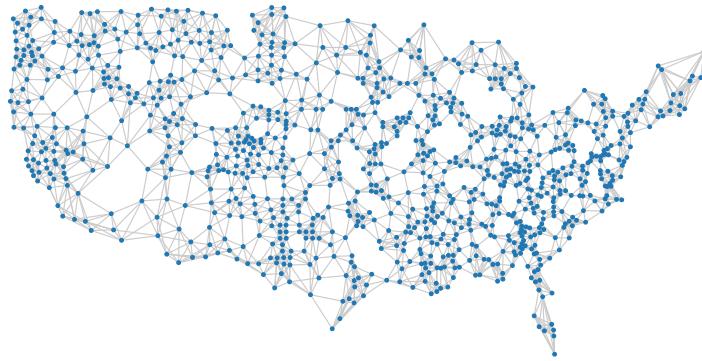
All variables relate to the financial health of american population.

Fig. 6.16 displays each quaternion component as a real-valued graph signal. From the plot (a) to (d), the signals contain the features, respectively: *bachelors\_2017*, *median\_household\_income\_2017*, *unemployment\_rate\_2017* and *uninsured\_2017*.

This example is focused on the use of QGSP for compression of multidimensional data. By keeping only a few frequency components with large fraction of the whole signal energy, and given that the underlying graph shift operator is known, the signal representation may be highly condensed. Let us create a Hermitian adjacency matrix, using (6.25) along with the four sociodemographic indicators and the geographic location of each of the 1000 US counties. After the diagonalization of the shift operator the frequency ordering through the total variation metric (see Fig. 6.17), the QGFT of the graph signal was computed and is depicted in Fig. 6.18.



**Fig. 6.14.:** QLMS cost function for each iteration and for each step size, in the scenario with symmetric graph adjacency matrix.



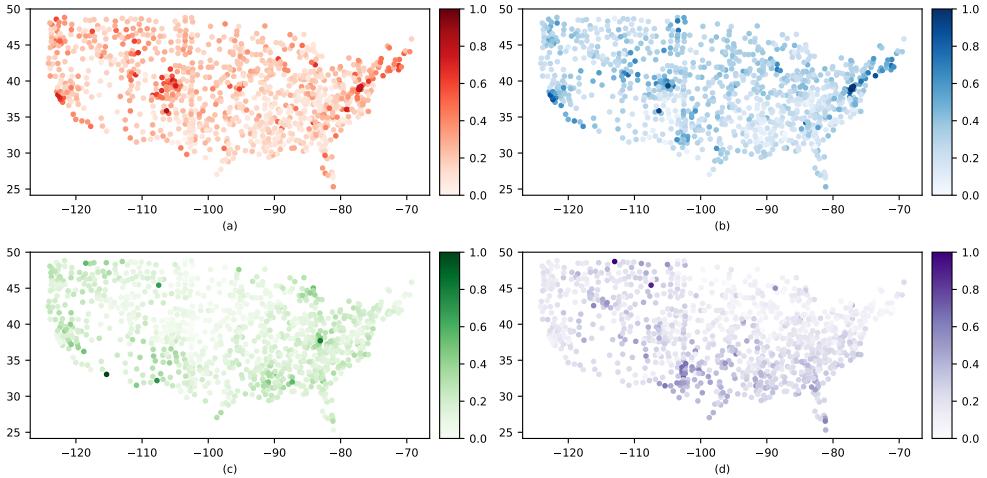
**Fig. 6.15.:** Nearest-neighbors graph of a few of the United States counties.

Before proceeding, however, a few notes regarding the computational challenges of the task must be presented. All calculations were performed (and images were generated) using our `gspx`<sup>6</sup> Python package, in a Jupyter Notebook inside a Google Colab Pro instance with 27 GB of RAM. As a representation of the computational cost of computing the QGFT in this 1000-vertices graph, using the mentioned hardware and custom-made code, the time it took to eigendecompose the graph shift operator, sort the frequencies based on total variation and normalize each eigenvector by its L1-norm was roughly around 9 minutes.

Although the signal presents a heavy noise along all the spectrum, it clearly possesses a prominent low-pass characteristic. Let us measure how well a small

---

<sup>6</sup>The library is available as an open repository: <https://github.com/gboaviagem/gspx>.



**Fig. 6.16.:** Sociodemographic quaternion-valued graph signal, split into its components: (a) real part, (b)  $i$ , (c)  $j$  and (d)  $k$ .

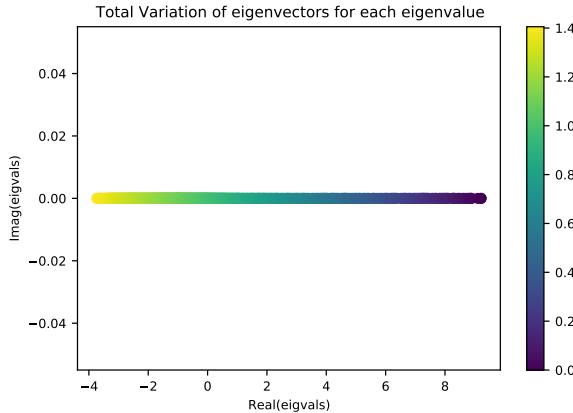
fraction of the frequency components with highest energy are able to convey most of the signal.

To this end, 20 values of compression rate were taken, linearly spaced between 5% and 95% (inclusive). A compression rate of 5%, for example, meant that only 5% of the frequency components with highest energy were kept, the rest being set to zero. After compressing the signal, it was inverse-transformed by the QGFT and compared to the original signal, by means of the same normalized mean squared error used in (6.27).

Fig. 6.19 shows how the error changed as a function of the compression rate. Since very few frequency components concentrate a relevant amount of energy, a compression rate of 95% (only 50 components preserved) reaches an NMSE of only 15.4%. A drop from 95% to 90% in the compression rate produces a steep decay in the NMSE, which goes to 11.8%, a result that makes sense for a low-pass signal: with a few more frequency components, a large chunk of the signal energy is captured. As the compression decreases (i. e., more components of the spectrum are kept) the NMSE reduction becomes less and less significant, because smaller fractions of the energy are added.

