

POLITECNICO DI MILANO  
SCHOOL OF INDUSTRIAL AND INFORMATION  
ENGINEERING  
DEPARTMENT OF MATHEMATICS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE  
LAUSANNE  
SCHOOL OF BASIC SCIENCES  
DEPARTMENT OF MATHEMATICS



**POLITECNICO**  
MILANO 1863

**EPFL**

MASTER THESIS IN COMPUTATIONAL SCIENCE AND  
ENGINEERING

---

# Graph Laplacians on the Sphere for Rotation Equivariant Neural Networks

---

Supervisors:    Michaël Defferrard  
                      prof. Pierre Vandergheynst  
                      prof. Piercesare Secchi  
Co-supervisor: Ph.D. Nathanaël Perraudin  
  
Candidate:        Martino Milani

Academic Year 2018/2019

## Abstract

A fundamental problem in signal processing is to design computationally efficient algorithms to filter signals. In many applications, the signals to filter lie on a sphere. Meaningful examples of data of this kind are weather data on the Earth, or images of the sky. It is then important to design filtering algorithms that are computationally efficient and capable of exploiting the rotational symmetry of the problem. In these applications, given a continuous signal  $f : \mathbb{S}^2 \rightarrow \mathbb{R}$  on a 2-sphere  $\mathbb{S}^2 \subset \mathbb{R}^3$ , we can only know the vector of its sampled values  $\mathbf{f} \in \mathbb{R}^N : (\mathbf{f})_i = f(\mathbf{x}_i)$  in a finite set of points  $\mathcal{P} \subset \mathbb{S}^2$ ,  $\mathcal{P} = \{\mathbf{x}_i\}_{i=0}^{n-1}$  where our sensors are located. Perraudin et al. in [18] construct a sparse graph  $G$  on the vertex set  $\mathcal{P}$  and then use a polynomial of the corresponding graph Laplacian matrix  $\mathbf{L} \in \mathbb{R}^{n \times n}$  to perform a computationally efficient -  $\mathcal{O}(n)$  - filtering of the sampled signal  $\mathbf{f}$ . In order to study how well this algorithm respects the symmetry of the problem - i.e it is equivariant to the rotation group  $\text{SO}(3)$  - it is important to guarantee that the spectrum of  $\mathbf{L}$  and spectrum of the Laplace-Beltrami operator  $\Delta_{\mathbb{S}}^2$  are somewhat “close”.

We study the spectral properties of such graph Laplacian matrix in the special case of [18] where the sampling  $\mathcal{P}$  is the so called HEALPix sampling (acronym for **H**ierarchical **E**qual **A**rea iso**L**atitude **P**ixelization) and we show a way to build a graph  $G'$  such that the corresponding graph Laplacian matrix  $\mathbf{L}'$  shows better spectral properties than the one presented in [18].

We investigate other different methods of building the matrix  $\mathbf{L}$  better suited to non uniform sampling measures. In particular, we studied the Finite Element Method approximation of the Laplace-Beltrami operator on the sphere, and how FEM filtering relates to graph filtering, showing the importance of non symmetric discrete Laplacians when it comes to non uniform sampling measures. We finish by showing how the graph Laplacian  $\mathbf{L}'$  proposed in this work improved the performances of DeepSphere in a well known classification task using different sampling schemes of the sphere, and by comparing the different Discrete Laplacians introduced in this work.

## ACKNOWLEDGEMENT

I would first like to thank my thesis advisor Michaël Defferrard of the School of Electrical Engineering at EPF Lausanne. He was always there whenever I had problems or had any question. He let this work to be of my own, but he was always there to guide me in the right direction whenever it was needed. His enthusiasm and passion about research were contagious, and working with him was a real pleasure.

I would like to thank Prof. Pierre Vandergheynst of the Institute of Electrical Engineering at EPF Lausanne and Prof. Piercesare Secchi of the Department of Mathematics at Politecnico di Milano, who with their help contributed significantly to this work. Thanks to Ph.D. Nathanaël Perraudin of the Swiss Data Science Center at ETH Zürich for having co-supervised my work. His intuition helped me a lot and with the numerous meetings we had he gave to this work a very valuable contribution. Thanks to Prof. Fabio Nobile of the Department of Mathematics at EPF Lausanne, who helped me in choosing the right direction to take when I needed it.

I would also like to thank Frédéric Gusset and Charles Gallay for all the interesting discussions we had in our office at the Signal Processing Laboratory at EPF Lausanne, and Luca Zampieri, for offering me a coffee whenever I needed it and for always motivating me with his incredible hard working attitude.

I then must express my deepest gratitude to my family. To my parents for having taught me to always work hard and for their unconditional support, and to my brother, for his love and for having always been by my side. This accomplishment would not have been possible without them. Thank you.

Finally, it is a pleasure to thank all my friends, those who are near and those who are far.

# Contents

<b>1</b>	<b>Introduction and general background</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	Fourier Transforms and Convolutions on the 2-Sphere . . . . .	3
1.3	Spherical Convolutional Neural Networks . . . . .	7
1.4	Spectral Graph Theory . . . . .	7
1.5	The Equivariance error for graph convolutions . . . . .	11
<b>2</b>	<b>Discrete Laplacians</b>	<b>14</b>
2.1	Notions of Differential Geometry . . . . .	14
2.2	Discrete Laplacians from Differential Geometry . . . . .	15
2.3	Linear Finite Element Method Laplacian . . . . .	16
2.4	Graph Laplacian for manifolds . . . . .	17
<b>3</b>	<b>Graph Spherical Convolutions</b>	<b>20</b>
3.1	Graph Spherical Convolutional Neural Networks . . . . .	20
3.2	Pointwise convergence of the Heat Kernel Graph Laplacian on the Sphere . . . . .	23
3.2.1	Proof of the pointwise convergence of the Heat Kernel Graph Laplacian on the Sphere for an equi area sampling scheme . . . . .	27
3.3	How to build a good graph to approximate spherical convolutions	31
3.3.1	Full graph, $n \rightarrow \infty$ . . . . .	32
3.3.2	Full graph, $t \rightarrow 0$ . . . . .	34
3.3.3	Putting it together: full graph, $n \rightarrow \infty$ and $t \rightarrow 0$ . . .	36
3.3.4	Reducing the number of neighbors . . . . .	36
3.3.5	Equivariance error . . . . .	40
<b>4</b>	<b>Non uniform sampling schemes and the FEM Laplacian</b>	<b>44</b>
4.1	Graph Laplacian on the Equiangular Sampling . . . . .	44
4.1.1	The Equiangular Sampling . . . . .	44
4.1.2	Heat Kernel Graph Laplacian on the equiangular sam- pling scheme . . . . .	46
4.1.3	A graph alternative to the HKGL for the equiangular sampling . . . . .	46



4.2	The Finite Element Method approximation of the Laplace-Beltrami operator on the sphere . . . . .	49
4.3	How to filter a signal with the linear FEM . . . . .	51
4.3.1	A confront between FEM filtering and graph filtering .	52
<b>5</b>	<b>Conclusions</b>	<b>58</b>
5.1	Experimental validation: SHREC17 . . . . .	58
5.2	Confront of different Discrete Laplacians on the equiangular sampling . . . . .	59
5.3	Final considerations and future work . . . . .	59
<b>6</b>	<b>Appendix</b>	<b>65</b>
6.0.1	Weak formulation of a PDE and Galerkin Method . . .	66
6.0.2	The Finite Element Method . . . . .	69

# 1 Introduction and general background

## 1.1 Introduction

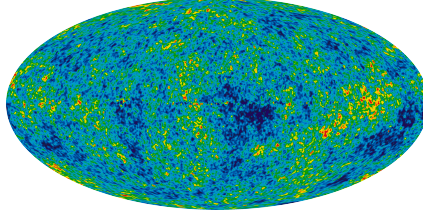
Neural Networks (NNs) are popular algorithms for regression and classification tasks. Taking as example an image classification problem, a neural network perform multiple combinations of linear and non-linear transformations of each image  $I$  to assign it a label  $C_I$  chosen in the set of all the possible labels  $\mathcal{C}$ . The first *layer* of the neural network transforms the input image  $I$  in a vector - called *feature map* -  $\mathbf{f}_1$  through a function  $\phi_1$ . The output feature map of the first layer is used as input of the second layer that transforms it through a function  $\phi_2$ , and so on, until the original image  $I$  is mapped into a label  $C_I$  by the last,  $n$ -th layer of the neural network:

$$C_I = \phi_n \circ \phi_{n-1} \circ \dots \phi_2 \circ \phi_1(I)$$

With a large *training set* of pre labeled images at its disposal, a NN is capable of learning the optimal transformations  $\phi_i$  that let it map each input image to its correct label. Since the functions  $\phi_i$  have many degrees of freedom - even millions - a neural network is able to learn very complex transformations. In the work of Csáji [5], NNs have been proved to be universal function approximators, meaning that with a sufficient number of parameters NNs are able to approximate any continuous function on a compact domain. This makes NNs the optimal tool for complex tasks such as image classification, image segmentation, speech recognition and natural language processing.

Convolutional Neural Networks (CNNs) are a subset of NNs whose layer structure has been specifically designed for image recognition and segmentation. For this purpose, they don't have all the degrees of freedom of a *fully connected* neural network: each layer is constrained to learn only those transformations of the input that are *equivariant* to translations of the input. This means that a translation of the input image will not result in a change of class. The layers  $\phi_i$  of a CNN are *convolutions* with some kernels  $k_i$ , that were learned during the training phase. Thanks to their design, training of CNNs is faster - thanks to the smaller number of parameters to be learned compared to a fully connected NN -, easier - since there's no need of artificially *augmenting* the

Figure 1: Cosmic microwave background map, the oldest electromagnetic radiation in the universe. Source: Wikipedia



dataset with translated copies of the same image -, and leads to very accurate results [4], [8].

Spherical convolutional neural networks (SCNNs) are CNNs that have been designed to deal with spherical data, whose layer design makes them equivariant to *rotations* of the input. Examples of tasks where data is naturally represented on a sphere are (i) climate science, where data is sampled on the surface of the Earth, (ii) cosmology, where observations are naturally projected on a sphere centered around the observer (see Figure 1), and (iii) virtual reality, where the images are represented on a sphere centered around the player. Being able to come up with rotation equivariant architectures brings with it all the advantages that traditional CNNs have brought for traditional (euclidean) image classification tasks: training is faster, easier and results are very accurate. Each layer of a SCNN performs a *spherical convolution* of the input feature map with a kernel  $k_i$  learned during the training phase. One of the main issues with traditional SCNNs is the computational complexity of computing at each layer the Spherical Harmonic Transform of the data to perform the convolution. To overcome this issue, Perraudin et al. [18] proposed a Graph Convolutional Neural Network (GCNN) that is almost equivariant to rotations, replacing the SHT with a more efficient Graph Convolution.

This work is organized as follows: in Chapter 1 we start by presenting fundamental concepts of spectral theory on the sphere and we present classical ways of building rotation equivariant neural networks through the use of the classical SHT. We present then some basics of Spectral Graph Theory that lay the foundations of Graph Convolutional Neural Networks. In Chapter 2 we present the general framework of how to discretize the Laplace-Beltrami operator on a general manifold, concluding with the special case of the Heat Kernel Graph Laplacian (HKGL) approximation, together with some conver-

gence results. We continue in Chapter 3 by introducing DeepSphere [18], a Graph Spherical Convolutional Neural Network (GCNN) that uses a graph Laplacian matrix  $\mathbf{L}$  similar to the HKGL to perform graph convolutions that are almost equivariant to rotations. We study the spectral properties and the equivariance error of DeepSphere and we show a way to build a graph  $G'$  such that the corresponding graph Laplacian matrix  $\mathbf{L}'$  shows better spectral and equivariance properties. In Chapter 4 we show better graph constructions than the HKGL on non uniform sampling measures. To conclude, we show a different approach to perform rotation invariant convolutions that uses the Finite Element Method (FEM) approximation of the Laplace-Beltrami operator on the sphere. Chapter 5 concludes this work by presenting some experimental results obtained by Gusset et al. [11] that implemented the graph  $G'$  on a GCNN and compared its performances to DeepSphere on a well known dataset [22] showing that the new graph  $G'$  performs better in real applications. We finish by comparing the FEM and the graph approach, discussing the general problem of how to incorporate geometrical informations about the sphere in the graph.

## 1.2 Fourier Transforms and Convolutions on the 2-Sphere

The goal of this section is to present to the reader some fundamental results of spectral theory on the sphere that we will need in this work. We present a brief review of Banach and Hilbert spaces, spherical harmonics, Fourier transform and convolution on  $\mathbb{S}^2$ . We refer to Sections 2 and 3 of the work of Driscoll and Healy [7] for a more detailed and effective review of spectral theory on the Sphere.

**Banach and Hilbert spaces.** A *norm*  $\|\cdot\| : X \rightarrow \mathbb{R}$  on a vector space  $X$  is a subadditive, positive definite function such that  $\|x + y\| \leq \|x\| + \|y\|$ ,  $\forall x, y \in X$  (triangle inequality). A *Cauchy sequence*  $(x_n) \subset X$  is a sequence such that  $\forall \epsilon > 0 \exists M > 0 : \forall i, j > M \ \|x_i - x_j\| < \epsilon$ . A *Banach space*  $(X, \|\cdot\|)$  is a normed vector space on the scalar field  $F$  that is *complete*, meaning that  $X$  is "big enough" such that for every Cauchy sequence  $(x_n) \subset X$  there exist a  $x \in X$  such that  $x$  is the limit of  $(x_n)$  in  $X$  i.e.  $\|x_n - x\| \rightarrow 0$ . A *basis* of  $(X, \|\cdot\|)$  is a minimal set of linearly independent vectors  $\mathcal{B} \subset X$  such that

every element of  $X$  can be written as linear combination of the elements of  $\mathcal{B}$ . A scalar product is a function  $\langle \cdot, \cdot \rangle : X \times X \rightarrow \mathbb{F}$  that is linear in the first argument, positive definite and conjugate symmetric. Through a scalar product we can define the notion of angle  $\theta$  between two elements  $x, y \in X$  through the following formula:

$$\cos \theta = \frac{\langle x, y \rangle}{\|x\| \|y\|}$$

In particular we can define the notion of orthogonality: two elements  $x, y \in X$  are orthogonal if and only if  $\langle x, y \rangle = 0$ . We are interested in those particular Banach spaces where we can define a notion of orthogonality between vectors. A Banach space  $(X, \|\cdot\|)$  is a *Hilbert space* when the norm  $\|\cdot\|$  can be induced by a *scalar product*:  $\|\cdot\| = \sqrt{\langle \cdot, \cdot \rangle}$ . We can now define an *orthonormal* basis of  $X$ : a basis  $\mathcal{B} \subset X$  such that  $\forall x, y \in \mathcal{B}, \|x\| = \|y\| = 1$  and  $\langle x, y \rangle = 0$ . Given an orthonormal basis  $\mathcal{B} = \{b_i\}_{i \in I}$  we can write each vector in its *Fourier series*

$$x = \sum_{i \in I} \langle x, b_i \rangle b_i \quad (1)$$

If the set  $I$  is countable the Hilbert space  $(X, \|\cdot\|)$  is called *separable*. Having a countable orthonormal basis, and thus the possibility of representing each vector through its Fourier series enormously simplifies many problems.