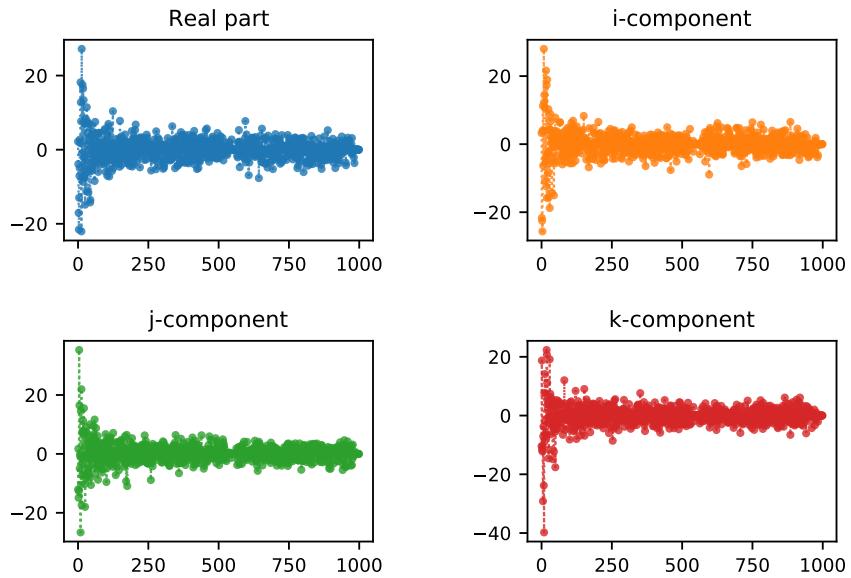
## 6.4 Summary

We have proposed a new extension of the field of graph signal processing, by pondering the question of (and proposing) how to perform signal analysis and

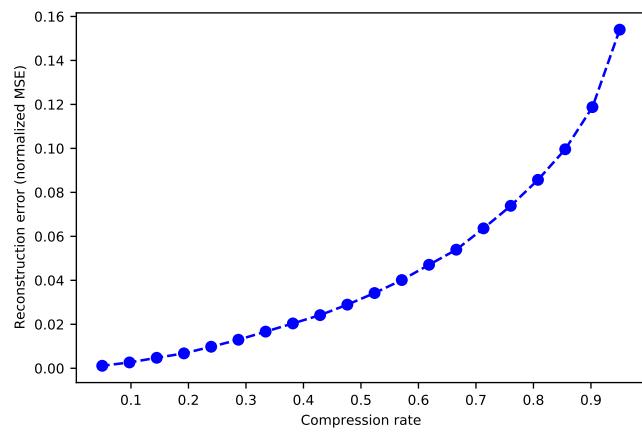


**Fig. 6.17.:** Total variation for each eigenvector of the (Hermitian) adjacency matrix of the US graph.

filtering on quaternion-valued signals defined on graphs with quaternion-valued edge weights. By sewing together tools and concepts from quaternion algebra, quaternion adaptive filtering and graph signal processing, the foundations for handling holistically 4-dimensional graph signals were laid. Besides presenting the core definitions of frequency ordering and Fourier transform, it was presented a condition under which a graph's (quaternion) adjacency matrix is diagonalizable and has unitary eigenvector matrix. Extensive examples with two real-world datasets also illustrate spectral analysis, compression and denoising with QLMS low-pass filters.



**Fig. 6.18.:** US graph signal spectrum.



**Fig. 6.19.:** Normalized mean squared error as a function of the compression rate.

# Conclusion

*Quaternions came from Hamilton after his really good work had been done; and, though beautifully ingenious, have been an unmixed evil to those who have touched them in any way [...].*

— Lord Kelvin, letter to Hayward, 1892, as cited in (Altmann 1989).

This thesis has tackled the problem of extending the basic tools of graph signal processing by changing the subjacent algebra in which signals and edge weights were defined, moving from the complex to the quaternion numbers. The exploration has also led to contributions on fractional linear operators, specifically proposing the fractional graph shift and a new way of computing the fractional QDFT, the latter having been used on an encryption scheme of color images in the form of a multiparametric transform.

The opening quote of this chapter is intended to be read in a light and ironic tone. The only “evil” the quaternions have bestowed on this thesis – if we can humorously declare this way – was the struggles with understanding the intricacies of its algebra and application on signal processing. But instead of causing any harm, the journey has borne fruit. And has done so in three fronts, let us recall the main contributions in each of them.

In the proposed fractional shift operator, it was demonstrated how it approximates the classical ideal interpolating filter in the case of ring graphs and how it can be implemented as an LSI graph filter, even for arbitrary graphs. Also, it was shown how the operator may be used in filter design to improve the frequency response with the same filter length.

In the study of the fractionalization of the QDFT, it was demonstrated how this transform share symmetric eigenvectors with the DFT, what allowed for the construction of an orthogonal eigenbasis of the QDFT and subsequent definition of its fractional (and even multiparametric) version. Since quaternions possess 4 real-valued components, an application of the 2D-MFrQDFT was proposed in the form of a holistic encryption scheme for color images with opacity layer, which was shown to provide satisfactorily large key space and key sensitivity.

Finally, the proposition of quaternion graph signal processing brought some interesting results. For instance, it was stated that a way to enforce a diagonalizable graph adjacency matrix is by making it Hermitian, which will provide also real-valued eigenvalues and a unitary QGFT matrix. During the examples provided, it was shown a method for graph inference which extracts the phase of the difference in adjacent samples and their geographic distance to build the edge weights, as well as a way of designing FIR LSI filters using QLMS optimization. All computations in the aforementioned examples were performed using a new open-source Python package, released as a Github repository for peer-review and free reuse.

## 7.1 Challenges and future works

Applying the quaternion algebra into GSP is an endeavor which is far from complete. The *foundations* of QGSP have been laid, but there are certainly challenges ahead and future work opportunities. Starting from the computational cost of QGSP algorithms, there is room for improvement in the efficiency of certain steps of signal frequency analysis. As mentioned in Section 6.1.2, the closed formula for quaternion matrix inversion presented in (Cohen and De Leo 1999) has potential to be used to accelerate the QGFT matrix calculation, specifically when computing the inverse of the eigenvector matrix when the QGSO is not Hermitian.

Another improvement in QGSP which may be explored is using better versions of the QLMS algorithm (Ogunfunmi 2015) in the design of LSI filters, aiming at faster convergence or computational efficiency. Similarly, QGSP may benefit from existing adaptive graph filtering techniques, such as using the Normalized LMS for graph signals (Spelta and Martins 2020) to design quaternion graph filters.

The contributions on fractional linear operators also leave space for future works. Regarding the fractional graph shift, an interesting thread of investigation is to find possible relationships between the FrGSO and mathematical tools for fractional diffusion in networks, since fractional diffusion has already been used to model certain phenomena that allow long-range interactions and are non-local in nature (Ilic et al. 2005; Riascos and Mateos 2014; Estrada 2021; Antil et al. 2021). In the fractional QDFT front, specifically regarding the use of the MFrQDFT for color image encryption, the interested cryptography researcher could investigate how to embed this system in real-world scenarios, or focus on other hypercomplex-based transform to expand our work, ofr instance using some hypercomplex image moments, such as ternary radial harmonic Fourier moments and quaternion polar

harmonic moments (Wang et al. 2019a; Wang et al. 2018), to perform color image encryption.