**Spherical Harmonics.** Given the usual parametrization  $x = x(\theta, \phi), \theta \in [0, \pi], \phi \in [0, 2\pi]$  of the sphere

$$\begin{aligned} \mathbb{S}^2 &= \left\{ \omega = (\omega_1, \omega_2, \omega_3) \in \mathbb{R}^3 : \|\omega\|_{\mathbb{R}^3} = (\omega_1^2 + \omega_2^2 + \omega_3^2)^{1/2} = 1 \right\} \\ \omega_1 &= \cos(\phi) \sin(\theta), \quad \omega_2 = \sin(\phi) \sin(\theta), \quad \omega_3 = \cos(\theta) \end{aligned}$$

the Hilbert space  $L^2(\mathbb{S}^2)$  is defined as the space of square-integrable functions endowed with the scalar product  $\langle f, g \rangle = \int_{\mathbb{S}^2} f(\omega) \bar{g}(\omega) d\omega$  where the measure  $d\omega$  is the rotation-invariant measure such that

$$\int_{\omega \in \mathbb{S}^2} f(\omega) d\omega = \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} f(\omega(\theta, \phi)) \sin \theta d\theta d\phi \quad (2)$$

$$\int_{\omega \in \mathbb{S}^2} f(g\omega) d\omega = \int_{\omega \in \mathbb{S}^2} f(\omega) d\omega, \quad g \in SO(3) \quad (3)$$

For each rotation  $g \in SO(3)$  we define a corresponding rotation operator  $\Lambda(g)$  by

$$\Lambda(g)f(\omega) = f(g^{-1}\omega) \quad (4)$$

A space is invariant under the rotations  $g$  in  $SO(3)$  if all operators  $\Lambda(g)$  take each function of the space back into the space. As very well written by Driscoll et al [7]:

*Fourier analysis on the sphere amounts to the decomposition of the space of square integrable functions on  $\mathbb{S}^2$  in minimal subspaces  $V_\ell$  invariant under all of the rotations in  $SO(3)$ , thus simplifying the analysis of rotation-invariant operators.*

It's a well known fact [7] that the  $\ell$ -th invariant subspace  $V_\ell \subset L^2(\mathbb{S}^2)$  is made of polynomials of  $\mathbb{R}^3$  of degree  $\ell$  restricted to  $\mathbb{S}^2$ , and has dimension  $2\ell + 1$ . Its elements are called *spherical harmonics* of degree  $\ell$ . These subspaces are orthogonal between them, and correspond to the eigenspaces of the Laplace-Beltrami operator  $\Delta_{\mathbb{S}^2}$ . For a thorough introduction to how to define the Laplace-Beltrami operator on a manifold and its properties, see [21]. The set of all the orthonormal basis  $Y_\ell^m$ ,  $-\ell \leq m \leq \ell$  of each subspace  $V_\ell$  gives an orthonormal basis of  $L^2(\mathbb{S}^2)$ . The analytical expression of the spherical harmonic  $Y_\ell^m(\theta, \phi)$  is actually known [7]:

$$Y_\ell^m(\theta, \phi) = (-1)^m \sqrt{\frac{(2\ell+1)(\ell-m)!}{4\pi(\ell+m)!}} P_\ell^m(\cos \theta) e^{im\phi} \quad (5)$$

where  $P_\ell^m$  are the *Legendre functions* as defined in [7].

*Remark.* Saying that the space  $V_\ell$  is invariant under rotations  $SO(3)$  means that under any rotation  $g \in SO(3)$ , any spherical harmonic  $Y_\ell^m \in V_\ell$  is transformed into a linear combination of the others spherical harmonics of the same degree  $\ell$ :

$$\Lambda(g)Y_\ell^m(\omega) = \sum_{|k| \leq \ell} Y_\ell^k(\omega) \alpha_{k,m}^{(\ell)}(g).$$

**Fourier transform.** We can now expand each function  $f \in L^2(\mathbb{S}^2)$  in the orthonormal coordinate system given by the spherical harmonics

$$f(\omega) = \sum_{\ell \in \mathbb{N}} \sum_{|m| \leq \ell} \hat{f}(\ell, m) Y_\ell^m(\omega) \quad (6)$$

$$\hat{f}(\ell, m) = \int_{\omega \in \mathbb{S}^2} f(\omega) Y_\ell^m(\omega) d\omega \quad (7)$$

where the coefficients  $\hat{f}(\ell, m)$  are the *Fourier coefficients* of  $f$ . The computation of  $\hat{f}(\ell, m)$  is called Spherical Harmonic Transform (SHT). Thanks to equation (5) we can decompose the computation of the SHT (7) in the two directions  $(\theta, \phi)$ . One reason for which the most popular sampling schemes of the sphere have the pixels lie on isolatitude circles is that it is possible to use standard one-dimensional FFT algorithms to compute the longitudinal part of the transform, making the computation of the SHT  $\mathcal{O}(n^{3/2})$ , where  $n$  is the number of pixels [10].

**Convolutions.** Convolution on the sphere is profoundly different than convolution on the Euclidean plane  $\mathbb{R}^2$ . Since translations are isomorphic to  $\mathbb{R}^2$ , the convolution  $f * g(x)$  of two functions  $f, g \in L^2(\mathbb{R}^2)$  is itself a function on the plane:

$$\int_{\mathbb{R}^2} f(y) g(x - y) dy = f * g(x) : \quad \mathbb{R}^2 \rightarrow \mathbb{R} \quad (8)$$

On the sphere things work differently: translations are replaced by rotations, but due to the fact that  $SO(3)$  is not isomorphic to  $\mathbb{S}^2$ , if we define the convolution on  $\mathbb{S}^2$  as follows:

$$f * k(g) := \int_{\eta \in \mathbb{S}^2} \Lambda(g) k(\eta) f(\eta) d\eta = \int_{\eta \in \mathbb{S}^2} k(g^{-1}\eta) f(\eta) d\eta \quad (9)$$

$f * k(g) : SO(3) \rightarrow \mathbb{R}$  is not a function of the sphere anymore, but it is a function of the special rotation group  $SO(3)$ . In section 1.3 we will explain how Cohen et al. [4] use in their work this definition of convolution on the sphere to construct a rotation equivariant NN. However, the definition of convolution that we will use in this work is the following, where the integral is performed not on the sphere but on the rotation group  $SO(3)$ :

$$k * f(\omega) = \int_{g \in SO(3)} k(g\eta) f(g^{-1}\omega) dg \quad (10)$$

where  $dg$  is the measure on  $SO(3)$  that can be written in terms of the three Euler angles  $(\theta, \phi, \psi)$

$$dg = \sin \theta d\theta d\phi d\psi$$

In this way  $k * f(\omega)$  is still a function defined on  $\mathbb{S}^2$ . However, integrating on  $SO(3)$  means integrating on the third Euler angle  $\psi$ , that in practice means using definition (9) with the use of *radial* kernels  $k$  only. Using the convolution defined in equation (10), the following theorem [7] generalizes on the sphere a well known property of convolutions and Fourier transforms:

*Theorem 1.1.* Given two functions  $f, h$  in  $L^2(\mathbb{S}^2)$ , the Fourier transform of the convolution is a pointwise product of the transforms

$$(f \hat{*} h)(\ell, m) = 2\pi \sqrt{\frac{4\pi}{2\ell + 1}} \hat{f}(\ell, m) \hat{h}(\ell, 0).$$

### 1.3 Spherical Convolutional Neural Networks

Cohen et al. [4] proposed a NN where the first layer performs a convolution on the sphere as defined by equation (9). The output feature map - a signal on  $SO(3)$  - is processed by the deeper layers that perform other convolutions in  $SO(3)$ . All these convolutions are performed in the spectral domain as in theorem 1.1, meaning that every signal has to be Fourier-transformed first, at each forward and backward step of the training phase of the NN. This approach, even with the use of Generalized FFT algorithms for  $\mathbb{S}^2$  and  $SO(3)$ , remains both computationally expensive ( $\mathcal{O}(n^{3/2})$ ) and memory expensive, due to the need of storing kernels defined on the much bigger space  $SO(3)$ . In section 5.1 we report in table (4) the results of Gusset et al. [11], that compared both the training and inference time of Cohen’s SCNN, showing how slow this architecture is compared to other rotation equivariant NNs.

### 1.4 Spectral Graph Theory

**Graphs.** For the purposes of this work, a *weighted undirected graph*  $G(V, E, \mathbf{W})$  is defined by a vertex set  $V$ , an edge set  $E$ , where the edges are unordered



pairs of vertices, and the matrix  $\mathbf{W}$  whose entries  $w_{ij}$  represent the weight of the edge  $(v_i, v_j)$ .  $G$  is a *simple* graph, if  $w_{ij}$  assume only values in  $\{0, 1\}$ . Undirected graphs are common mathematical objects used to model simple, symmetric relationships between things. The edge  $e_{ij} = (v_i, v_j) \in E$  is the mathematical object that translates the fact that the vertices  $v_i, v_j$  are in a relationship, and the weight  $w_{ij}$  measures how strong this relationship is. Common examples of graphs include friendship graphs, where people are the vertices and the edges represent friendship, or electric network graphs, where vertices represent electronic components and edges represent wires.

**The graph Laplacian.** If  $\mathbf{D}$  is the diagonal matrix  $\mathbf{D}_{ii} = \sum_j w_{ij}$ , one can define [24] the combinatorial graph Laplacian  $\mathbf{L}$

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \quad (11)$$

and the symmetric normalized graph Laplacian  $\mathbf{L}'$

$$\mathbf{L}' = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2} \quad (12)$$

In a simple friendship graph  $G$ , one can define a vector  $\mathbf{f}$  such that each entry  $f_i$  is the age of the person associated with the vertex  $v_i$ , and could try to measure how much people tend to be friends with people of the same age. In other words, how smooth the signal  $\mathbf{f}$  is on the graph  $G$ . A good measure for the smoothness of a signal on a graph is given by the *Dirichlet energy* of the signal  $\mathbf{f}$ , i.e., the quadratic form associated with the normalized Laplace operator  $\mathbf{L}'$ :

$$\mathbf{f}^\top \mathbf{L}' \mathbf{f} = \sum_{(v_j, v_k) \in E} \frac{w_{jk}}{\sqrt{d_j d_k}} (f_j - f_k)^2 \quad (13)$$

The reason why the Dirichlet energy is a good measure of the smoothness of  $\mathbf{f}$  is easier to understand in the case of a simple graph, where it reduces to the sum

$$\mathbf{f}^\top \mathbf{L}' \mathbf{f} = \sum_{(v_j, v_k) \in E} (f_j - f_k)^2. \quad (14)$$

that will grow for each edge  $(v_i, v_j)$  connecting people with very different age. Although the Dirichlet energy (13) works also for the combinatorial graph Laplacian  $\mathbf{L}$ , in practice it is preferred to use the symmetric normalized Lapla-

cian when the degree distribution is wide. Another way of looking at equation (14) is as the following: the differences  $f_j - f_k$  can be seen as the *gradient*  $\nabla \mathbf{f}$  that is a signal on the edges  $(v_j, v_k)$  and equation (14) as the quadratic norm of such gradient  $\|\nabla \mathbf{f}\|^2$ .

**Graph Fourier transform** Since the graph Laplacian is a symmetric matrix, we can write its eigen decomposition [25]

$$\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$$

where  $\mathbf{V}$  is the orthonormal basis of  $\mathbb{R}^n$  of eigenvectors, and  $\mathbf{\Lambda}$  the real diagonal matrix of the eigenvalues  $\mathbf{\Lambda} = \text{diag}(\lambda_i \in \mathbb{R})$ . Similarly to the continuous domain, where the Fourier transform of a signal  $f$  is defined as the projection of  $f$  on the orthonormal eigenbasis of the Laplace-Beltrami operator  $\Delta$ , on a graph we can define a graph Fourier transform  $\mathcal{F}_G : \mathbb{R}^n \rightarrow \mathbb{R}^n$  of a discrete signal  $\mathbf{f} \in \mathbb{R}^n$  as the projection of  $\mathbf{f}$  on the eigenvectors of the graph Laplacian  $\mathbf{L}$ :

$$\mathcal{F}_G(\mathbf{f}) := \mathbf{V}^\top \mathbf{f} = \hat{\mathbf{f}} \quad (15)$$

The inverse graph Fourier transform  $\mathcal{F}_G^{-1}$  is thus

$$\mathcal{F}_G^{-1}(\hat{\mathbf{f}}) := \mathbf{V}\hat{\mathbf{f}} = \mathbf{V}\mathbf{V}^\top \mathbf{f} = \mathbf{f} \quad (16)$$

In the continuous case, the eigenvalues of the Laplace-Beltrami operator are associated with a notion of *frequency* of the corresponding eigenfunction. In a graph we have a similar notion: define the Rayleigh quotient of a vector  $\mathbf{v} \in \mathbb{R}^n$  to be

$$\mathcal{R}(\mathbf{v}) = \frac{\mathbf{v}^\top \mathbf{L} \mathbf{v}}{\mathbf{v}^\top \mathbf{v}} \quad (17)$$

The well known [25] Courant-Fischer characterization of eigenvalues and eigenvectors of symmetric matrices (18) can be interpreted in light of what we wrote about the interpretation of the Dirichlet energy (13)  $\mathbf{v}^\top \mathbf{L} \mathbf{v}$  as a measure of the smoothness of  $\mathbf{v}$ . The eigenvalue  $\lambda_i$  is the measure of smoothness of the eigenvector  $\mathbf{v}_i$ , that is the smoothest vector perpendicular to the lower-degree

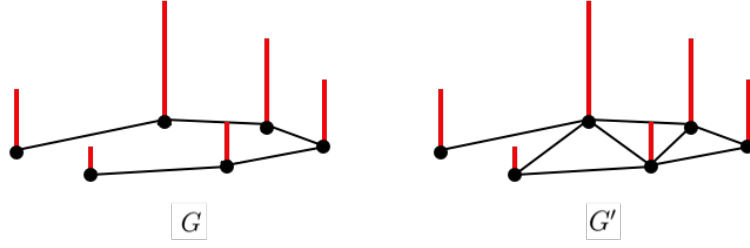


Figure 2: Different graph topologies can drastically change the measure of smoothness of a signal  $\mathbf{f}$ , here represented as red vertical bars on the vertices.

eigenvectors  $\mathbf{v}_1, \dots, \mathbf{v}_{i-1}$ .

$$\begin{aligned}
 \lambda_1 &= \min_{\mathbf{v} \neq 0} \mathcal{R}(v) \\
 \mathbf{v}_1 &= \operatorname{argmin}_{\|\mathbf{v}\|=1, \mathbf{v} \neq 0} \mathcal{R}(v) \\
 \begin{cases} \lambda_i = \min_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_1, \dots, \mathbf{v}_{i-1}} \mathcal{R}(v) \\ \mathbf{v}_i = \operatorname{arg} \min_{\|\mathbf{v}\|=1, \mathbf{v} \perp \mathbf{v}_1, \dots, \mathbf{v}_{i-1}} \mathcal{R}(v) \end{cases}
 \end{aligned} \tag{18}$$

*Remark.* It is interesting to notice that the Dirichlet energy of a signal  $\mathbf{f}$  on a graph  $G$  could change drastically by changing the underlying topology of  $G$ . In figure 2 we see two simple graphs  $G = (V, E)$ ,  $G' = (V, E')$  with the same signal  $\mathbf{f}$  represented as vertical red bars over the vertex set  $V$ . On  $G$ , the signal  $\mathbf{f}$  varies smoothly across the graph since the edges  $V$  connect only those vertices with similar values of  $\mathbf{f}$ . On  $G'$ , since we added edges between vertices with very different values of  $\mathbf{f}$ , we will have that the Dirichlet energy of  $\mathbf{f}$  calculated on the graph  $G'$  will be much higher than the one calculated on the graph  $G$ .

**Convolution and filtering on graphs** As the plane  $\mathbb{R}^2$  is symmetric with respect to any translation, and the sphere  $\mathbb{S}^2$  is symmetric with respect to any rotation, the respective definitions of convolution are equivariant respectively to these two symmetry groups. Since there are no such global symmetries in a general graph  $G$ , definitions (8), (10) can not be extended naturally on graphs. However, both on  $\mathbb{R}^2$  and on  $\mathbb{S}^2$ , the convolution of a signal  $f$  with a kernel  $k$  can be performed in the spectral domain by multiplying the transformed

signal  $\hat{f}$  times the transformed kernel  $\hat{k}$ :

$$f * k = \mathcal{F}^{-1}(\hat{f} \cdot \hat{k}) \quad (19)$$

In a similar way we can define a notion of convolution also in the graph spectral domain. We use the following definition [24] of the convolution of a signal  $\mathbf{f}$  times a kernel  $\mathbf{K}$ :

$$\Omega_{\mathbf{K}}\mathbf{f} = \mathcal{F}_G^{-1}(\mathbf{K}\hat{\mathbf{f}}) = \mathbf{V}\mathbf{K}\mathbf{V}^T\mathbf{f} \quad (20)$$

where  $\mathbf{K}$  is a *diagonal matrix*  $K_{ii} = k_i$ . Graphs convolutions are different from the ones we are used to define in Euclidean domains, since the graph kernels  $\mathbf{K}$  are diagonal matrices that can not be thought as the Fourier transform of a corresponding kernel defined in the spatial (vertex) domain. The diagonal elements  $k_i$  can be thought as functions of the corresponding eigenvalues

$$k_i = k(\lambda_i)$$

thus providing an intuitive frequency interpretation of the kernel  $\mathbf{K}$ . In this way the convolution can be seen as a *filtering* operation: for example, a kernel  $k(\lambda_i) = \exp(-\lambda_i)$  will be the kernel of a low-pass filter since it will cut the high frequencies.