# A

## Mapping a system of linear equations from $\mathbb{H}$ to $\mathbb{C}$

Theorem 3 requires a step the reader of (Zhang 1997) may have missed: demonstrating the equivalence between

$$\mathbf{A}\mathbf{v} = \mathbf{v}\lambda \quad (\text{A.1})$$

and

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ -\overline{\mathbf{A}}_2 & \overline{\mathbf{A}}_1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1 \\ -\overline{\mathbf{v}}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{v}_1 \\ -\overline{\mathbf{v}}_2 \end{pmatrix} \lambda, \quad (\text{A.2})$$

where  $\mathbf{A} \in \mathbb{H}^{n \times n}$ ,  $\mathbf{v} \in \mathbb{H}^n$ ,  $\lambda \in \mathbb{C}$ , and the subscripts  $(\cdot)_1$  a  $(\cdot)_2$  indicate the usual symplectic decomposition components, e. g.  $q = q_1 + q_2 j$ ,  $q \in \mathbb{H}$ ,  $q_1, q_2 \in \mathbb{C}_i$ .

To fully comprehend this equivalence, it suffices to prove that any system of  $m$  quaternion linear equations, expressed in matrix form as

$$\mathbf{M}\mathbf{x} = \mathbf{y}, \quad (\text{A.3})$$

in which  $\mathbf{M} \in \mathbb{H}^{m \times n}$ ,  $\mathbf{x} \in \mathbb{H}^{n \times 1}$ ,  $\mathbf{y} \in \mathbb{H}^{m \times 1}$ , is equivalent, given the symplectic decomposition of  $\mathbf{M}$ ,  $\mathbf{x}$  and  $\mathbf{y}$ , to a system of  $2m$  complex equations

$$\begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_2 \\ -\overline{\mathbf{M}}_2 & \overline{\mathbf{M}}_1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ -\overline{\mathbf{x}}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{y}_1 \\ -\overline{\mathbf{y}}_2 \end{pmatrix}. \quad (\text{A.4})$$

Following the expansion of the left-hand side in (A.3), one has

$$\begin{aligned} \mathbf{M}\mathbf{x} &= (\mathbf{M}_1 + \mathbf{M}_2 j)(\mathbf{x}_1 + \mathbf{x}_2 j) \\ &= \mathbf{M}_1 \mathbf{x}_1 + \mathbf{M}_1 \mathbf{x}_2 j + \mathbf{M}_2 j \mathbf{x}_1 + \mathbf{M}_2 j \mathbf{x}_2 j. \end{aligned} \quad (\text{A.5})$$

It is possible to manipulate that expression using (2.8), i. e.  $jx = \bar{x}j \forall x \in \mathbb{C}$ , so (A.5) equals

$$\begin{aligned} \mathbf{M}\mathbf{x} &= \mathbf{M}_1 \mathbf{x}_1 + j\overline{\mathbf{M}}_1 \bar{\mathbf{x}}_2 + j\overline{\mathbf{M}}_2 \mathbf{x}_1 + j\overline{\mathbf{M}}_2 \mathbf{x}_2 j \\ &= \mathbf{M}_1 \mathbf{x}_1 + j\overline{\mathbf{M}}_1 \bar{\mathbf{x}}_2 + j\overline{\mathbf{M}}_2 \mathbf{x}_1 + jj\mathbf{M}_2 \bar{\mathbf{x}}_2 \\ &= (\mathbf{M}_1 \mathbf{x}_1 - \mathbf{M}_2 \bar{\mathbf{x}}_2) + j(\overline{\mathbf{M}}_1 \bar{\mathbf{x}}_2 + \overline{\mathbf{M}}_2 \mathbf{x}_1). \end{aligned} \quad (\text{A.6})$$

Replacing (A.6) in (A.3) and using  $\mathbf{y} = \mathbf{y}_1 + \mathbf{y}_2 j$  yields

$$\begin{aligned} (\mathbf{M}_1 \mathbf{x}_1 - \mathbf{M}_2 \bar{\mathbf{x}}_2) + j(\bar{\mathbf{M}}_1 \bar{\mathbf{x}}_2 + \bar{\mathbf{M}}_2 \mathbf{x}_1) &= \mathbf{y}_1 + \mathbf{y}_2 j \\ &= \mathbf{y}_1 + j\bar{\mathbf{y}}_2, \end{aligned} \quad (\text{A.7})$$

what can be written as two systems of  $m$  linear complex equations,

$$\begin{aligned} \mathbf{M}_1 \mathbf{x}_1 - \mathbf{M}_2 \bar{\mathbf{x}}_2 &= \mathbf{y}_1 \\ \bar{\mathbf{M}}_2 \mathbf{x}_1 + \bar{\mathbf{M}}_1 \bar{\mathbf{x}}_2 &= \bar{\mathbf{y}}_2, \end{aligned} \quad (\text{A.8})$$

or equivalently,

$$\begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_2 \\ \bar{\mathbf{M}}_2 & -\bar{\mathbf{M}}_1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ -\bar{\mathbf{x}}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{y}_1 \\ \bar{\mathbf{y}}_2 \end{pmatrix}. \quad (\text{A.9})$$

Finally, multiplying the second matrix equation in (A.8) by  $-1$ , (A.9) becomes

$$\begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_2 \\ -\bar{\mathbf{M}}_2 & \bar{\mathbf{M}}_1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ -\bar{\mathbf{x}}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{y}_1 \\ -\bar{\mathbf{y}}_2 \end{pmatrix}, \quad (\text{A.10})$$

precisely as in (A.4).

# Bibliography

- Alam, K. M., Saini, M., and El Saddik, A. (2015). “Toward social Internet of vehicles: Concept, architecture, and applications”. In: *IEEE Access* 3, pp. 343–357 (cit. on p. 26).
- Almeida, L. B. (1994). “The fractional Fourier transform and time-frequency representations”. In: *IEEE Transactions on signal processing* 42.11, pp. 3084–3091 (cit. on pp. 24, 25).
- Almeida Neto, F. G. and Nascimento, V. H. (2014). “Low-Complexity Quaternion Adaptive Filters”. In: *arXiv e-prints*, arXiv–1410 (cit. on p. 96).
- Altmann, S. L. (1989). “Hamilton, Rodrigues, and the quaternion scandal”. In: *Mathematics Magazine* 62.5, pp. 291–308 (cit. on p. 111).
- Antil, H., Berry, T., and Harlim, J. (Sept. 2021). “Fractional diffusion maps”. In: *Applied and Computational Harmonic Analysis* 54, pp. 145–175 (cit. on pp. 48, 112).
- Ben Saad, L. and Beferull-Lozano, B. (2018). “Stochastic Graph Filtering Under Asymmetric Links in Wireless Sensor Networks”. In: *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5 (cit. on p. 43).
- Benzi, K., Kalofolias, V., Bresson, X., and Vandergheynst, P. (2016a). “Song recommendation with non-negative matrix factorization and graph total variation”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, pp. 2439–2443 (cit. on p. 2).
- Benzi, K., Kalofolias, V., Bresson, X., and Vandergheynst, P. (2016b). “Song recommendation with non-negative matrix factorization and graph total variation”. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2439–2443 (cit. on p. 43).
- Benzi, M. and Simunec, I. (2021). “Rational Krylov methods for fractional diffusion problems on graphs”. In: *BIT Numerical Mathematics* (cit. on p. 48).
- Blahut, R. E. (2010). *Fast algorithms for signal processing*. Cambridge University Press (cit. on pp. 1, 63).
- Bultheel, A. and Martinez Sulbaran, H. (2002). “A shattered survey of the Fractional Fourier Transform”. In: *TW Reports* (cit. on p. 24).
- Candan, Ç. (2011). “On the eigenstructure of DFT matrices [DSP education]”. In: *IEEE Signal Processing Magazine* 28.2, pp. 105–108 (cit. on p. 68).
- Chai, X., Fu, X., Gan, Z., Lu, Y., and Chen, Y. (2019). “A color image cryptosystem based on dynamic DNA encryption and chaos”. In: *Signal Processing* 155, pp. 44–62 (cit. on p. 76).
- Chandra, S. and Svalbe, I. (2014). “Exact image representation via a number-theoretic Radon transform”. In: *IET Computer Vision* 8.4, pp. 338–346 (cit. on pp. 1, 63).

- Chen, B., Yu, M., Tian, Y., Li, L., Wang, D., and Sun, X. (2018a). “Multiple-parameter fractional quaternion Fourier transform and its application in colour image encryption”. In: *IET Image Processing* 12.12, pp. 2238–2249 (cit. on pp. 68, 74).
- Chen, B., Zhou, C., Jeon, B., Zheng, Y., and Wang, J. (2018b). “Quaternion discrete fractional random transform for color image adaptive watermarking”. In: *Multimedia Tools and Applications* 77.16, pp. 20809–20837 (cit. on pp. 1, 63, 84).
- Chen, S., Cerda, F., Rizzo, P., Bielak, J., Garrett, J. H., and Kovačević, J. (2014a). “Semi-supervised multiresolution classification using adaptive graph filtering with application to indirect bridge structural health monitoring”. In: *IEEE Trans. Signal Process.* 62.11. DOI: 10.1109/TSP.2014.2313528, pp. 2879–2893 (cit. on p. 2).
- Chen, S., Sandryhaila, A., Moura, J. M. F., and Kovacevic, J. (2014b). “Signal denoising on graphs via graph filtering”. In: *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 872–876 (cit. on p. 43).
- Chen, S., Varma, R., Singh, A., and Kovačević, J. (2016). “Signal recovery on graphs: Fundamental limits of sampling strategies”. In: *IEEE Trans. Signal Inf. Process. Netw.* 2.4. DOI: 10.1109/TSIPN.2016.2614903, pp. 539–554 (cit. on p. 2).
- Chung, F. (2010). “Graph theory in the information age”. In: *Notices of the AMS* 57.6, pp. 726–732 (cit. on p. 26).
- Chung, F. R. K. (1997). *Spectral graph theory*. 92. American Mathematical Soc. (cit. on p. 30).
- Cohen, N. and De Leo, S. (1999). “The quaternionic determinant”. In: *arXiv preprint math-ph/9907015* (cit. on pp. 86, 112).
- De Leo, S. and Scolarici, G. (2000). “Right eigenvalue equation in quaternionic quantum mechanics”. In: *Journal of Physics A: Mathematical and General* 33.15, p. 2971 (cit. on p. 17).
- Deri, J. A. and Moura, J. M. (2017). “Spectral projector-based graph Fourier transforms”. In: *IEEE Journal of Selected Topics in Signal Processing* 11.6, pp. 785–795 (cit. on p. 36).
- Djuric, P. M. and Richard, C. (2018). *Cooperative and Graph Signal Processing: Principles and Applications*. Academic Press (cit. on p. 43).
- Ell, T. A. (1993). “Quaternion-Fourier transforms for analysis of two-dimensional linear time-invariant partial differential systems”. In: *Proceedings of 32nd IEEE Conference on Decision and Control*. IEEE, pp. 1830–1841 (cit. on pp. 1, 63).
- Ell, T. A., Le Bihan, N., and Sangwine, S. J. (2014). *Quaternion Fourier transforms for signal and image processing*. John Wiley & Sons (cit. on pp. 10, 12, 20–22, 65).
- Ell, T. A. and Sangwine, S. J. (2006). “Hypercomplex Fourier transforms of color images”. In: *IEEE Transactions on image processing* 16.1, pp. 22–35 (cit. on p. 68).
- Ell, T. A. and Sangwine, S. J. (2007). “Hypercomplex Fourier transforms of color images”. In: *IEEE Transactions on image processing* 16.1, pp. 22–35 (cit. on pp. 1, 63, 84).

- Estrada, E. (July 2021). "Path Laplacians versus fractional Laplacians as nonlocal operators on networks". In: *New Journal of Physics* 23.7, p. 073049 (cit. on pp. 48, 112).
- Evans, C. J., Sangwine, S. J., and Ell, T. A. (2000). "Hypercomplex color-sensitive smoothing filters". In: *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*. Vol. 1. IEEE, pp. 541–544 (cit. on pp. 1, 63, 84).
- Fan, J., Tepedelenlioglu, C., and Spanias, A. (2019a). "Global Optimization of Graph Filters with Multiple Shift Matrices". In: *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 2082–2086 (cit. on pp. 43, 44).
- Fan, J., Tepedelenlioglu, C., and Spanias, A. (2019b). "Graph Filtering with Multiple Shift Matrices". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3557–3561 (cit. on pp. 43, 44).
- Feofiloff, P., Kohayakawa, Y., and Wakabayashi, Y. (2011). "Uma Introdução Sucinta à Teoria dos Grafos". In: <https://www.ime.usp.br/~pf/teoriadosgrafos/> (cit. on p. 28).
- Ferreira, F. A. B. S. and Lima, J. B. (2020). "A robust 3D point cloud watermarking method based on the graph Fourier transform". In: *Multimedia Tools and Applications* 79.3, pp. 1921–1950 (cit. on p. 43).
- Flamant, J., Le Bihan, N., and Chainais, P. (2017a). "Time-frequency analysis of bivariate signals". In: *Appl. Comput. Harmon. Anal.* (cit. on pp. 1, 22, 63, 84).
- Flamant, J., Chainais, P., and Le Bihan, N. (2018). "A complete framework for linear filtering of bivariate signals". In: *IEEE Transactions on Signal Processing* 66.17, pp. 4541–4552 (cit. on pp. 1, 63, 84).
- Flamant, J., Le Bihan, N., and Chainais, P. (2017b). "Spectral analysis of stationary random bivariate signals". In: *IEEE Transactions on Signal Processing* 65.23, pp. 6135–6145 (cit. on pp. 1, 20, 63, 84).
- Gama, F., Isufi, E., Ribeiro, A., and Leus, G. (Dec. 2019). "Controllability of Bandlimited Graph Processes Over Random Time Varying Graphs". In: *IEEE Transactions on Signal Processing* 67.24, pp. 6440–6454 (cit. on p. 43).
- Gavili, A. and Zhang, X.-P. (Dec. 2017). "On the Shift Operator, Graph Frequency, and Optimal Filtering in Graph Signal Processing". In: *IEEE Transactions on Signal Processing* 65.23, pp. 6303–6318. DOI: 10.1109/TSP.2017.2752689 (cit. on pp. 43, 44).
- Girault, B., Goncalves, P., and Fleury, E. (Dec. 2015). "Translation on Graphs: An Isometric Shift Operator". In: *IEEE Signal Process. Lett.* 22.12, pp. 2416–2420. DOI: 10.1109/lsp.2015.2488279 (cit. on p. 43).
- Golubski, A. J., Westlund, E. E., Vandermeer, J., and Pascual, M. (May 2016). "Ecological Networks over the Edge: Hypergraph Trait-Mediated Indirect Interaction (TMII) Structure". In: *Trends in Ecology & Evolution* 31.5. DOI: 10.1016/j.tree.2016.02.006, pp. 344–354 (cit. on p. 26).
- Graf, S., Herbig, T., Buck, M., and Schmidt, G. (2015). "Features for voice activity detection: a comparative analysis". In: *EURASIP Journal on Advances in Signal Processing* 2015.1, p. 91 (cit. on p. 1).