## 1.5 The Equivariance error for graph convolutions

Take a sampling scheme  $V = \{v_i \in \mathbb{S}^2, i = 0, \dots, n\}$  of the sphere, a weighted undirected graph  $G(V, E, \mathbf{W})$ , a signal  $f : \mathbb{S}^2 \rightarrow \mathbb{R}$  and its sampled representation  $\mathbf{f} : f_i = f(v_i)$ . Suppose that there exists a sampling operator  $T_V : L^2(\mathbb{S}^2) \supset F \rightarrow \mathbb{R}^n$ ,  $T_V(f) = \mathbf{f}$  defined on a suitable subspace  $F$  of  $L^2(\mathbb{S}^2)$  such that it is invertible, i.e., we can unambiguously reconstruct the function  $f \in F$  from its sampled values  $\mathbf{f}$ . The existence of such subspace depends on the sampling scheme  $V$ , and its characterization is a common problem in signal processing [7]. Recall the definition (4) of the rotation operator  $\Lambda(g), g \in SO(3)$ .

We now want to understand how to set the edges and the weights of  $G$  such that

$$T\Lambda(g)T^{-1}\Omega_k T f = \Omega_k T\Lambda(g)f \quad (21)$$

i.e., the graph convolution  $\Omega_k$  and any rotation  $\Lambda(g)$  commute.

Verifying equation (21) is really hard in practice, due to the fact that for almost all samplings schemes  $V$  it is not known if there exists a space  $F$  in which  $T$  is invertible. A special case is the *equiangular sampling* scheme described in section 4.1 where the sampling theorem 4.1 holds [7]. With all the other sampling schemes, there are no sampling theorems available, but there are implementations of discrete SHT to reconstruct a sampled signal  $\mathbf{f}$ , thus providing a way to approximate  $T^{-1}$ . Thanks to this we are able, for a given sampling, a given function  $f$ , a given rotation  $g$ , and a given kernel  $k$ , to compute the *normalized equivariance error*

$$E_G(f, g) = \left( \frac{\|T\Lambda(g)T^{-1}\Omega_k T f - \Omega_k T\Lambda(g)f\|_{L^2(\mathbb{R}^2)}}{\|T f\|_{L^2(\mathbb{R}^2)}} \right)^2 \quad (22)$$

where  $T^{-1}$  is substituted with a discrete SHT in case  $T$  is not invertible. A measure of how equivariant a graph is with respect to rotations will then be given by the *mean equivariance error*

$$\overline{E}_G = \mathbb{E}_{f,g} E_G(f, g) \quad (23)$$

In practice the expected value is obtained by averaging over a finite number of random functions and random rotations. The mean equivariance error  $\overline{E}_G$  gives us an indication of how close the graph  $G$  is from being equivariant to rotations. Now we state an intuitive concept that explains how to construct rotation invariant graphs, i.e. graphs such that  $\overline{E}_G$  is small.

The mean equivariance error  $\overline{E}_G$  will be small if the scalar product  $\mathbf{f}^\top \mathbf{v}_{i(\ell, m)}$  well approximates  $\hat{f}(\ell, m)$  i.e., the  $L^2$  scalar product of the continuous signal  $\hat{f}(\ell, m) = \int_{\eta \in \mathbb{S}^2} f(\eta) Y_\ell^m(\eta) d\mu(\eta)$ .

If  $V$  is an equal area sampling scheme, i.e. the area around each pixel  $v_i$  is the same,  $\overline{E}_G$  will be small if the graph Laplacian  $\mathbf{L}$  is such that its eigenvectors  $\mathbf{v}_i$  well approximate the eigenfunctions of the Laplace-Beltrami operator  $\Delta_{\mathbb{S}^2}$  evaluated in the points of the sampling scheme, i.e.,

$$\mathbf{v}_{i(\ell, m)} \cong Y_\ell^m(x_i)$$

In this way we framed the problem of constructing a rotation invariant graph with the more general problem of coming up with a matrix  $\mathbf{L}$  with some specific spectral properties. Graphs are only one of many ways of coming up with such matrix  $\mathbf{L}$ , and many other methods have been already studied in the literature. In the next Chapter we present a brief overview of some of these methods providing the general context in which graph filtering can be framed.

## 2 Discrete Laplacians

In this Chapter we describe different methods to approximate the Laplace-Beltrami operator on manifolds, with particular attention to what we will call the Heat Kernel Graph Laplacian.

Let  $f$  be a sufficiently smooth function on a compact, closed infinitely differentiable manifold  $\mathcal{M}$ . The Laplacian eigenvalue problem on  $\mathcal{M}$  is defined as

$$\Delta_{\mathcal{M}}\psi = -\lambda\psi \quad (24)$$

Being the Laplace-Beltrami operator self-adjoint and semi-positive definite, there exists a basis  $\mathcal{B} = \{\psi_i\}_i$  of the space  $L^2(\mathcal{M})$  such that  $\Delta_{\mathcal{M}}\psi_i = -\lambda_i\psi_i$ ,  $\lambda_0 \leq \lambda_1 \leq \dots, \lambda_i \leq \lambda_{i+1} \dots \leq +\infty$ . See [21] for an introduction to the Laplace-Beltrami operator on manifolds.

There's been a lot of work in trying to calculate solutions to equation (24), leading to different ways to approximate the Laplace-Beltrami operator through what we call a *discrete Laplacian*. By discrete Laplacian we mean an operator  $L$  that, once evaluated on a signal  $f$  and on a vertex  $x_i$  can be written as a matrix  $\mathbf{L}$  in the following way:

$$\mathbf{L}f(\mathbf{x}_i) = \frac{1}{d_i} \sum_j w_{ij} (f(\mathbf{x}_i) - f(\mathbf{x}_j)) \quad (25)$$

Note that with unit masses  $d_i = 1$  we recover the same definition of Graph Laplacian. We need now to introduce some basic concepts of Differential Geometry, especially the definition of mean curvature of a manifold and its link with the Laplace-Beltrami operator.

### 2.1 Notions of Differential Geometry

For this short introduction to basic concepts of Differential Geometry, set the manifold  $\mathcal{M}$  to be a differentiable, two dimensional surface embedded in  $\mathbb{R}^3$ . The *curvature* of a curve on a plane is defined as the inverse of the radius  $R$  of the tangent circle. For each point on the manifold  $\mathcal{M}$ , define its tangent plane  $H$ , orthogonal to the normal vector  $\mathbf{n}$ . For every unit vector  $\mathbf{e}_\theta$  lying on

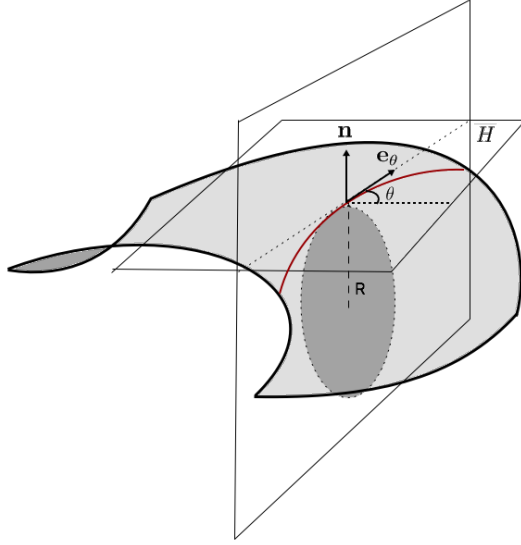


Figure 3: Curvature normal of a manifold

the tangent plane  $H$ , where  $\theta$  is an angle that measures the direction on the tangent plane of  $\mathbf{e}_\theta$ , the *normal curvature*  $\kappa(\theta)$  is defined as the curvature of the curve that is the intersection of the manifold  $\mathcal{M}$  and the plane containing both  $\mathbf{n}$  and  $\mathbf{e}_\theta$ . The *mean curvature*  $\bar{\kappa}$  is defined as the average on  $\theta$  of the normal curvatures:

$$\bar{\kappa} = \frac{1}{2\pi} \int_0^{2\pi} \kappa(\theta) d\theta \quad (26)$$

It can be proved that the Laplace-Beltrami operator applied on the identity function  $\mathbf{x} \rightarrow \mathbf{x}$ ,  $\forall \mathbf{x} \in \mathcal{M}$  is directly linked to the *mean curvature normal*  $\bar{\kappa}\mathbf{n}$  by the following formula:

$$\Delta_{\mathcal{M}} \mathbf{x} = -2\bar{\kappa}\mathbf{n} \quad (27)$$

This equation provides us a way to approximate the Laplace-Beltrami operator through the approximation of the mean curvature normal. This fact is exploited by methods presented in the next section.

## 2.2 Discrete Laplacians from Differential Geometry

Desbrun et al. [6] construct a triangulation  $\mathcal{T}_h$  with the vertices in the sampling  $x_0, x_1, \dots, x_{n-1}$  approximating the manifold  $\mathcal{M}$ , and then use the following discrete expression for the *discrete normal curvature*  $\bar{\kappa}_h\mathbf{n}$  of the manifold



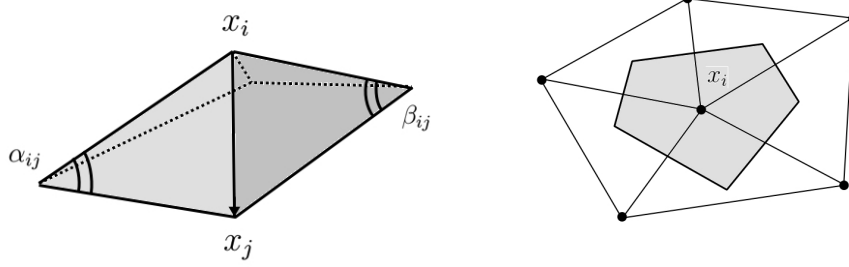


Figure 4: One term of curvature normal formula and one Voronoi cell constructed around the node  $x_i$

$\mathcal{M}$ :

$$-\overline{\kappa}_h = \frac{1}{4A_i} \sum_{j \in N_1(i)} (\cot \alpha_{ij} + \cot \beta_{ij}) (x_j - x_i) \quad (28)$$

where  $A_i$  is the area of all the triangles of the mesh sharing the node  $x_i$ ;  $N_1(i)$  is the first ring of neighbors of the  $i$ th node;  $\alpha_{ij}$ ,  $\beta_{ij}$  are the angles of the triangles of the mesh that lie on the opposite side to the edge  $(i, j)$  with respect to the node  $x_i$  (Figure 4). Observe that for a flat surface the discrete curvature is equal to zero  $\overline{\kappa}_h = 0$ . This is a geometric approach that relies on the intrinsic properties of the triangulation  $\mathcal{T}_h$  and is based on the geometric meaning of the curvature normal  $\overline{\kappa}$ . Using equations (28) and (27) it can be shown [20] that this approach leads to a discrete Laplacian with masses

$$d_i = \frac{A_i}{3}$$

where  $A_i$  is the area of all the triangles of the mesh with a vertex in  $x_i$ , and weights

$$w_{ij} = \frac{\cot(\alpha_{ij}) + \cot(\beta_{ij})}{2}$$

### 2.3 Linear Finite Element Method Laplacian

The eigenvalue problem (24) can be rewritten in the equivalent weak form

$$\langle \nabla f, \nabla v \rangle_{L^2(\mathbb{S}^2)} = \lambda \langle f, v \rangle_{L^2(\mathbb{S}^2)} \quad \forall v \in L^2(\mathbb{S}^2) \quad (29)$$

The Finite Element Method (FEM) is a numerical algorithm that allows to calculate a discrete approximation of the solution  $f$  through a functional discretization of the weak eigenvalue problem (29). We will discuss this method deeper in section 4.2. By projecting equation (29) on a finite dimensional functional subspace of  $L^2(\mathbb{S}^2)$  spanned by  $n$  basis functions  $\phi_i$ , by writing  $n$  times the equation (41), setting each time the *test* function  $v$  equal to the  $i$ th basis function  $\phi_i$  we obtain the generalized algebraic eigenvalue problem

$$\begin{aligned} &\text{Find } (f, \lambda) \text{ such that } \mathbf{A}\mathbf{f} = \lambda\mathbf{B}\mathbf{f} \\ &\begin{cases} (\mathbf{A})_{ij} &= \int_{\mathbb{S}^2} \nabla \phi_i(\mathbf{x}) \cdot \nabla \phi_j(\mathbf{x}) d\mathbf{x} \\ (\mathbf{B})_{ij} &= \int_{\mathbb{S}^2} \phi_i(\mathbf{x}) \phi_j(\mathbf{x}) d\mathbf{x} \\ (\mathbf{f})_i &= f_i : \quad f(\mathbf{x}) = f_0 \phi_0(\mathbf{x}) + \dots + f_{n-1} \phi_{n-1}(\mathbf{x}) \end{cases} \end{aligned}$$

that can be solved through usual algebraic solvers.

**FEM Laplacian as a Differential Geometry Laplacian.** Levy [14] showed that by using the *lumped* mass matrix  $\mathbf{D}_{ii} = \sum_j \mathbf{B}_{ij}$  the FEM Laplacian

$$\mathbf{D}^{-1}\mathbf{A}$$

is the same Laplacian of Desbrun et al. [6] introduced in the previous section. Meyer et al. [16] proposed another discrete Laplacian by modifying the masses of Desbrun et al. setting them to

$$d_i = a_V(i),$$

where  $a_V(i)$  is the area of the polygon obtained by joining the circumcenters of the triangles surrounding node  $i$  (i.e. the Voronoi cell, figure 4).

## 2.4 Graph Laplacian for manifolds

Belkin et al. [3] prove convergence of eigenvectors of the *Heat Kernel Graph Laplacian*  $\mathbf{L}_n^t$  (HKGL) of a data point cloud to the eigenfunctions of the Laplace Beltrami operator  $\Delta_{\mathcal{M}}$  on the manifold  $\mathcal{M}$ , when the data is sampled from a uniform distribution on  $\mathcal{M}$ . For this result to hold, they suppose

the manifold  $\mathcal{M}$  to be compact, infinitely differentiable and without boundary. We just point out that being  $\mathcal{M}$  compact,  $\Delta_{\mathcal{M}}$  has a discrete spectrum. The graph they use to approximate the manifold is constructed as follows: given a sampling  $\mathcal{P} = \{x_i \in \mathcal{M}\}_{i=0}^{n-1}$  on a  $k$ -dimensional manifold  $\mathcal{M} \subset \mathbb{R}^N$  they construct the full graph defined by the weights

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{4t}\right)$$

where  $\|\cdot\|$  is the Euclidean norm in the ambient space  $\mathbb{R}^N$  and whose Laplacian matrix

$$\mathbf{L}_n^t$$

we call *Heat Kernel Graph Laplacian*.

Observe that given a function  $f : \mathcal{P} \rightarrow \mathbb{R}$  defined on the sampling  $\mathcal{P}$  and defined the vector  $\mathbf{f} \in \mathbb{R}^n$  such that  $\mathbf{f}_i = f(x_i)$ , the Heat Kernel Graph Laplacian matrix acts on  $\mathbf{f}$  in the following way:

$$(\mathbf{L}_n^t \mathbf{f})_i := \sum_{j=0}^{n-1} e^{-\frac{\|x_i - x_j\|^2}{4t}} (f(x_i) - f(x_j)) \quad (30)$$

This graph construction is motivated by the fact that the HKGL is nothing else than the natural discretization of the continuous *functional approximation to the Laplace-Beltrami operator*  $L^t : L^2(\mathcal{M}) \rightarrow L^2(\mathcal{M})$  whose eigenvectors and eigenvalues are proven to converge to the ones of  $\Delta_{\mathcal{M}}$ .

**Definition 2.1.** ([2, Belkin et al.] *Functional approximation to the Laplace-Beltrami operator*)

Let  $\mu$  be the uniform probability measure on the manifold  $\mathcal{M}$ , where  $\text{vol}(\mathcal{M})$  is the volume of  $\mathcal{M}$ . We define the functional approximation to the Laplace-Beltrami operator to be the operator  $L^t : L^2(\mathcal{M}) \rightarrow L^2(\mathcal{M})$  such that

$$L^t f(y) = \int_{\mathcal{M}} e^{-\frac{\|y-x\|^2}{4t}} (f(y) - f(x)) d\mu(x)$$

We end this Chapter by stating the theorem of spectral convergence of the HKGL to the Laplace-Beltrami operator  $\Delta_{\mathcal{M}}$ , that makes it a really good candidate to construct rotation invariant graphs.