- Grigoryan, A. M. and Agaian, S. S. (2018). "Quaternion and octonion color image processing with MATLAB". In: SPIE-The International Society for Optical Engineering (cit. on p. 22).
- Guanlei, X., Xiaotong, W., and Xiaogang, X. (2008). "Fractional quaternion Fourier transform, convolution and correlation". In: *Signal Processing* 88.10, pp. 2511–2517 (cit. on pp. 20, 63).
- Guo, L., Ning, Z., Song, Q., Zhang, L., and Jamalipour, A. (2016). "A QoS-oriented High-Efficiency Resource Allocation Scheme in Wireless Multimedia Sensor Networks". In: *IEEE Sensors J.* DOI: 10.1109/JSEN.2016.2645709 (cit. on p. 26).
- Hamilton, W. R. (1837). "Theory of conjugate functions, or algebraic couples; with a preliminary and elementary essay on algebra as the science of pure time". In: *Transactions of the Royal Irish Academy* 17.part 1, pp. 293–422 (cit. on p. 7).
- Higham, N. J. (2008). *Functions of Matrices: Theory and Computation (Other Titles in Applied Mathematics)*. Society for Industrial & Applied Mathematics, U.S. (cit. on p. 46).
- Hsue, W.-L. (2018). "Enhancing Security of Double Random Phase Encryption Schemes Based on Discrete Fractional Fourier Transforms". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* (cit. on pp. 64, 74, 76).
- Hu, T., Liu, Y., Gong, L.-H., Guo, S.-F., and Yuan, H.-M. (2017). "Chaotic image cryptosystem using DNA deletion and DNA insertion". In: *Signal Processing* 134, pp. 234–243 (cit. on p. 76).
- Huang, W., Marques, A. G., and Ribeiro, A. R. (Oct. 2018). "Rating Prediction via Graph Signal Processing". In: *IEEE Transactions on Signal Processing* 66.19, pp. 5066–5081 (cit. on p. 43).
- Ilic, M., Liu, F., Turner, I., and Anh, V. (2005). "Numerical approximation of a fractional-in-space diffusion equation, I". In: *Fractional Calculus & Applied Analysis* 8.3, pp. 323–341 (cit. on pp. 48, 112).
- Jahanchahi, C., Took, C. C., and Mandic, D. P. (2012). "On gradient calculation in quaternion adaptive filtering". In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 3773–3776 (cit. on p. 96).
- Jain, R. K., Moura, J. M. F., and Kontokosta, C. E. (2014). "Big data + big cities: Graph signals of urban air pollution [exploratory SP]". In: *IEEE Signal Process. Mag.* 31.5. DOI: 10.1109/MSP.2014.2330357, pp. 130–136 (cit. on p. 27).
- Jiang, M., Liu, W., Li, Y., and Zhang, X. (2013). "Frequency-domain quaternion-valued adaptive filtering and its application to wind profile prediction". In: *2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)*. IEEE, pp. 1–5 (cit. on p. 20).
- Jiang, M., Liu, W., and Li, Y. (2014). "A general quaternion-valued gradient operator and its applications to computational fluid dynamics and adaptive beamforming". In: *2014 19th International Conference on Digital Signal Processing*. IEEE, pp. 821–826 (cit. on pp. 2, 19, 22).
- Jiang, X., Tian, Z., and Li, K. (Oct. 2021). "A Graph-Based Approach for Missing Sensor Data Imputation". In: *IEEE Sensors Journal* 21.20, pp. 23133–23144 (cit. on p. 43).

- Kang, X., Ming, A., and Tao, R. (2018). “Reality-preserving multiple parameter discrete fractional angular transform and its application to color image encryption”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 29.6, pp. 1595–1607 (cit. on pp. 26, 72).
- Kang, X. and Tao, R. (2018). “Color image encryption using pixel scrambling operator and reality-preserving MPFRHT”. In: *IEEE Trans. Circuits Syst. Video Technol.* (cit. on pp. 26, 72).
- Kleiner, I. (2007). “History of Ring Theory”. In: *A History of Abstract Algebra*. Birkhäuser Boston, pp. 41–61. DOI: 10.1007/978-0-8176-4685-1\_3 (cit. on p. 9).
- Knockaert, L. (1994). “A generalized Möbius transform and arithmetic Fourier transforms”. In: *IEEE Trans. Signal Process.* 42.11, pp. 2967–2971 (cit. on p. 63).
- Lee, H.-C. (1948). “Eigenvalues and canonical forms of matrices with quaternion coefficients”. In: *Proceedings of the Royal Irish Academy. Section A: Mathematical and Physical Sciences*. Vol. 52. JSTOR, pp. 253–260 (cit. on p. 18).
- Li, Y. N. (2013). “Quaternion polar harmonic transforms for color images”. In: *IEEE Signal Processing Letters* 20.8, pp. 803–806 (cit. on pp. 1, 63, 84).
- Lima, J. B. and Campello de Souza, R. M. (Apr. 2013). “Fractional cosine and sine transforms over finite fields”. In: *Linear Algebra and its Applications* 438.8, pp. 3217–3230 (cit. on p. 63).
- Lima, J. B., Oliveira Neto, J. R. de, and Figueiredo, R. B. D. (July 2018). “A unified approach for defining random discrete fractional transforms”. In: *Optik* 165, pp. 388–394 (cit. on pp. 24, 72).
- Lima, J. B. and Campello de Souza, R. M. (2011). “Finite field trigonometric transforms”. In: *Applicable Algebra in Engineering, communication and computing* 22.5, pp. 393–411 (cit. on p. 1).
- Liu, H. and Wang, X. (2010). “Color image encryption based on one-time keys and robust chaotic maps”. In: *Computers & Mathematics with Applications* 59.10, pp. 3320–3327 (cit. on p. 64).
- Liu, H. and Wang, X. (2011). “Color image encryption using spatial bit-level permutation and high-dimension chaotic system”. In: *Optics Communications* 284.16-17, pp. 3895–3903 (cit. on p. 64).
- Liu, H., Wang, X., et al. (2012). “Image encryption using DNA complementary rule and chaotic maps”. In: *Applied Soft Computing* 12.5, pp. 1457–1466 (cit. on p. 64).
- Liu, Y., Yang, L., You, K., Guo, W., and Wang, W. (2019). “Graph Learning Based on Spatiotemporal Smoothness for Time-Varying Graph Signal”. In: *IEEE Access* 7, pp. 62372–62386 (cit. on p. 43).
- Longxuan, C., Renmin, H., and Liangtao, W. (June 1996). “Jordan canonical forms of matrices over quaternion field”. In: *Applied Mathematics and Mechanics* 17.6, pp. 559–568. DOI: 10.1007/bf00119754 (cit. on p. 85).

- Lu, Z., Xu, Y., Yang, X., Song, L., and Traversoni, L. (2007). “2D quaternion Fourier transform: the spectrum properties and its application in color image registration”. In: *2007 IEEE International Conference on Multimedia and Expo*. IEEE, pp. 1715–1718 (cit. on p. 68).
- Ma, F., Yao, B., and Yao, M. (2016). “Non-planar unclustered Peterson graphs as scale-free models of the Internet of Things”. In: *Information Technology, Networking, Electronic and Automation Control Conference, IEEE*. IEEE, pp. 1040–1043 (cit. on p. 26).
- Mallat, S. (1999). *A wavelet tour of signal processing*. Academic press (cit. on p. 38).
- McKinsey, G. I. et al. (2011). “Big data: the next frontier for innovation, competition and productivity”. In: *McKinsey Global Institute* (cit. on p. 26).
- Mei, J. and Moura, J. M. (2016). “Signal Processing on Graphs: Performance of Graph Structure Estimation”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, pp. 6165–6169 (cit. on pp. 27, 33).
- Mendlovic, D. and Ozaktas, H. M. (1993). “Fractional Fourier transforms and their optical implementation: I”. In: *JOSA A* 10.9, pp. 1875–1881 (cit. on p. 23).
- Mollaebrahim, S. and Beferull-Lozano, B. (2021). “Design of Asymmetric Shift Operators for Efficient Decentralized Subspace Projection”. In: *IEEE Transactions on Signal Processing* 69, pp. 2056–2069 (cit. on pp. 43, 44).
- Murty, U. and Bondy, A. (2008). *Graph Theory (graduate texts in mathematics 244)* (cit. on p. 30).
- Murugan, B. and Gounder, A. G. N. (2016). “Image encryption scheme based on block-based confusion and multiple levels of diffusion”. In: *IET Computer Vision* 10.6, pp. 593–602 (cit. on p. 76).
- Namias, V. (1980). “The fractional order Fourier transform and its application to quantum mechanics”. In: *IMA Journal of Applied Mathematics* 25.3, pp. 241–265 (cit. on pp. 23, 24).
- National Climatic Data Center* (n.d.). URL: <ftp://ftp.ncdc.%20noaa.gov/pub/data/gsod> (cit. on p. 54).
- Ogunfunmi, T. (2015). “Adaptive filtering using complex data and quaternions”. In: *Procedia Computer Science* 61, pp. 334–340 (cit. on pp. 96, 112).
- Oliveira Neto, J. R. de and Lima, J. B. (2017). “Discrete fractional Fourier transforms based on closed-form Hermite–Gaussian-like DFT eigenvectors”. In: *IEEE Transactions on Signal Processing* 65.23, pp. 6171–6184 (cit. on pp. 70, 81).
- Oppenheim, A. V. and Schafer, R. W. (2013). *Discrete-time Signal Processing*. Pearson (cit. on p. 49).
- Oppenheim, A. V., Willsky, A. S., and Nawab, S. H. (1997). *Signals and Systems*. Prentice-Hall signal processing series. Prentice Hall (cit. on p. 35).
- Oppenheim, A. V. (1999). *Discrete-time signal processing*. Pearson Education India (cit. on p. 1).

- Ortega, A., Frossard, P., Kovacevic, J., Moura, J. M. F., and Vandergheynst, P. (May 2018). “Graph Signal Processing: Overview, Challenges, and Applications”. In: *Proceedings of the IEEE* 106.5, pp. 808–828. DOI: 10.1109/jproc.2018.2820126 (cit. on p. 43).
- Ortolani, F. and Uncini, A. (2016). “Quaternion digital signal processing: A hypercomplex approach to information processing”. In: *Control and Communications (SIBCON), 2016 International Siberian Conference on*. IEEE, pp. 1–7 (cit. on p. 96).
- Ortolani, F., Comminiello, D., Scarpiniti, M., and Uncini, A. (2017). “Frequency domain quaternion adaptive filters: Algorithms and convergence performance”. In: *Signal Processing* 136, pp. 69–80 (cit. on p. 95).
- Pedrouzo-Ulla, A., Troncoso-Pastoriza, J. R., and Pérez-González, F. (2017). “Number theoretic transforms for secure signal processing”. In: *IEEE Trans. Inf. Forensics Security* 12.5, pp. 1125–1140 (cit. on pp. 1, 63).
- Pei, S.-C., Ding, J.-J., and Chang, K.-W. (2010). “Eigenfunctions, eigenvalues, and fractionalization of the quaternion and biquaternion Fourier transforms”. In: *2010 18th European Signal Processing Conference*. IEEE, pp. 1874–1878 (cit. on p. 66).
- Pirayre, A., Couprie, C., Bidard, F., Duval, L., and Pesquet, J.-C. (2015). “BRANE Cut: biologically-related a priori network enhancement with graph cuts for gene regulatory network inference”. In: *BMC Bioinformatics* 16.1. DOI: 10.1186/s12859-015-0754-2, p. 368 (cit. on p. 2).
- Pirayre, A., Couprie, C., Duval, L., and Pesquet, J.-C. (2017). “BRANE Clust: Cluster-Assisted Gene Regulatory Network Inference Refinement”. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. DOI: 10.1109/TCBB.2017.2688355 (cit. on p. 2).
- Püschel, M. and Moura, J. M. F. (2008a). “Algebraic Signal Processing Theory: 1-D Space”. In: *IEEE Trans. Signal Process.* 56.8. DOI: 10.1109/TSP.2008.925259, pp. 3586–3599 (cit. on p. 33).
- Püschel, M. and Moura, J. M. F. (2008b). “Algebraic Signal Processing Theory: Foundation and 1-D Time”. In: *IEEE Trans. Signal Process.* 56.8. DOI: 10.1109/TSP.2008.925261, pp. 3572–3585 (cit. on p. 33).
- Rabiner, L. and Schafer, R. (2010). *Theory and applications of digital speech processing*. Prentice Hall Press (cit. on p. 1).
- Rajapaksha, N., Madanayake, A., Cintra, R. J., Adikari, J., and Dimitrov, V. S. (2014). “VLSI Computational Architectures for the Arithmetic Cosine Transform”. In: *IEEE Trans. Comput.* 64.9, pp. 2708–2715 (cit. on p. 63).
- Riascos, A. P. and Mateos, J. L. (Sept. 2014). “Fractional dynamics on networks: Emergence of anomalous diffusion and Lévy flights”. In: *Physical Review E* 90 (3), p. 032809 (cit. on pp. 48, 112).
- Ribeiro, G. B. and Lima, J. B. (July 2018). “Graph Signal Processing in a Nutshell”. In: *Journal of Communication and Information Systems* 33.1 (cit. on pp. 36, 43).
- Ribeiro, G. B., Neto, J. R. D. O., and Lima, J. B. (2022). “On the Fractionalization of the Shift Operator on Graphs”. In: *IEEE Access* 10, pp. 16468–16478. DOI: 10.1109/ACCESS.2022.3149755 (cit. on p. 44).