*Theorem 2.1.* (Belkin et al., [3]) Let  $\lambda_{n,i}^t$  be the  $i$ th eigenvalue of

$$\frac{(4\pi t)^{-(k+2)/2}}{n} \mathbf{L}_n^t$$

and  $\mathbf{v}_{n,i}^t$  be the corresponding eigenvector. Let  $\lambda_i$  and  $v_i$  be the corresponding eigenvalue and eigenfunction of  $\Delta$  respectively. Then there exists a sequence  $t_n \rightarrow 0$ , such that

$$\begin{aligned} \lim_{n \rightarrow \infty} \lambda_{n,i}^{t_n} &= \lambda_i \\ \lim_{n \rightarrow \infty} \left\| \mathbf{v}_{n,i}^{t_n} - v_i(\mathbf{x}) \right\|_2 &= 0 \end{aligned} \tag{31}$$

where the limits are in probability.

### 3 Graph Spherical Convolutions

In this chapter we first introduce DeepSphere [18], an example of Graph Spherical Convolutional Neural Network that uses a modified version of the HKGL to perform graph convolutions and hierarchical pooling to achieve rotation equivariance, and computational efficiency. In section 3.2 we prove a pointwise convergence result of the full HKGL in the case of the sphere using a regular deterministic sampling scheme. In section 3.3 we show a way of modifying the original graph of DeepSphere to improve its spectral convergence to  $\Delta_{\mathbb{S}^2}$  while managing to contain the computational costs of graph convolutions.

#### 3.1 Graph Spherical Convolutional Neural Networks

Perraudin et al. [18] have proposed a Spherical CNN to process and analyze spherical sky maps, as the Cosmic Radiation map in figure 1. Sky images are modeled as signals on the vertices of a *sparse* graph  $G$  on the vertex set  $V$  of the image pixels  $(v_i)$  with weights

$$w_{ij} = \exp\left(-\frac{\|v_i - v_j\|^2}{4t}\right)$$

where the kernel width  $t$  is a parameter to optimize. In an earlier work, Belkin et al. [3] proved the convergence of eigenvectors of the *full* graph  $G$  to the eigenfunctions of the Laplace Beltrami operator  $\Delta_{\mathbb{S}^2}$ , when the data is sampled from a uniform distribution on  $\mathbb{S}^2$  (see Section 2.4, theorem 2.1). For this reason and for the intuition we presented at the beginning of Chapter 2 when introducing the mean equivariance error  $\overline{E}_G$ , we expect the construction of Perraudin et al. to work well only for images that were sampled with equi area sampling schemes. On such sampling schemes, since the graph Laplacian eigenvectors well approximate the spherical harmonics, the graph convolution (20) well approximates the true spherical convolution (10).

$$\mathbf{f}^\top \mathbf{v}_{i(\ell, m)} \approx \int_{\eta \in \mathbb{S}^2} f(\eta) Y_\ell^m(\eta) d\mu(\eta) = \hat{f}(\ell, m) \quad (32)$$

The most used sampling scheme by cosmologists and astrophysics is called HEALPix [10], and is the one implemented in DeepSphere. HEALPix is an

acronym for Hierarchical Equal Area isoLatitude Pixelization of a sphere. This sampling scheme produces a subdivision of the sphere in which each pixel covers the same surface area as every other pixel. It is parametrized by a parameter  $N_{side} \in \mathbb{N}$ , and is made of  $n = 12N_{side}^2$  pixels. The points of this sampling lie on isolatitude rings, making it possible to implement an FFT algorithm for the discrete SHT. The minimal resolution for HEALPix is given by  $N_{side} = 1$  and is made by 12 pixels. For each increasing value of  $N_{side}$  each patch is divided into 4 equal area patches centered around the pixels of the new sampling (figure 5).

In chapter 5 of this work we'll deepen the relationship between the continuous spherical Fourier transform and the graph Fourier transform in case of non-uniform sampling measures. Perraudin et al. propose an efficient implementation of the graph convolution (20) to be implemented in each layer of their Graph Spherical Convolutional Neural Network. They propose to learn only those filters that are polynomials of degree  $q$  of the eigenvalues  $\lambda_i$

$$\begin{aligned} k(\lambda) &= \sum_{j=0}^q \theta_j \lambda^j \\ \Omega_k \mathbf{f} &= \mathbf{V} \left( \sum_{j=0}^q \theta_j \Lambda^j \right) \mathbf{V}^\top \mathbf{f} = \sum_{j=0}^q \theta_j \mathbf{L}^j \mathbf{f} \end{aligned} \tag{33}$$

Learning the filter  $k$  means learning the  $q + 1$  coefficients  $\theta_j$ . In this way they solve different problems at once: first, to compute the graph convolution (20) there's no need of computing the expensive eigen decomposition of  $\mathbf{L}$ , but they just need to evaluate a polynomial of the sparse matrix  $\mathbf{L}$ . Thanks to a suitable parametrization of the polynomial  $\sum_{j=0}^q \theta_j \mathbf{L}^j$  in term of Chebyshev polynomials, they manage to reduce the computations needed to evaluate such filter to  $\mathcal{O}(|E|)$ . Since in a NN this filtering operation has to be computed in every forward and backward step of the training phase, this gain in efficiency is dramatically important and led to speedups of different orders of magnitude compared to the architecture of Cohen et al. (see Chapter 5, table 4). The filtering operation (33) can be seen also in the vertex domain as a weighted sum of the  $q$ -neighborhoods of each vertex. This is due to the fact that  $\mathbf{L}$  has the same sparsity structure of the adjacency matrix of the graph, and thus  $(\mathbf{L}^q)_{ij}$  will be non-zero if and only if the vertices  $v_i, v_j$  are connected by a path of length  $q$ .

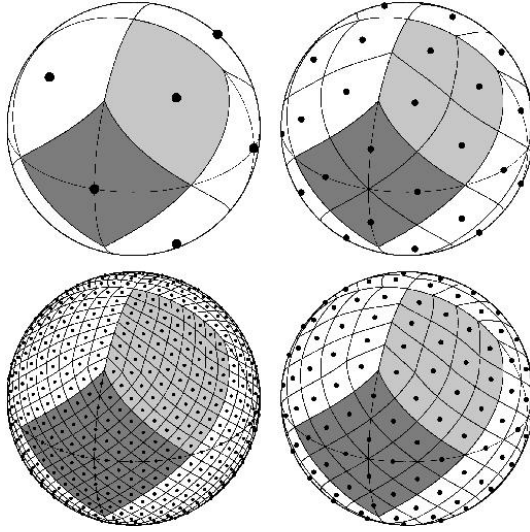


Figure 5: HEALPix sampling for  $N_{side} = 1, 2, 3, 4$  [10]

Despite theorem 2.1 [2] states the spectral convergence of the *full* HKGL to  $\Delta_{S^2}$ , the *sparse* version of the HKGL of Perraudin et al. does not seem to show such convergence. In figure (6) we see the correspondence between the subspaces spanned by the graph Fourier modes of the graph Laplacian used by Perraudin et al. and the true spherical harmonics. We can also see the plot of the graph eigenvalues: we can see that they clearly come in groups of  $(2\ell + 1)$  eigenvalues corresponding to each  $\ell$ th degree of the spherical harmonics. We thus call  $\mathbf{v}_{i(\ell,m)}$  the  $i$ th graph eigenmode corresponding to the one of degree  $\ell$  and order  $m$ . We compute the normalized Discrete Spherical Harmonic Transform (DSHT) of each  $\mathbf{v}_{i(\ell,m)}$  up to the degree  $\ell_{\max}$ . The entry  $(\ell, \kappa)$  of the matrix represented in the figure corresponds to the percentage of energy of the  $\ell$ th eigenspace  $V_\ell = \text{span}\{\mathbf{v}_{i(\ell,-\ell)}, \mathbf{v}_{i(\ell,-\ell+1)}, \dots, \mathbf{v}_{i(\ell,\ell)}\}$  contained in the  $\kappa$ th eigenspace of the true spherical harmonics. In a perfect situation, this matrix would be the identity matrix, being all the energy of the  $\ell$ th graph eigenspace contained in the corresponding one spanned by the true spherical harmonics. It can be seen that the eigenmodes of the graph Laplacian span almost the same subspaces as the spherical harmonics in the low frequencies, but this alignment gets worse at higher frequencies. Furthermore, it can be noticed that even by improving the resolution of the graph, the low frequency eigenspaces do not get better aligned.

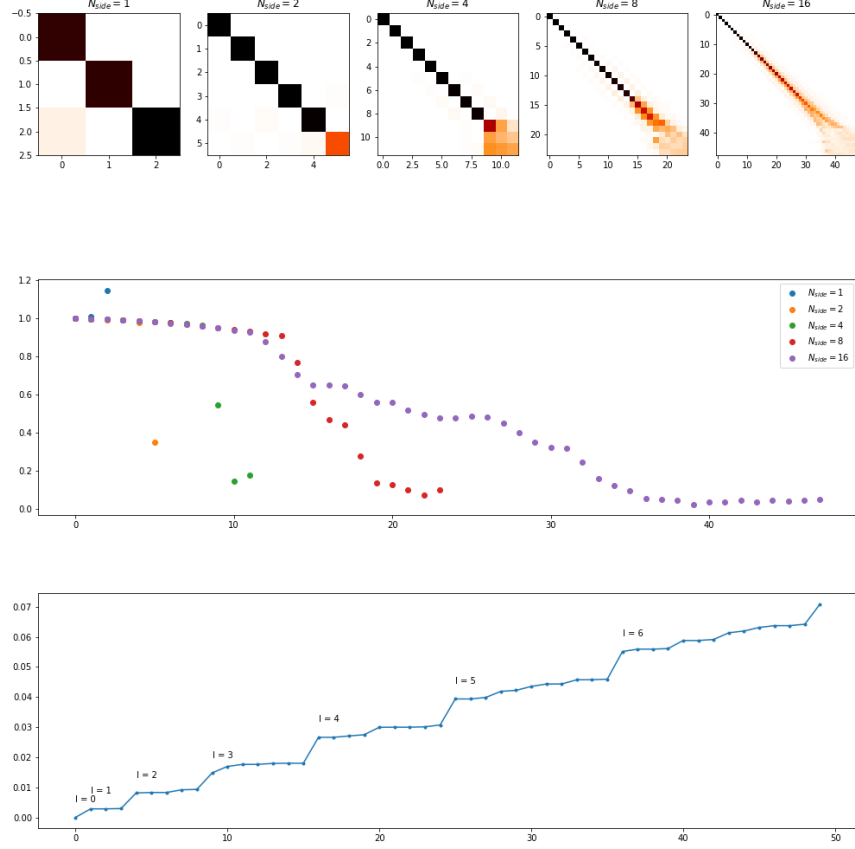


Figure 6: Alignment of the eigenvectors of the graph Laplacian of the DeepSphere graph, the starting point of this work. In the middle we plot the diagonals of the matrices on top, and on the bottom we plot its spectrum for  $N_{side} = 16$ .

### 3.2 Pointwise convergence of the Heat Kernel Graph Laplacian on the Sphere

Here we prove a pointwise convergence result of the full graph Laplacian in the case of the sphere on a deterministic sampling scheme that is regular enough. Our proof will be constructed following the ideas presented in the proof of theorem 3.1.

**Definition 3.1.** (*Heat Kernel Graph Laplacian operator*)

Given a sampling  $\{x_i \in \mathcal{M}\}_{i=0}^{n-1}$  of the manifold we define the **operator**  $L_n^t$  such that

$$L_n^t f(y) := \frac{1}{n} \left[ \sum_{i=0}^{n-1} \exp -\frac{\|x_i - y\|^2}{4t} (f(y) - f(x_i)) \right]$$



Observe that the Heat Kernel Graph Laplacian operator restricted on the sample points  $x_0, \dots, x_{n-1}$  acts as the usual Heat Kernel Graph Laplacian matrix  $\mathbf{L}_n^t$  rescaled by a factor of  $\frac{1}{n}$ :

$$L_n^t f(x_i) = \frac{1}{n} (\mathbf{L}_n^t \mathbf{f})_i$$

.

*Theorem 3.1.* from [2, Belkin et al.]

Let  $\mathcal{M}$  be a  $k$ -dimensional compact smooth manifold embedded in some euclidean space  $\mathbb{R}^N$ , and fix  $p \in \mathcal{M}$ . Let the data points  $x_1, \dots, x_n$  be sampled from a uniform distribution on the manifold  $\mathcal{M}$ . Set  $t_n = n^{-\frac{1}{k+2+\alpha}}$ , for any  $\alpha > 0$  and let  $f \in \mathcal{C}_\infty(\mathcal{M})$ . Then:

$$\forall \epsilon > 0 \quad \mathbb{P} \left[ \left| \frac{1}{t} \frac{1}{(4\pi t)^{k/2}} L_n^{t_n} f(p) - \frac{1}{\text{vol}(\mathcal{M})} L^{t_n} f(p) \right| > \epsilon \right] \xrightarrow{n \rightarrow \infty} 0$$

This theorem states a convergence in probability of  $L_n^t$  to  $L^t$ , that is far from being as strong as spectral convergence of theorem 2.1. However, we want to show that a similar result still holds in the specific case of the manifold  $\mathcal{M}$  being the 2-Sphere  $\mathbb{S}^2$  and where the points  $x_1, \dots, x_n$  are not sampled from a random distribution on  $\mathbb{S}^2$ , but are defined by the HEALPix sampling. To understand the differences between theorem 3.1 and theorem 3.2 it is useful to first review the proof of theorem 3.1. For this proof we'll need to use the Hoeffding's inequality that we recall here under:

(*Hoeffding's inequality*)

Let  $X_1, \dots, X_n$  be independent identically distributed random variables, such that  $|X_i| \leq K$ . Then

$$\mathbb{P} \left\{ \left| \frac{\sum_i X_i}{n} - \mathbb{E} X_i \right| > \epsilon \right\} < 2 \exp \left( -\frac{\epsilon^2 n}{2K^2} \right) \quad (34)$$

*Proof of Theorem 3.1.* The first step is to observe that for any fixed  $t > 0$ , any fixed function  $f$  and a fixed point  $y \in \mathbb{S}^2$ , the Heat Kernel Graph Laplacian  $L_n^t$  is an unbiased estimator for the Functional Approximation of the Laplace-Beltrami  $L^t$ . In other words,  $L_n^t f(y)$  is the empirical average of  $n$  i.i.d. random

variables  $X_i = e^{-\frac{\|x_i - y\|^2}{4t}} (f(y) - f(x_i))$  with expected value corresponding to  $L^t f(y)$ . Thus,

$$\mathbb{E} L_n^t f(y) = \mathbb{E} \frac{1}{n} X_i = \mathbb{E} X_i = L^t f(y), \quad (35)$$

and by the strong law of large numbers we have that

$$\lim_{n \rightarrow \infty} L_n^t f(y) = L^t f(y). \quad (36)$$

The core of the work of Belkin et al. is the proof, that we will not discuss, of the following proposition.

**Proposition 1.** *Under the same hypothesis of theorem 3.1, we have the following pointwise convergence*

$$\frac{1}{t} \frac{1}{(4\pi t)^{k/2}} L^t f(p) \xrightarrow{t \rightarrow 0} \frac{1}{\text{vol}(\mathcal{M})} \Delta_{\mathcal{M}} f(p).$$

Thanks to Proposition 1 and equation (36), a straightforward application of Hoeffding's inequality with  $K = \frac{1}{t} \frac{1}{(4\pi t)^{k/2}}$  together with equation (35) leads to

$$\mathbb{P} \left[ \frac{1}{t(4\pi t)^{k/2}} |L_n^t f(y) - L^t f(y)| > \epsilon \right] \leq 2e^{-1/2\epsilon^2 n t (4\pi t)^{k/2}} \quad (37)$$

We want the right hand side of equation (37) to go to 0 for  $n \rightarrow \infty, t \rightarrow 0$  at the same time. For this to happen, we need to find a sequence  $(t_n)$  such that

$$\begin{cases} t_n \xrightarrow{n \rightarrow \infty} 0 \\ 2e^{-1/2\epsilon^2 n t_n (4\pi t_n)^{k/2}} \xrightarrow{n \rightarrow \infty} 0 \end{cases}$$

By fixing  $t_n = n^{-\frac{1}{k+2+\alpha}}$ , for any  $\alpha > 0$ , it is easy to check that  $-1/2\epsilon^2 n t_n (4\pi t_n)^{k/2} \xrightarrow{n \rightarrow \infty} +\infty$ , thus concluding the proof.

□

Now we can observe that in order to adapt this proof to the case of the sphere with an equi area sampling scheme we need to modify two key things. First, due to the deterministic nature of the sampling scheme, we need to prove that

for any fixed  $t > 0$ , any fixed function  $f$  and any point  $y \in \mathbb{S}^2$

$$\left| L_n^t f(y) - L^t f(y) \right| \xrightarrow{n \rightarrow \infty} 0, \quad (38)$$

without relying on the strong law of large numbers. Once proven such result, we need to prove that

$$\left| \frac{1}{4\pi t^2} \left( L_n^t f(x) - L^t f(x) \right) \right| \xrightarrow[n \rightarrow \infty]{t \rightarrow 0} 0$$

We need now to define some geometrical quantities that we'll need. Given a sampling  $x_0, \dots, x_{n-1}$  define  $\sigma_i$  to be the patch of the surface of the sphere corresponding to the  $i$ th point of the sampling, define  $A_i$  to be its corresponding area and  $d_i$  to be the radius of the smallest ball in  $\mathbb{R}^3$  containing the  $i$ -th patch (see Figure 7). Define  $d^{(n)} := \max_{i=0, \dots, n} d_i$  and  $A^{(n)} = \max_{i=0, \dots, n} A_i$ .

Once proven the limit (38), Proposition 1 leads to our main result:

*Theorem 3.2.* For a sampling  $\mathcal{P} = \{x_i \in \mathbb{S}^2\}_{i=0}^{n-1}$  of the sphere that is equi area and such that  $d^{(n)} \leq \frac{C}{\sqrt{n}}$ , for all  $f : \mathbb{S}^2 \rightarrow \mathbb{R}$  Lipschitz with respect to the euclidean distance in  $\mathbb{R}^3$ , for all  $y \in \mathbb{S}^2$ , there exists a sequence  $t_n = n^\beta$  such that the rescaled Heat Kernel Graph Laplacian operator  $\frac{|\mathbb{S}^2|}{4\pi t_n} L_n^t$  converges pointwise to the Laplace Beltrami operator on the sphere  $\Delta_{\mathbb{S}^2}$  for  $n \rightarrow \infty$ :

$$\lim_{n \rightarrow \infty} \frac{|\mathbb{S}^2|}{4\pi t_n} L_n^{t_n} f(y) = \Delta_{\mathbb{S}^2} f(y).$$

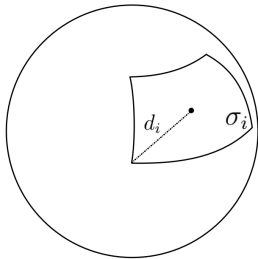


Figure 7: Geometric characteristics of the  $i$ th patch

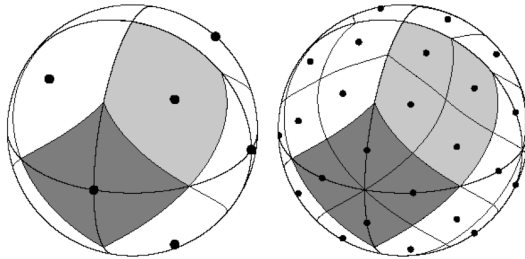


Figure 8: HEALPix equal areas patches for  $N_{side} = 1$ ,  $N_{side} = 2$

### 3.2.1 Proof of the pointwise convergence of the Heat Kernel Graph Laplacian on the Sphere for an equi area sampling scheme

Our first goal is to prove the following Proposition:

**Proposition 2.** *For an equal area sampling  $\{x_i \in \mathbb{S}^2\}_{i=0}^{n-1} : A_i = A_j \forall i, j$  of the sphere it is true that for all  $f : \mathbb{S}^2 \rightarrow \mathbb{R}$  Lipschitz with respect to the euclidean distance  $\|\cdot\|$  with Lipschitz constant  $\mathcal{L}_f$*

$$\left| \int_{\mathbb{S}^2} f(x) d\mu(x) - \frac{1}{n} \sum_i f(x_i) \right| \leq \mathcal{L}_f d^{(n)}.$$

Furthermore, for all  $y \in \mathbb{S}^2$  the Heat Kernel Graph Laplacian operator  $L_n^t$  converges pointwise to the functional approximation of the Laplace Beltrami operator  $L^t$

$$L_n^t f(y) \xrightarrow{n \rightarrow \infty} L^t f(y).$$

*Proof.* Let us assume that the function  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$  is Lipschitz with Lipschitz constant  $\mathcal{L}_f$ , we have

$$\left| \int_{\sigma_i} f(x) d\mu(x) - \frac{1}{n} f(x_i) \right| \leq \mathcal{L}_f d^{(n)} \frac{1}{n}$$

So, by triangular inequality and by summing all the contributions of all the  $n$  patches

$$\left| \int_{\mathbb{S}^2} f(x) d\mu(x) - \frac{1}{n} \sum_i f(x_i) \right| \leq \sum_i \left| \int_{\sigma_i} f(x) d\mu(x) - \frac{1}{n} f(x_i) \right| \leq n \mathcal{L}_f d^{(n)} \frac{1}{n} = \mathcal{L}_f d^{(n)}$$

Thanks to this result, we have the following two pointwise convergences

$$\forall f \text{ Lipschitz}, \quad \forall y \in \mathbb{S}^2, \quad \frac{1}{n} \sum_i e^{-\frac{\|x_i - y\|^2}{4t}} \rightarrow \int e^{-\frac{\|x - y\|^2}{4t}} d\mu(x)$$

$$\forall f \text{ Lipschitz}, \quad \forall y \in \mathbb{S}^2, \quad \frac{1}{n} \sum_i e^{-\frac{\|x_i - y\|^2}{4t}} f(x_i) \rightarrow \int e^{-\frac{\|x - y\|^2}{4t}} f(x) d\mu(x)$$

Definitions 3.1 and 2.1 end the proof. □

Now, we just proved that *keeping  $t$  fixed*  $L_n^t f(x) \rightarrow L^t f(x)$ . Now our goal is to prove that:

**Proposition 3.** *Given a sampling regular enough i.e., for which we assume  $A_i = A_j \ \forall i, j$  and  $d^{(n)} \leq \frac{C}{\sqrt{n}}$ , for a fixed  $t > 0$ , a fixed Lipschitz function  $f$  and a fixed point  $y \in \mathbb{S}^2$  there exists a sequence  $t_n = n^\beta, \beta < 0$  such that*

$$\forall f \text{ Lipschitz}, \forall x \in \mathbb{S}^2 \quad \left| \frac{1}{4\pi t_n^2} \left( L_n^{t_n} f(x) - L^{t_n} f(x) \right) \right| \xrightarrow{n \rightarrow \infty} 0.$$

The main result of this section, theorem 3.2, is then an immediate consequence of Proposition 3 and Proposition 1.

*Proof of Proposition 3.* We define for simplicity of notation

$$\begin{aligned} \phi^t(x; y) &:= e^{-\frac{\|x-y\|^2}{4t}} (f(y) - f(x)) \\ K^t(x, y) &:= e^{-\frac{\|x-y\|^2}{4t}} \end{aligned}$$

We start by writing the following chain of inequalities

$$\begin{aligned} \|L_n^t f - L^t f\|_\infty &= \max_{y \in \mathbb{S}^2} |L_n^t f(y) - L^t f(y)| \\ &= \max_{y \in \mathbb{S}^2} \left| \frac{1}{n} \sum_{i=1}^n \phi^t(x_i; y) - \int_{\mathbb{S}^2} \phi^t(x; y) d\mu(x) \right| \\ &\leq \max_{y \in \mathbb{S}^2} \sum_{i=1}^n \left| \frac{1}{n} \phi^t(x_i; y) - \int_{\sigma_i} \phi^t(x; y) d\mu(x) \right| \\ &\leq \max_{y \in \mathbb{S}^2} \left[ \mathcal{L}_{\phi_y^t} d^{(n)} \right] \end{aligned}$$

where  $\mathcal{L}_{\phi_y^t}$  is the Lipschitz constant of  $x \rightarrow \phi^t(x, y)$  and where we used for the last inequality Proposition 2. If we assume  $d^{(n)} \leq \frac{C}{\sqrt{n}}$  we have that

$$\|L_n^t f - L^t f\|_\infty \leq \max_{y \in \mathbb{S}^2} \left[ \mathcal{L}_{\phi_y^t} \frac{C}{\sqrt{n}} \right]$$

Let's now find the explicit dependence  $t \rightarrow \mathcal{L}_{\phi_y^t}$

$$\begin{aligned}
\mathcal{L}_{\phi_y^t} &= \|\partial_x \phi^t(\cdot; y)\|_\infty \\
&= \|\partial_x (K^t(\cdot; y)f)\|_\infty \\
&= \|\partial_x K^t(\cdot; y)f + K^t(\cdot; y)\partial_x f\|_\infty \\
&\leq \|\partial_x K^t(\cdot; y)f\|_\infty + \|K^t(\cdot; y)\partial_x f\|_\infty \\
&\leq \|\partial_x K^t(\cdot; y)\|_\infty \|f\|_\infty + \|K^t(\cdot; y)\|_\infty \|\partial_x f\|_\infty \\
&= \|\partial_x K^t(\cdot; y)\|_\infty \|f\|_\infty + \|\partial_x f\|_\infty \\
&= \mathcal{L}_{K_y^t} \|f\|_\infty + \|\partial_x f\|_\infty \\
&= \mathcal{L}_{K_y^t} \|f\|_\infty + \mathcal{L}_f
\end{aligned}$$

where  $\mathcal{L}_{K_y^t}$  is the Lipschitz constant of  $x \rightarrow K^t(x; y)$ . We can observe that such constant does not depend on  $y$ :

$$\mathcal{L}_{K_y^t} = \left\| \partial_x e^{-\frac{x^2}{4t}} \right\|_\infty = \left\| \frac{x}{2t} e^{-\frac{x^2}{4t}} \right\|_\infty = \frac{x}{2t} e^{-\frac{x^2}{4t}} \Big|_{x=\sqrt{2t}} = (2et)^{-\frac{1}{2}} \propto t^{-\frac{1}{2}}$$

So we can continue

$$\begin{aligned}
\max_{y \in \mathbb{S}^2} \left[ \mathcal{L}_{\phi_y^t} \frac{C}{\sqrt{n}} \right] &\leq \frac{C}{\sqrt{n}} \left( (2et)^{-\frac{1}{2}} \|f\|_\infty + \mathcal{L}_f \right) \\
&\leq \frac{C \|f\|_\infty}{\sqrt{n}(2et)^{\frac{1}{2}}} + \frac{C}{\sqrt{n}} \mathcal{L}_f
\end{aligned}$$

So we have that, rescaling by a factor  $\frac{1}{4\pi t^2}$

$$\begin{aligned}
\left\| \frac{1}{4\pi t^2} (L_n^t f - L^t f) \right\|_\infty &\leq \frac{1}{4\pi t^2} \|L_n^t f - L^t f\|_\infty \\
&\leq \frac{C}{4\pi} \left[ \frac{\|f\|_\infty}{\sqrt{2e}} \frac{1}{\sqrt{nt}^{\frac{5}{2}}} + \frac{\mathcal{L}_f}{\sqrt{nt^2}} \right]
\end{aligned}$$

$$\text{we want } \begin{cases} t \rightarrow 0 \\ n \rightarrow \infty \\ \sqrt{nt}^{\frac{5}{2}} \rightarrow \infty \\ \sqrt{nt^2} \rightarrow \infty \end{cases} \quad \text{in order for } \frac{C}{4\pi} \left[ \frac{\|f\|_\infty}{\sqrt{2e}} \frac{1}{\sqrt{nt}^{\frac{5}{2}}} + \frac{\mathcal{L}_f}{\sqrt{nt^2}} \right] \xrightarrow[t \rightarrow 0]{n \rightarrow \infty} 0$$

This is true if  $\begin{cases} t(n) = n^\beta, & \beta \in (-\frac{1}{5}, 0) \\ t(n) = n^\beta, & \beta \in (-\frac{1}{4}, 0) \end{cases} \implies t(n) = n^\beta, \quad \beta \in (-\frac{1}{5}, 0)$

Indeed

$$\sqrt{n}t^{\frac{5}{2}} = n^{5/2\beta+1/2} \xrightarrow{n \rightarrow \infty} \infty \text{ since } \frac{5}{2}\beta + 1/2 > 0 \iff \beta > -\frac{1}{5}$$

$$\sqrt{n}t^2 = n^{2\beta+1/2} \xrightarrow{N \rightarrow \infty} \infty \text{ since } 2\beta + 1/2 > 0 \iff \beta > -\frac{1}{4}$$

So, for  $t = n^\beta$  with  $\beta \in (-\frac{1}{5}, 0)$  we have that

$$\begin{cases} (t_n) \xrightarrow{n \rightarrow \infty} 0 \\ \left\| \frac{1}{4\pi t_n^2} L_n^{t_n} f - \frac{1}{4\pi t_n^2} L^{t_n} f \right\|_\infty \xrightarrow{n \rightarrow \infty} 0 \end{cases}$$

□

The proof of theorem 3.2 is now trivial:

*Proof of Theorem 3.2.* Thanks to Proposition 3 and Proposition 1 we conclude that  $\forall y \in \mathbb{S}^2$

$$\lim_{n \rightarrow \infty} \frac{1}{4\pi t_n^2} L_n^{t_n} f(y) = \lim_{n \rightarrow \infty} \frac{1}{4\pi t_n^2} L^{t_n} f(y) = \frac{1}{|\mathbb{S}^2|} \Delta_{\mathbb{S}^2} f(y)$$

□

The proof of this result is instructive since it shows that we need to impose some regularity conditions on the sampling. If the sampling is equal area as HEALPix, meaning that all the patches  $\sigma_i$  have the same area (i.e., HEALPix, see figure 8), then we need to impose that  $d^{(n)} \leq \frac{1}{\sqrt{n}}$ . If the sampling is not equal area, meaning that in general  $A_i \neq A_j$ , it can be shown that we need a slightly more complex condition:  $\max_{i=0, \dots, n-1} d_i A_i \leq C n^{-\frac{3}{2}}$ .

In the work of Belkin et al. [2] the sampling is drawn from a uniform random distribution on the sphere, and their proof heavily relies on the uniformity properties of the distribution from which the sampling is drawn. In our case the sampling is deterministic, and the fact that for a sphere there doesn't exist a regular sampling with more than 12 points (the vertices of a icosahedron) is indeed a problem that we need to overcome by imposing the regularity conditions above.

To conclude, we can see that the result obtained has the same form than the result obtained in [2]. Given the kernel density  $t(n) = n^\beta$ , if Belkin et al. proved convergence in the random case for  $\beta \in (-\frac{1}{4}, 0)$ , we proved convergence in the HEALPix case for  $\beta \in (-\frac{1}{5}, 0)$ . This kind of result can be interpreted in the following way. In order to have this pointwise convergence, we need to reduce the kernel width but *not so fast* compared to the resolution of the graph. In other words, the kernel width has to be reduced but is somewhat limited by the resolution of the graph. In the next section we'll see how to set in practice a good kernel width  $t$  given a graph resolution  $n$ .

*Remark.* Pointwise convergence is just a necessary condition for spectral convergence. Theorem 3.2 does not imply convergence of eigenvalues and eigenvectors.

### 3.3 How to build a good graph to approximate spherical convolutions

The current state of the art of rotation equivariant Graph CNN is DeepSphere [18]. However, if we measure the alignment of the eigenspaces spanned by the eigenvectors of its graph Laplacian and the ones spanned by the spherical harmonics we see that it does not get better as  $N_{side}$  increases (figure ??). We'll see that the main cause of this bad behavior of the eigenspaces is the fixed number of neighbors used for the construction of the graph. In this subsection we'll see that *to obtain the desired spectral convergence it is necessary to increase the number of neighbors as we decrease the kernel width  $t$* . We'll follow in practice what we did in proving theorem 3.2: first we'll build a full graph, and let the number of pixels  $n$  increase while keeping the kernel width  $t$  fixed. After having discussed the results, we'll try to find a sequence  $(t_{N_{side}})$  to obtain the expected spectral convergence. Only in the end we'll find a way to make the graph sparse to limit the computational costs of graph convolutions but keeping the eigen decomposition of the graph Laplacian as close to the spherical harmonics as possible.