- Ricaud, B., Borgnat, P., Tremblay, N., Gonçalves, P., and Vandergheynst, P. (2019). “Fourier could be a data scientist: From graph Fourier transform to signal processing on graphs”. In: *Comptes Rendus Physique* 20.5, pp. 474–488 (cit. on p. 2).
- Roopkumar, R. (2016). “Quaternionic one-dimensional fractional Fourier transform”. In: *Optik* 127.24, pp. 11657–11661 (cit. on pp. 63, 72).
- Rossi, M. and Frossard, P. (2017). “Graph-based light field super-resolution”. In: *Multimedia Signal Processing (MMSP), 2017 IEEE 19th International Workshop on*. IEEE, pp. 1–6 (cit. on p. 2).
- Rudin, W. (1987). *Real and complex analysis*. Tata McGraw-Hill Education (cit. on p. 38).
- Sandryhaila, A. and Moura, J. M. F. (2013a). “Discrete signal processing on graphs”. In: *IEEE Trans. Signal Process.* 61.7. DOI: 10.1109/TSP.2013.2238935, pp. 1644–1656 (cit. on p. 35).
- Sandryhaila, A. and Moura, J. M. F. (May 2013b). “Discrete signal processing on graphs: Graph filters”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. DOI: 10.1109%2Ficassp.2013.6638849. Institute of Electrical and Electronics Engineers (IEEE) (cit. on pp. 2, 34, 35).
- Sandryhaila, A. and Moura, J. M. F. (May 2013c). “Discrete signal processing on graphs: Graph Fourier transform”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. DOI: 10.1109%2Ficassp.2013.6638850. Institute of Electrical and Electronics Engineers (IEEE) (cit. on pp. 2, 36).
- Sandryhaila, A. and Moura, J. M. F. (2014a). “Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure”. In: *IEEE Signal Process. Mag.* 31.5. DOI: 10.1109/MSP.2014.2329213, pp. 80–90 (cit. on pp. 27, 44).
- Sandryhaila, A. and Moura, J. M. F. (June 2014b). “Discrete Signal Processing on Graphs: Frequency Analysis”. In: *IEEE Trans. Signal Process.* 62.12. DOI: 10.1109%2Ftsp.2014.2321121, pp. 3042–3054 (cit. on pp. 37–39, 52, 54).
- Sandryhaila, A. and Moura, J. M. (2012). “Nearest-neighbor Image Model”. In: *Image Processing (ICIP), 2012 19th IEEE International Conference on*. IEEE, pp. 2521–2524 (cit. on p. 31).
- Sangwine, S. J. (1996). “Fourier transforms of colour images using quaternion or hypercomplex numbers”. In: *Electronics letters* 32.21, pp. 1979–1980 (cit. on pp. 1, 63, 83).
- Santos, J. S. and Ferreira, M. A. (2011). “Álgebra de Quaternions”. In: (cit. on p. 7).
- Sardellitti, S., Barbarossa, S., and Di Lorenzo, P. (2017). “On the graph Fourier transform for directed graphs”. In: *IEEE J. Sel. Topics Signal Process.* DOI: 10.1109/JSTSP.2017.2726979 (cit. on p. 2).
- Sardellitti, S., Barbarossa, S., and Lorenzo, P. D. (Dec. 2016). “Graph topology inference based on transform learning”. In: *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. DOI: 10.1109/globalsip.2016.7905863. IEEE (cit. on p. 33).

- Schafer, R. (1955). "An Introduction to Nonassociative Algebras". In: *American Mathematical Society* 61, pp. 469–484 (cit. on p. 7).
- Segarra, S., Marques, A. G., Leus, G., and Ribeiro, A. (2015). "Interpolation of graph signals using shift-invariant graph filters". In: *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, pp. 210–214 (cit. on p. 2).
- Shafipour, R., Khodabakhsh, A., Mateos, G., and Nikolova, E. (2018). "Digraph Fourier Transform via Spectral Dispersion Minimization". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6284–6288 (cit. on pp. 43, 44).
- Shafipour, R., Khodabakhsh, A., Mateos, G., and Nikolova, E. (2019). "A Directed Graph Fourier Transform With Spread Frequency Components". In: *IEEE Transactions on Signal Processing* 67.4, pp. 946–960 (cit. on pp. 43, 44).
- Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. (2013). "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains". In: *IEEE Signal Process. Mag.* 30.3. DOI: 10.1109/MSP.2012.2235192, pp. 83–98 (cit. on pp. 27, 28, 33, 83).
- Silva Jr., L. C. da and Lima, J. B. (Dec. 2018). "The Quaternion Fourier Number Transform". In: *Circuits, Systems and Signal Processing* 37.12, pp. 5486–5506 (cit. on p. 63).
- Singh, N. and Sinha, A. (2008). "Optical image encryption using fractional Fourier transform and chaos". In: *Optics and Lasers in Engineering* 46.2, pp. 117–123 (cit. on pp. 74, 75).
- Smart, N. et al. (2018). "Algorithms, Key Size and Protocols Report (2018)". In: *ECRYPT-CSA, H2020-ICT-2014 – Project 645421* (cit. on p. 76).
- Spelta, M. J. M. and Martins, W. A. (2020). "Normalized LMS algorithm and data-selective strategies for adaptive graph signal estimation". In: *Signal Processing* 167, p. 107326 (cit. on p. 112).
- Su, X., Rizkallah, M., Maugey, T., and Guillemot, C. (2017). "Graph-based light fields representation and coding using geometry information". In: *IEEE International Conference on Image Processing (ICIP)* (cit. on p. 2).
- Sun, B., Zhao, D., Shi, X., and He, Y. (2021). "Modeling Global Spatial-Temporal Graph Attention Network for Traffic Prediction". In: *IEEE Access* 9, pp. 8581–8594 (cit. on p. 43).
- Tao, R., Meng, X.-Y., and Wang, Y. (2010). "Image encryption with multiorders of fractional Fourier transforms". In: *IEEE Trans. Inf. Forensics Security* 5.4, pp. 734–738 (cit. on pp. 26, 72).
- Thanou, D., Dong, X., Kressner, D., and Frossard, P. (Sept. 2017). "Learning Heat Diffusion Graphs". In: *IEEE Transactions on Signal and Information Processing over Networks* 3.3, pp. 484–499 (cit. on p. 48).
- Took, C. C. and Mandic, D. P. (2008). "The quaternion LMS algorithm for adaptive filtering of hypercomplex processes". In: *IEEE Transactions on Signal Processing* 57.4, pp. 1316–1327 (cit. on pp. 1, 84, 96, 97).