### 3.3.1 Full graph, $n \rightarrow \infty$

Here we analyze what happens to the power density spectrum of the *full* Heat Kernel Graph Laplacian as we make  $n$  go to infinity while keeping  $t$  fixed. Since in the previous section we proved that (Proposition 1) for a sampling regular enough and a fixed  $t$ , a fixed function  $f$ , a fixed point  $y$

$$L_n^t f(y) \xrightarrow{n \rightarrow \infty} L^t f(y)$$

Since HEALPix is a very regular sampling of the sphere, we expect to observe (even if we didn't prove it) the corresponding spectral convergence proved in theorem 2.1. The results obtained are in figure 9, 10.

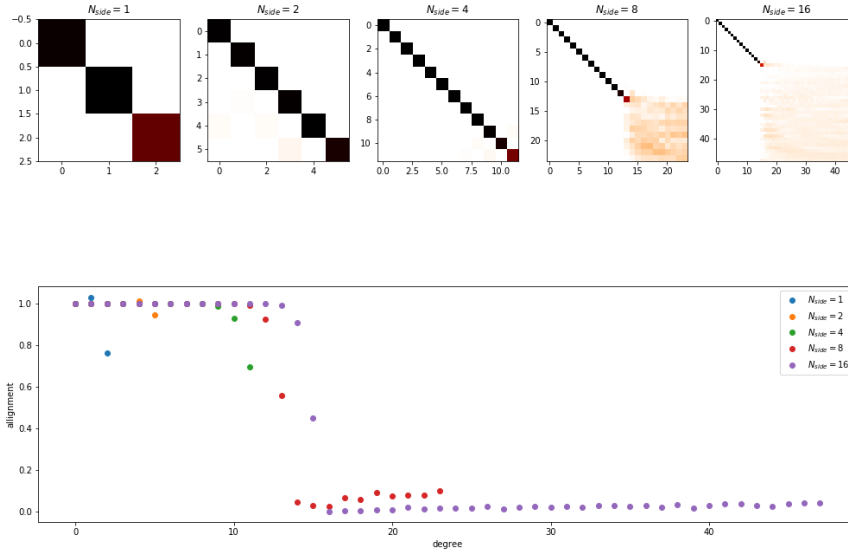


Figure 9: Alignment of the eigenvectors of the HKGL with a fixed kernel width  $t$

In figure 9 we see two things: first that there's a frequency threshold beyond which the Graph Laplacian is completely blind, approximately located at the 15th degree, and second that before this frequency threshold, we actually see the convergence expected: the alignment gets better as  $n$  gets bigger.

To explain this we refer to figure 11 where we show a simplified situation where we are sampling the interval  $[0, 2]$  and we plot the Gaussian kernel centered around the first pixel of the sampling corresponding to the origin. On the left-most image the pixels are correctly spaced with respect to the kernel width, in

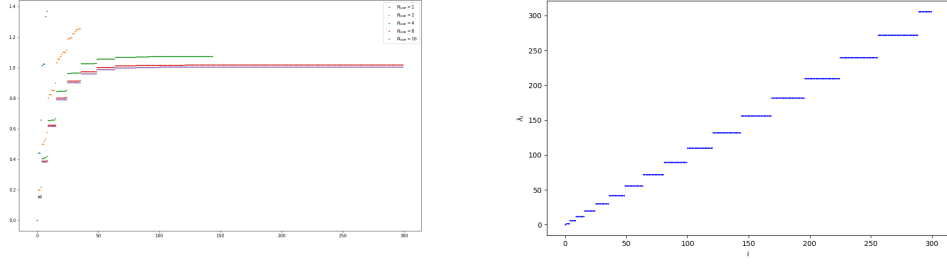


Figure 10: Left: spectrum of the HKGL with a fixed kernel width  $t$ . Right: true spectrum of  $\Delta_{S^2}$

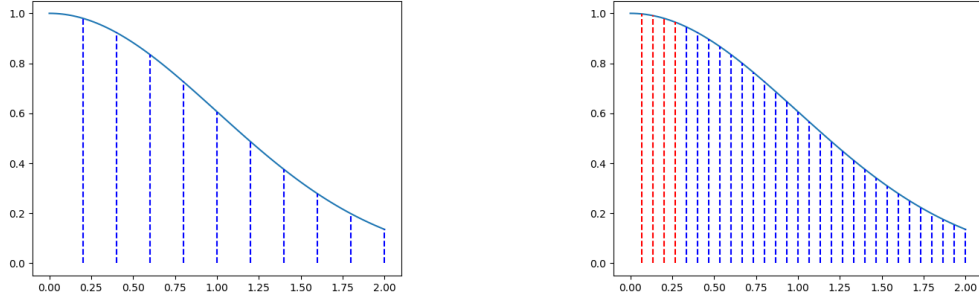


Figure 11: Frequency threshold explained.

the sense that the values of the kernel evaluated on the pixels are well far apart from each other. This makes the graph able to "see" all the pixels differently, and thus all the frequencies with wavelength around the order of magnitude of the average pixel distance will be captured by the graph. On the rightmost image in figure 11 there are too many pixels with respect to the kernel width: the values of the kernel evaluated on the pixels close to the origin, because of the slope of the kernel being almost zero are too close to each other (in red); because of this any variation of a signal on the red pixels would be almost invisible to the graph Laplacian. With this fixed kernel width  $t$ , no matter how much we sample the interval  $[0, 2]$ , any frequency with wavelength shorter than the radius  $r \approx 0.25$  becomes invisible to the graph Laplacian.

This phenomenon can be seen also in the spectrum represented in figure 10 where we plot the eigenvalues of the matrix  $\mathbf{L}_n^t$ : as  $N_{side}$  gets bigger the eigenvalues get more and more grouped in the usual groups of the same multiplicity of the corresponding spherical harmonics; however, there's a frequency (corresponding approximately to the degree  $\ell = 15$ ) from which all the eigenspaces tend to merge into one, corresponding to the eigenvalue 1.

### 3.3.2 Full graph, $t \rightarrow 0$

In this section we fix the parameter  $N_{side}$  and we make the kernel width  $t$  go to 0.

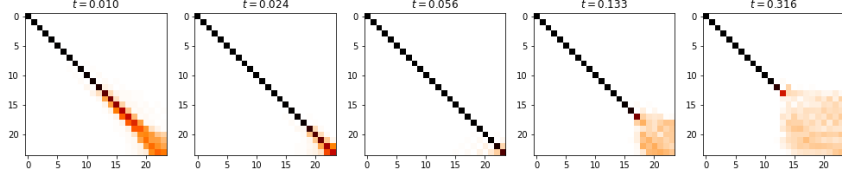


Figure 12: Alignment of the eigenspaces of the HKGL with a fixed number of points  $n$  corresponding to  $N_{side} = 8$

Results are in figure 12, 13: Starting from  $t = 0.32$ , the error in the high frequencies starts to get smaller, while the error in the low frequencies keeps staying low (Figure 14) up to  $t = 0.05$ . For  $t$  that gets smaller and smaller up to  $t = 0.01$ , we get worse alignment both in high and low frequencies (Figure 15). This behavior can be explained with the same arguments used in the previous section: high values of the kernel width correspond to a very flat kernel (figure 16, left), and thus the graph loses the capacity of individuating high frequencies as discussed before. On the opposite side, low values of the kernel width correspond to a very peaked kernel, that causes all the weights to be close to zero and thus the graph becomes less and less connected losing every capability of identifying different frequencies (figure 16, right).

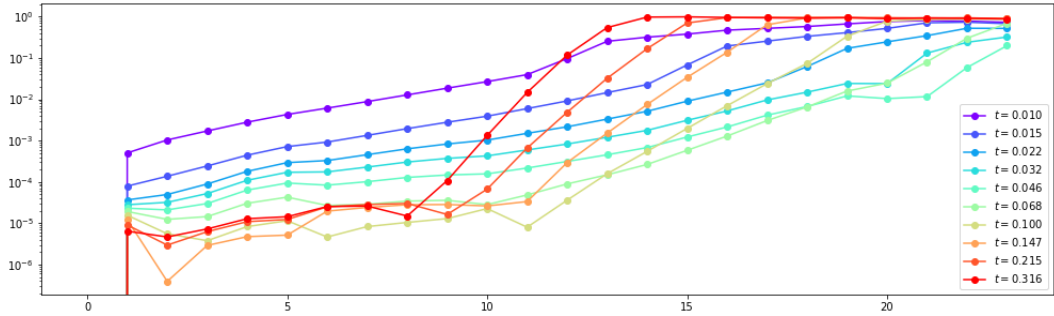


Figure 13: Whole trend

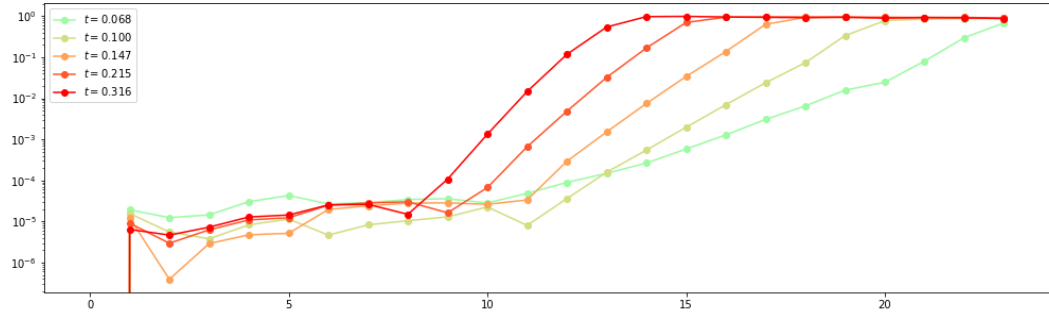


Figure 14: First trend: error stays low for low frequencies, and gets lower for high frequencies

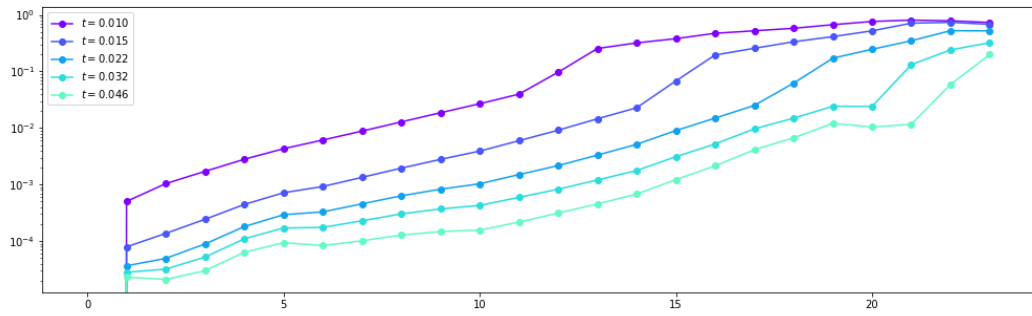


Figure 15: Second trend: error gets higher for both high and low frequencies

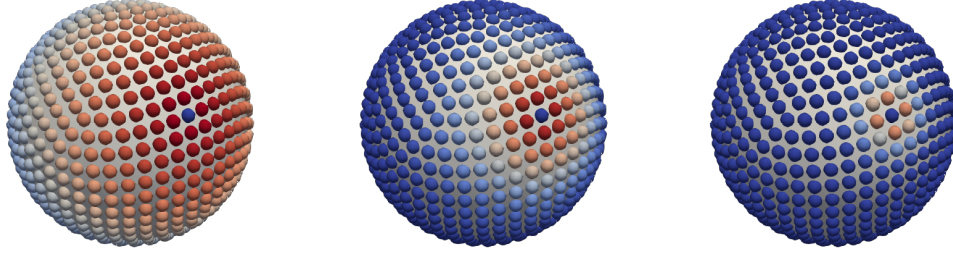


Figure 16: One row of the weight matrix for different values of the kernel width  $t$  plotted on the HEALPix sampling with  $N_{side} = 8$ . On the left, for a too large  $t$ , the HKGL can not capture high frequencies. On the right, for a too small  $t$ , the graph becomes almost completely disconnected. In the center a good value for  $t$  makes the HKGL able to see the most frequencies.

### 3.3.3 Putting it together: full graph, $n \rightarrow \infty$ and $t \rightarrow 0$

A grid search has been used to find the following optimal kernel width for different values of  $N_{side}$ , always in the case of a full graph, where we maximized the number of graph eigenspaces with the alignment value in figure 18 bigger than 80%. The optimal values of the kernel width  $t$  are shown in figure 17. In DeepSphere  $t$  is set to the average of the non zeros weight matrix entries, where the number of neighbors of each vertex is fixed between 7 and 8. We can see that the heuristic way of DeepSphere of setting the standard deviation  $t$  produces results of the same order of magnitude of the optimal value. It can be seen that the optimal values of  $t$  are very close to a linear trend in the log-log plot, showing an approximately polynomial relationship with the parameter  $N_{side}$  that could be used to extrapolate possible values of  $t$  for higher  $N_{side}$ . In figure 18 we can appreciate that for a full graph, each time we double the parameter  $N_{side}$ , we approximately double the degree  $\ell$  at which the graph eigenvectors are correctly aligned with the spherical harmonics.

### 3.3.4 Reducing the number of neighbors

For what it concerns how to make the graph sparse, the intuition is the following: remember that we want our graph Laplacian to approximate the operator

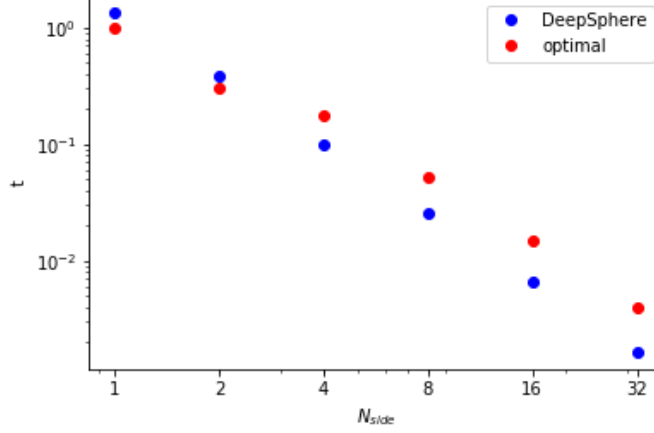


Figure 17: Standard deviation of the Gaussian kernel in a log-log plot. A straight line indicates a polynomial relation.

$L^t$ , that for sufficiently small  $t$  approximates  $\Delta$

$$\frac{1}{n} \left( \sum_i e^{-\frac{\|x_i - y\|^2}{4t}} (f(y) - f(x_i)) \right) \approx \int_{\mathbb{S}^2} e^{-\frac{\|x - y\|^2}{4t}} (f(y) - f(x)) d\mu(x) \approx \Delta_{\mathbb{S}^2} f(y)$$

So far we showed how to do so optimally with a full graph; however, a full graph comes at the cost of leading to a matrix  $\mathbf{L}_n^t$  that is full, and thus to a graph filtering cost of  $\mathcal{O}(n^2)$ , worse than the common SHT cost of  $\mathcal{O}(n^{3/2})$ . Perraudin et al. [18] constructed a nearest neighbor graph constraining the number of neighbors for each vertex to be fixed, making the graph filtering cost linear in the number of pixels. However, as we saw at the beginning of this section this leads to a poor alignment of the graph eigenvectors with the spherical harmonics and thus to a not so optimal rotation equivariance. Here we propose a different approach, based on the following intuition: making the graph sparse means deciding which weights  $w_{ij} = \exp -\frac{\|x_i - x_j\|^2}{4t}$  to set to zero. For this approximation to be accurate we want to set to zero only those weights that are small enough: let's define a new *epsilon graph*  $G'$  by fixing a threshold  $\epsilon$  on  $w_{ij} = e^{-\frac{\|x_i - x_j\|^2}{4t}}$  such that

$$w'_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{4t}} & \text{if } e^{-\frac{\|x_i - x_j\|^2}{4t}} \geq \epsilon \\ 0 & \text{if } e^{-\frac{\|x_i - x_j\|^2}{4t}} < \epsilon \end{cases}$$

By setting  $\epsilon = 0.01$  - or equivalently, thresholding the weights at

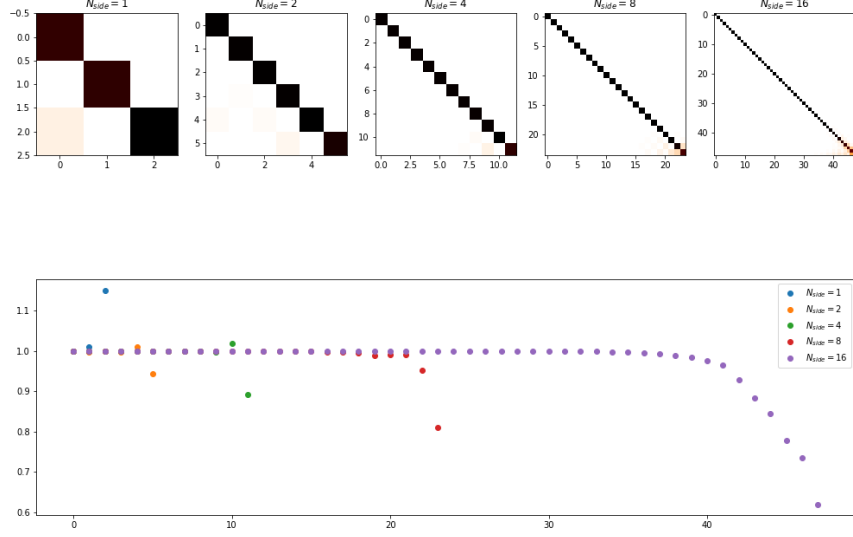


Figure 18: Alignment of eigenspaces of the optimal full graph.

$\|x_i - x_j\| \approx 3\sigma$  where  $\sigma = \sqrt{2t}$  is the standard deviation of the kernel - in figure 19 we can see the usual alignment plots of the graph  $G'$ :

$N_{side}$	Number of neighbors
1	11
2	16
4	37
8	43
16	52

Table 1

We see that we need to increase the number of neighbors as  $N_{side}$  gets bigger. Again, the intuition is the following: to have spectral convergence (a strong type of convergence) we need more and more global information and more precise. By fitting the relationship

$$\text{Number of neighbors} = (N_{side})^\alpha$$

to the data in table 1 we obtain that  $\alpha$  should be close to  $1/2$ , meaning that the complexity graph filtering with  $G'$  could be approximated by

$$\mathcal{O}(|E|) = \mathcal{O}(n\sqrt{N_{side}}) = \mathcal{O}(n^{5/4}).$$

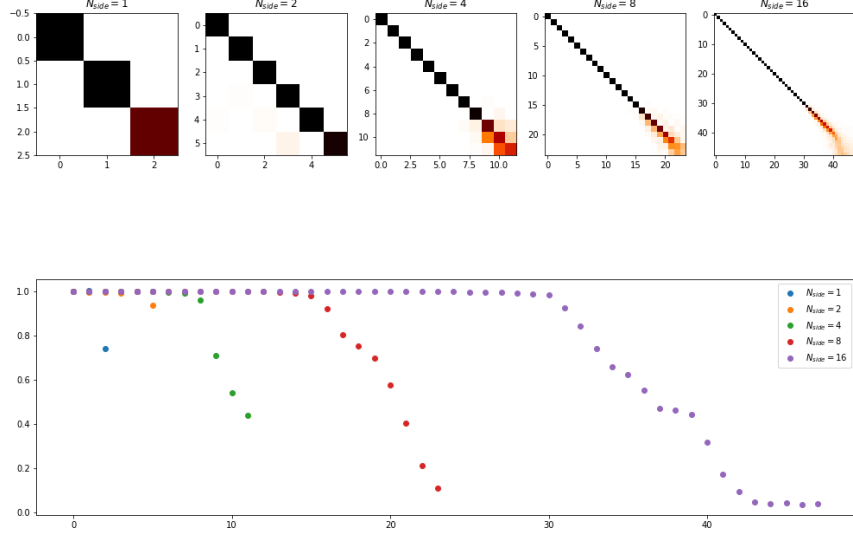


Figure 19: Optimal construction **thresholded at**  $k = 0.01$

where  $n$  is the number of vertices of the graph. This complexity is exactly in the middle between the linear complexity of DeepSphere and the complexity  $\mathcal{O}(n^{3/2})$  of the SCNNs of Cohen and Esteves [4] [8]. In practice the number of neighbors grows very slowly with the number of pixels, making graph convolutions with  $G'$  still very efficient and fast (see section 5.1, table 5).

To conclude, we show in figure 20, 21 a confront between the alignment of the graph Laplacian eigenvectors of the DeepSphere graph  $G$ , the starting point of this work, and of the graph  $G'$ . It can be appreciated how the alignment plots show a much better behavior of the graph Laplacian eigenvectors of  $G'$ , and how the spectrum of  $G'$  resembles more accurately the spectrum of  $\Delta_{\mathbb{S}^2}$ .



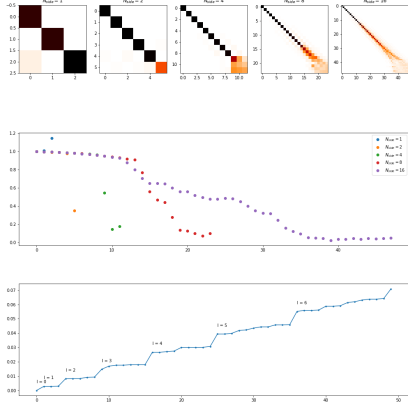


Figure 20: Alignment of the graph Laplacian eigenvectors of the DeepSphere graph  $G$ , the starting point of this work, and its spectrum.

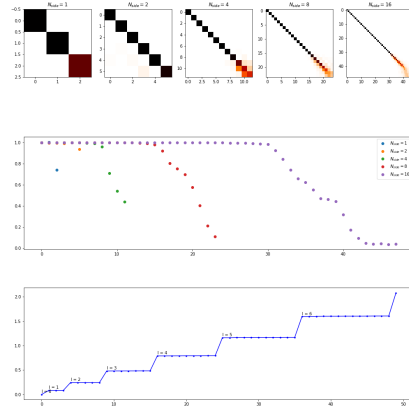


Figure 21: Alignment of the graph Laplacian eigenvectors of the proposed graph  $G'$ , and its spectrum.

### 3.3.5 Equivariance error

So far we used the plots of the alignment of the eigenvectors with the spherical harmonics as a proxy of the quantity we are really interested in, the mean equivariance error  $\overline{E}_G$ , because they gave us more valuable interpretations about what was happening. Now we want to have the confirmation that the proposed graph  $G'$  led to a smaller mean equivariance error than the one of the original graph  $G$  of DeepSphere. In figure 22 we plot the mean equivariance error

$$\overline{E} = \mathbb{E}_{f,g} E(f,g)$$

of the diffusion filter  $k(\lambda_i) = \exp(-\lambda_i)$  for both graphs  $G, G'$  by spherical harmonic degree  $\ell$  at different resolutions. This was obtained as the empirical average over a uniform sample of rotations  $g \in SO(3)$  and a uniform sample of functions  $f \in V_\ell = \text{span}\{Y_\ell^m, |m| \leq \ell\}$ . We recall that for HEALPix there's no sampling theorem that guarantees the existence of an exact reconstruction operator  $T^{-1}$ , so to calculate this quantity we had to rotate the sampled signal in the discrete domain, introducing important interpolation errors. We can see that DeepSphere has a mean equivariance error that is almost 30% in the low frequencies, and decreases slowly for higher ones. The graph  $G'$  has a much better behavior: the error stays low for the small frequencies, rising up for the higher ones and always remaining confined under 5%. To compare it we

reported also the results for the full HKGL, where we can appreciate a behavior similar to  $G'$  but with an error always smaller than 2%.

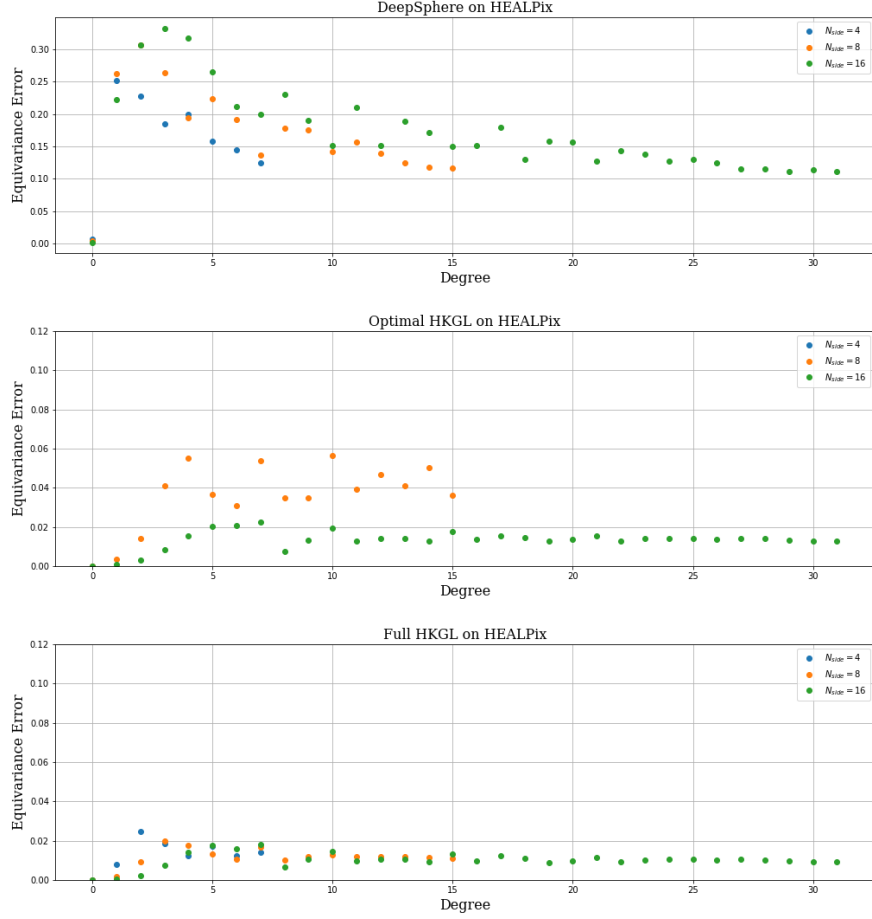


Figure 22: Mean equivariance error of the diffusion filter  $\exp(-\Lambda)$  for  $G$ ,  $G'$  and the full HKGL, by spherical harmonic degree. Notice the difference in the scale of the y axis for DeepSphere, that reaches errors up to 30%.

In figure 23 we see the visualization of the equivariance error of DeepSphere: on the left, the original sampled signal  $Tf$ . Top right, the rotated and filtered signal  $\Omega_k T\Lambda(g)f$ . Bottom right, the filtered and rotated signal  $T\Lambda(g)T^{-1}\Omega_k Tf$ . In figure 24 we see the visualization of the equivariance error of the graph  $G'$ : on the left, the same, original sampled signal  $Tf$ . Top right, the rotated and filtered signal  $\Omega_k T\Lambda(g)f$ . Bottom right, the filtered and rotated signal  $T\Lambda(g)T^{-1}\Omega_k Tf$ . No difference can be appreciated at a visual analysis for the graph  $G'$ , while for DeepSphere the difference is clearly visible.

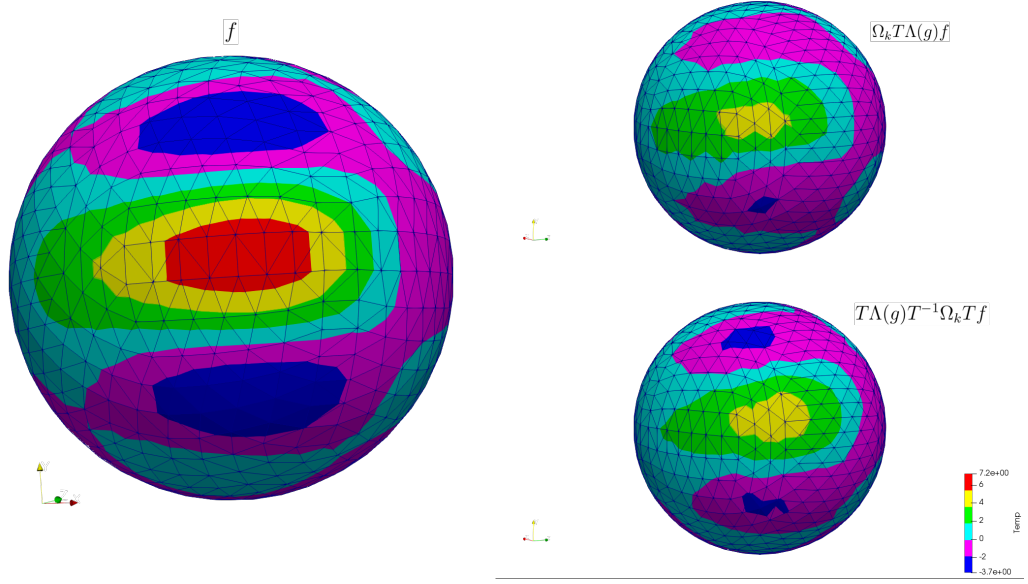


Figure 23: DeepSphere V1 equivariance error. On the left, a signal  $f$ . Top right,  $f$  was first rotated and then filtered through a diffusion filter  $k(\lambda) = \exp(-\lambda)$ . Bottom right,  $f$  was first filtered and then rotated. The difference in the two outcomes is evident.

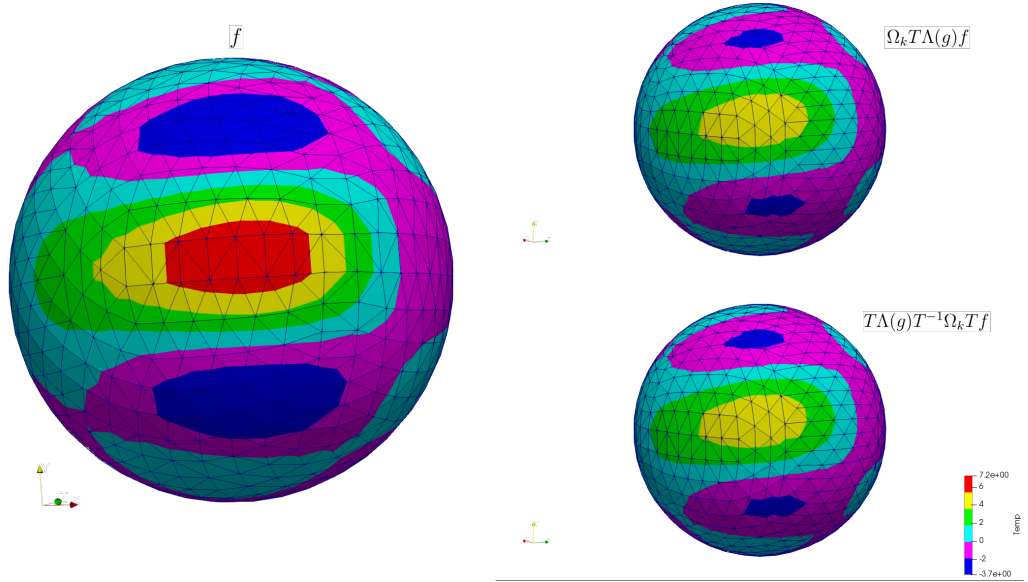


Figure 24: DeepSphere V2 equivariance error. On the left, a signal  $f$ . Top right,  $f$  was first rotated and then filtered through a diffusion filter  $k(\lambda) = \exp(-\lambda)$ . Bottom right,  $f$  was first filtered and then rotated. No difference in the two outcomes is visible to the human eye.

Mean equivariance error $\overline{E}$	$N_{side} = 4$	$N_{side} = 8$	$N_{side} = 16$
DeepSphere graph $G$	12.37%	12.03%	12.23%
Optimal graph $G'$	4.57%	3.98 %	1.54%

Table 2

To conclude, we report in table 2, as final metric for the evaluation of rotation equivariance of the two graphs  $G$ ,  $G'$ , the mean equivariance error for a different values of  $N_{side}$  that we computed by sampling random coefficients  $\theta_\ell^m \in (0, 1)$  of linear combinations of all the spherical harmonics up to degree  $\ell = 16$

$$f(x) = \sum_{\ell \leq 16, |m| \leq \ell} \theta_\ell^m Y_\ell^m(x)$$

and by averaging on random rotations  $g \in SO(3)$ . Our graph shows lower errors as  $N_{side}$  grows, and a much lower error than DeepSphere.