- Tremblay, N. and Borgnat, P. (2014). "Graph wavelets for multiscale community mining". In: *IEEE Trans. Signal Process.* 62.20. DOI: 10.1109/TSP.2014.2345355, pp. 5227–5239 (cit. on p. 2).
- Tsitsvero, M., Barbarossa, S., and Di Lorenzo, P. (2016). "Signals on graphs: Uncertainty principle and sampling". In: *IEEE Trans. Signal Process.* 64.18. DOI: 10.1109/TSP.2016.2573748, pp. 4845–4860 (cit. on p. 2).
- Välimäki, V. (1995). *Discrete-time modeling of acoustic tubes using fractional delay filters*. Helsinki University of Technology (cit. on p. 49).
- Vergin, R., O'Shaughnessy, D., and Farhat, A. (1999). "Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition". In: *IEEE Transactions on speech and audio processing* 7.5, pp. 525–532 (cit. on p. 1).
- Wang, C., Wang, X., Li, Y., Xia, Z., and Zhang, C. (2018). "Quaternion polar harmonic Fourier moments for color images". In: *Information Sciences* 450, pp. 141–156 (cit. on pp. 81, 113).
- Wang, C., Wang, X., Xia, Z., and Zhang, C. (2019a). "Ternary radial harmonic Fourier moments based robust stereo image zero-watermarking algorithm". In: *Information Sciences* 470, pp. 109–120 (cit. on pp. 81, 113).
- Wang, X., Chen, J., and Gu, Y. (2015a). "Generalized graph signal sampling and reconstruction". In: *Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on*. IEEE, pp. 567–571 (cit. on p. 2).
- Wang, X.-Y., Yang, L., Liu, R., and Kadir, A. (2010). "A chaotic image encryption algorithm based on perceptron model". In: *Nonlinear Dynamics* 62.3, pp. 615–621 (cit. on p. 64).
- Wang, X.-Y., Zhang, Y.-Q., and Bao, X.-M. (2015b). "A novel chaotic image encryption scheme using DNA sequence operations". In: *Optics and Lasers in Engineering* 73, pp. 53–61 (cit. on p. 77).
- Wang, X., Feng, L., and Zhao, H. (2019b). "Fast image encryption algorithm based on parallel computing system". In: *Information Sciences* 486, pp. 340–358 (cit. on p. 64).
- Wang, X. and Gao, S. (2020). "Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory". In: *Information Sciences* 507, pp. 16–36 (cit. on p. 64).
- Wang, X., Liu, L., and Zhang, Y. (2015c). "A novel chaotic block image encryption algorithm based on dynamic random growth technique". In: *Optics and Lasers in Engineering* 66, pp. 10–18 (cit. on p. 77).
- Wang, X., Teng, L., and Qin, X. (2012). "A novel colour image encryption algorithm based on chaos". In: *Signal Processing* 92.4, pp. 1101–1108 (cit. on p. 76).
- Ward, J. P. (2012). *Quaternions and Cayley numbers: Algebra and applications*. Vol. 403. Springer Science & Business Media (cit. on p. 13).
- Wei, D. and Li, Y. (2013). "Different forms of Plancherel theorem for fractional quaternion Fourier transform". In: *Optik* 124.24, pp. 6999–7002 (cit. on p. 63).

- Xiao, Z., Fang, H., and Wang, X. (Mar. 2021). “Anomalous IoT Sensor Data Detection: An Efficient Approach Enabled by Nonlinear Frequency-Domain Graph Analysis”. In: *IEEE Internet of Things Journal* 8.5, pp. 3812–3821 (cit. on p. 43).
- Xu, D., Jahanchahi, C., Took, C. C., and Mandic, D. P. (2015). “Enabling quaternion derivatives: the generalized HR calculus”. In: *Royal Society open science* 2.8, p. 150255 (cit. on p. 96).
- Yu, L., Feng, L., Chen, C., Qiu, T., Li, L., and Wu, J. (2016). “A Novel Multi-Feature Representation of Images for Heterogeneous IoTs”. In: *IEEE Access* 4. DOI: 10.1109/ACCESS.2016.2607841, pp. 6204–6215 (cit. on p. 26).
- Zhang, C., Florêncio, D., and Loop, C. (2014). “Point cloud attribute compression with graph transform”. In: *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 2066–2070. DOI: 10.1109/ICIP.2014.7025414 (cit. on p. 43).
- Zhang, F. and Hancock, E. R. (Nov. 2008). “Graph spectral image smoothing using the heat kernel”. In: *Pattern Recognition* 41.11, pp. 3328–3342 (cit. on p. 48).
- Zhang, F. (1997). “Quaternions and matrices of quaternions”. In: *Linear algebra and its applications* 251, pp. 21–57 (cit. on pp. 13, 15–17, 19, 22, 90, 115).
- Zhang, Q., Tan, V. Y. F., and Suh, C. (2021). “Community Detection and Matrix Completion With Social and Item Similarity Graphs”. In: *IEEE Transactions on Signal Processing* 69, pp. 917–931 (cit. on p. 43).
- Zhang, S., Tay, Y., Yao, L., and Liu, Q. (2019). “Quaternion knowledge graph embeddings”. In: *Advances in Neural Information Processing Systems*, pp. 2735–2745 (cit. on p. 2).
- Zhang, S., Cui, S., and Ding, Z. (2020). “Hypergraph Spectral Clustering for Point Cloud Segmentation”. In: *IEEE Signal Processing Letters* 27, pp. 1655–1659. DOI: 10.1109/LSP.2020.3023587 (cit. on p. 43).
- Zhang, Y.-Q. and Wang, X.-Y. (2014). “A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice”. In: *Information Sciences* 273, pp. 329–351 (cit. on p. 77).
- Zhang, Y.-Q. and Wang, X.-Y. (2015). “A new image encryption algorithm based on non-adjacent coupled map lattices”. In: *Applied Soft Computing* 26, pp. 10–20 (cit. on p. 77).