## 4 Non uniform sampling schemes and the FEM Laplacian

In Chapter 3, we used the fact that when the sampling scheme of the sphere is regular enough, the graph  $G$  is such that the corresponding graph Laplacian  $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$  has eigenvectors that are close enough to the ones of  $\Delta_{\mathbb{S}^2}$  to design a graph with a low mean equivariance error. We showed a way to construct a graph  $G'$  such that its graph Laplacian well approximates  $\Delta_{\mathbb{S}^2}$  in the case of an equiarea sampling scheme of the sphere and we tested it on the HEALPix sampling scheme. In this Chapter we focus on sampling schemes that are less uniform than HEALPix. The sampling scheme  $V = \{v_i \in \mathbb{S}^2\}$  that we will use for our study, very used in applications, is the so called *equiangular sampling scheme* [7]. This chapter is organized as follows: In section 4.1 we first introduce the equiangular sampling, and then we present the results that we obtained with two different Graph Laplacians: the HKGL, and a graph proposed by Khasanova et al. [9], specifically designed for this sampling scheme. In section 4.2 we deepen how to use the Finite Element Method (FEM) to construct a discrete approximation of  $\Delta_{\mathbb{S}^2}$  and how to derive a graph-like Laplacian from it, that shows a low equivariance error.

### 4.1 Graph Laplacian on the Equiangular Sampling

#### 4.1.1 The Equiangular Sampling

Given the usual parametrization  $x = x(\theta, \phi)$  of the sphere

$$\begin{aligned} \mathbb{S}^2 &= \left\{ x = (x_1, x_2, x_3) \in \mathbb{R}^3 : \|x\|_{\mathbb{R}^3} = (x_1^2 + x_2^2 + x_3^2)^{1/2} = 1 \right\} \\ x_1 &= \cos(\phi) \sin(\theta), \quad x_2 = \sin(\phi) \sin(\theta), \quad x_3 = \cos(\theta) \end{aligned}$$

Let  $m \in \mathbb{N}$ , the *equiangular sampling* of bandwidth  $b = 2^m$  is given by  $x_{jk}^{(b)} =$

$x(\theta_j^{(b)}, \phi_k^{(b)})$  where

$$\theta_j^{(b)} := \pi \frac{j}{2b}, \quad \phi_k^{(b)} := 2\pi \frac{k}{2b}$$

$$j = 0, \dots, 2b-1 \text{ and } k = 0, \dots, 2b-1$$

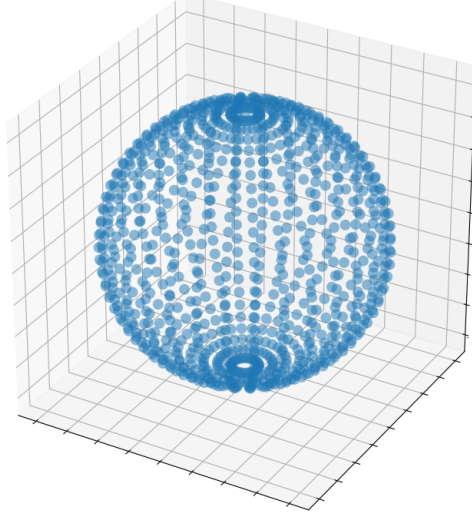


Figure 25: Equiangular sampling with bandwidth  $b = 8$

One has thus  $n = 4b^2$  points on the sphere, where all the points  $x_{0k}^{(b)}$  correspond to the north pole for every  $k = 0, \dots, 2b-1$ . Notice also that the south pole is never sampled. In figure 25 it can also be appreciated how the area close to the poles is much more sampled than the equator. One reason for which this sampling scheme is very used in application is the existence of the following result from [7], that states that any band limited function can be exactly recovered from its sampled values  $f(x_{jk}^{(b)})$ :

*Theorem 4.1.* Let  $l_0 \in \mathbb{N}$  and  $m_0 \in \mathbb{Z}$ ,  $|m_0| \leq l_0$ . If  $f = \sum_{l=0}^{b-1} \sum_{m=-l}^l \hat{f}(l, m) Y_l^m$  then

$$\begin{aligned} \hat{f}(l_0, m_0) = & \frac{1}{4b^2} \sum_{j=0}^{2b-1} \sum_{k=0}^{2b-1} f(x_{jk}^{(b)}) \overline{Y_{l_0}^{m_0}(x_{jk}^{(b)})} \sin(\theta_j^{(b)}) \times \\ & \times \frac{4}{\pi} \sum_{l=0}^{b-1} \frac{1}{2l+1} \sin((2l+1)\theta_j^{(b)}) \end{aligned}$$

Theorem 4.1 is the equivalent on the sphere of the well known Shannon's

sampling theorem, that states the minimum sampling frequency at which a band limited signal  $f : \mathbb{R} \rightarrow \mathbb{R}$  can be perfectly reconstructed, and is a precious tool when doing signal processing on the sphere.

#### 4.1.2 Heat Kernel Graph Laplacian on the equiangular sampling scheme

Thanks to theorem 4.1 not only we have a characterization of the space  $F = \{f = \sum_{l=0}^{b-1} \sum_{m=-l}^l \hat{f}(l, m) Y_l^m\} \subset L^2(\mathbb{S}^2)$  of band limited functions on which the sampling operator  $T : F \rightarrow \mathbb{R}^n$  is invertible, but we also an analytic expression for  $T^{-1}$ . Thanks to this we can calculate up to machine precision the equivariance error for any function  $f \in F$  and any rotation  $g \in SO(3)$ . In figure 26 we calculated the mean equivariance error of the full HKGL averaging on mono-frequency signals for different bandwidths. We can observe a much cleaner behavior of the mean equivariance error than the one we observed for the HEALPix sampling, most probably due to the fact that this time we can exactly calculate  $T^{-1}$ . The error increases linearly with the frequency of the signal for all bandwidths, and it seems to converge to zero as  $b$  increases. However, the spectrum of the graph Laplacian does not show the eigenvalues grouped in the expected pattern.

$b$	$t$
4	0.5
8	0.3
16	0.1

Table 3: Kernel width  $t$  used to construct the HKGL for each bandwidth  $b$

#### 4.1.3 A graph alternative to the HKGL for the equiangular sampling

Khasanova et al. [9] designed a discrete Laplacian that is explicitly intended to work on the sphere with the equiangular sampling. They studied a way to build a graph to analyze images produced by omnidirectional cameras. In their work they assume that the image is sampled on the sphere on the equiangular sampling. They consider the set  $\mathcal{G}$  of all the possible graphs where each node is

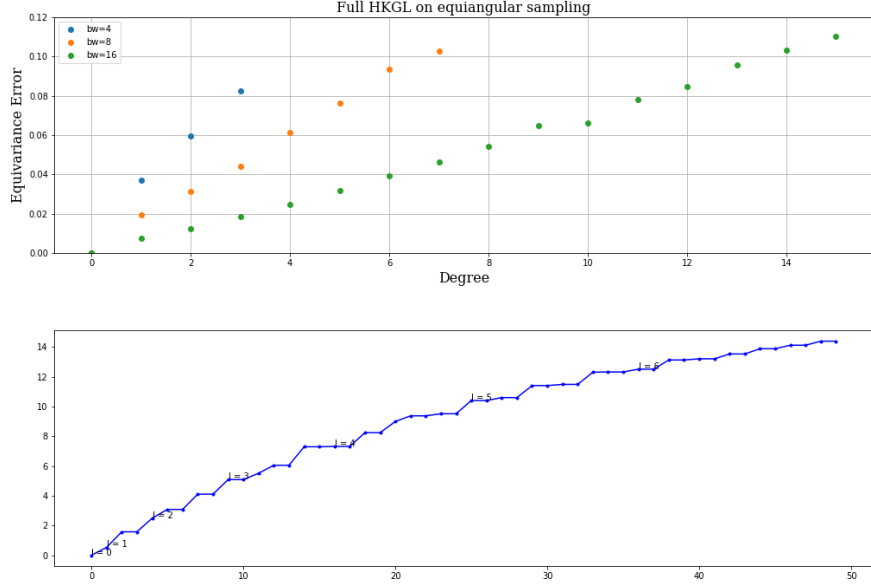


Figure 26: Equivariance error of the full HKGL on the equiangular sampling by spherical harmonic degree  $\ell$ , and its spectrum for the bandwidth  $b = 8$  .

connected only to four of its nearest neighbours (North, Sud, West, East) and propose a weighting scheme  $w_{ij}$  that minimizes the difference in the response to the polynomial spectral filter  $\mathcal{F} = \mathbf{L}$  evaluated on images of the same object seen at different latitudes. In other words, they solve the minimization problem

$$\min_{W \in \mathcal{G}} |\mathcal{F}(\mathbf{y}(v_e)) - \mathcal{F}(\mathbf{y}(v_i))| \quad (39)$$

for the adjacency matrix  $W$ , where  $\mathbf{y}(v_i)$  is the image of the object on the

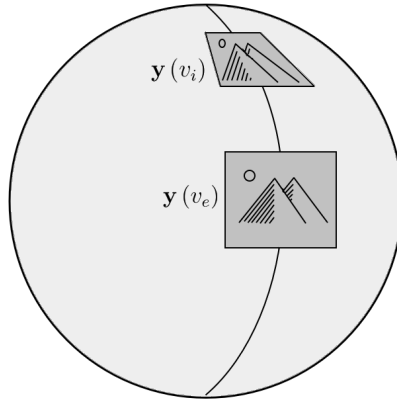


Figure 27: Khasanova et al. setting.



sphere centered on the vertex  $v_i$ , and  $\mathcal{F}(\mathbf{y}(v_e))$  is the response of the filter at the vertex  $v_e$  that lies at the same longitude of the vertex  $v_i$  but on the equator (figure 27). In their work they prove that the optimal weights solving the minimization problem (39) are given by weights  $w_{ij}$  inversely proportional to the Euclidean distance between vertices:

$$w_{ij} = \frac{1}{\|x_i - x_j\|} \quad (40)$$

This construction is interesting since it is adapted to the equiangular sampling, and leads to a very sparse graph with only 4 neighbors per vertex. Furthermore, to obtain the weights (40) every calculation was done in the *spatial domain*, without any consideration about the spectral interpretation of the filter. In order to compare it to the HKGL we show the equivariance error by spherical harmonic degree in figure 28. It can be appreciated how this construction performs a little worse than the full HKGL for low bandwidth samplings, but converges faster than the *full* HKGL, and its spectrum looks much more similar to the one of  $\Delta_{\mathbb{S}^2}$ .

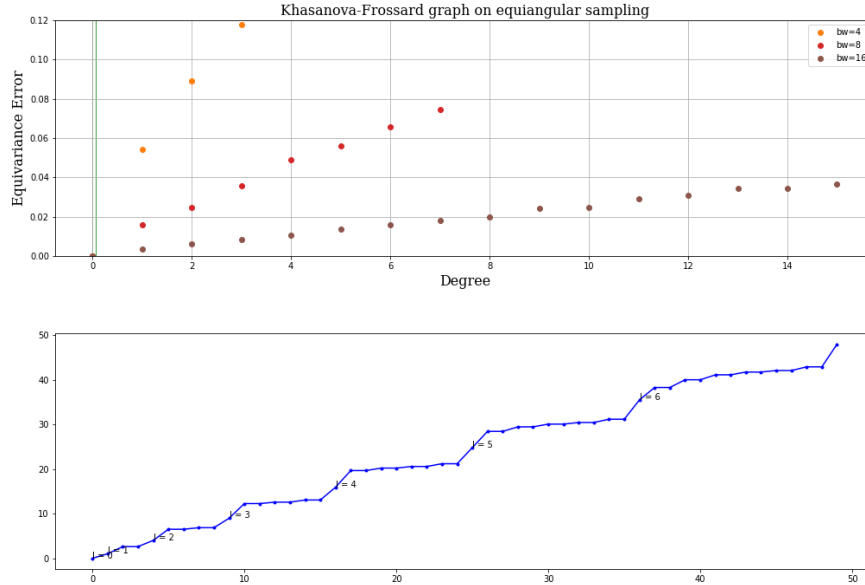


Figure 28: Equivariance error of the Khasanova-Frossard graph on the equiangular sampling with bandwidth  $b = 8$  by spherical harmonic degree  $\ell$ , and its spectrum.

The Khasanova-Frossard graph showed us that it is possible to do better than the HKGL on the equiangular sampling. We asked ourselves if there's a general

way of building a more equivariant graph than the HKGL that is straightforward to implement on any sampling scheme of the sphere. To answer this question in the next section we study a more complex way to approximate the Laplace-Beltrami operator  $\Delta_{\mathbb{S}^2}$ : the linear FEM Laplacian. We will see that the comparison between the FEM Laplacian and the graph Laplacian will give us precious insights to better understand the limitations of graph Laplacians when it comes to deal with non uniform samplings of the sphere.

## 4.2 The Finite Element Method approximation of the Laplace-Beltrami operator on the sphere

The Finite Element Method (FEM) is a numerical algorithm that allows to calculate a discrete approximation of the solution of the Laplace-Beltrami eigenvalue problem through a functional discretization of the differential operator  $\Delta_{\mathbb{S}^2}$ . We put the necessary definitions and mathematical concepts necessary to properly introduce the weak formulation of a differential problem, the Galerkin method and finally the Finite Element Method in Appendix. For a detailed introduction to the FEM, we refer the reader to [19]

Let's transform the strong form of the eigenvalue problem on the Sphere on its *weak* formulation. Let's multiply equation (24) by a sufficiently regular function  $v$  and integrate on  $\mathbb{S}^2$ . Since the sphere is a closed manifold and has no border, integrating by parts yields

$$\begin{aligned} &\text{Find } f \in H^1(\mathbb{S}^2), \lambda \in \mathbb{R} \text{ such that} \\ &\int_{\mathbb{S}^2} \nabla f(\mathbf{x}) \cdot \nabla v(\mathbf{x}) d\mathbf{x} = \lambda \int_{\mathbb{S}^2} f(\mathbf{x}) \cdot v(\mathbf{x}) d\mathbf{x} \quad \forall v \in H^1(\mathbb{S}^2) \end{aligned} \tag{41}$$

where  $v$  has been chosen to belong to  $H^1 \subset L^2(\mathbb{S}^2)$ , the Sobolev space of all the functions with derivative  $\nabla f \in L^2(\mathbb{S}^2)$ . The Finite Element Method first approximates the domain  $\Omega$  with a triangulation  $\mathcal{T}_h = \{\tau_i\}$ , and then *projects* the weak problem (41) into a finite dimensional subspace  $V_h \subset H^1(\mathcal{T}_h)$  made by all the piecewise linear polynomials on the triangulation  $\mathcal{T}_h$ . Being the sphere convex, the triangulation  $\mathcal{T}_h$  has been obtained by calculating the triangulation of the convex hull of the vertices of the chosen sampling scheme through the Qhull algorithm [1]. Now, define  $X_h^1$  to be the space of all the continuous,

piecewise linear functions on  $\Omega$

$$X_h^1 = \{v_h : v_h \in C(\Omega) : v_h|_{\tau} \in \mathbb{P}^1 \ \forall \tau \in \mathcal{T}_h\}$$

and set  $V_h = X_h^1$ . Since for the functions in  $X_h^1$  the number of degrees of freedom is the same of the number of vertices of the mesh  $n$ , we need  $n$  basis functions  $\phi_i, i = 0, \dots, n-1$  to fully describe  $X_h^1$ .  $\phi_i$  is defined as the continuous piecewise linear function such that

$$\phi_i(x_j) = \delta_{ij} \quad i = 0, \dots, n-1$$

where  $x_i$  are the points of the sampling scheme that have been used as vertices of the triangles  $\tau$  of the triangulation  $\mathcal{T}_h$ . The support of  $\phi_i$  i.e., the subset of  $\mathcal{T}_h$  where  $\phi_i$  is not zero, is made by all the triangles sharing the  $i$ th vertex. An example is shown in figure 30.

*Remark.* For a function  $v_h \in X_h^1$ ,  $v_h = v_0\phi_0 + \dots v_n\phi_n$  the coefficient  $v_i$  is equal to the function  $v_h$  evaluated in the  $i$ th vertex

$$v_i = v_h(x_i) \tag{42}$$

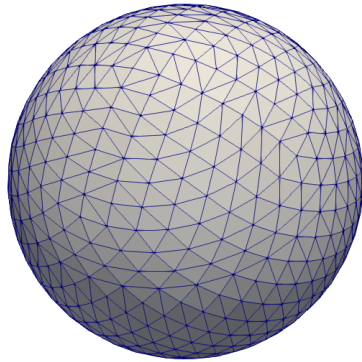


Figure 29: A triangulation  $\mathcal{T}_h$  of the sphere made with the vertices of the HEALPix sampling scheme with  $N_{side} = 8$ .

By writing  $n$  times the equation (41), setting each time the test function  $v_h$  equal to the  $i$ th basis function  $\phi_i$  of the space  $X_h^1$ , we obtain the generalized

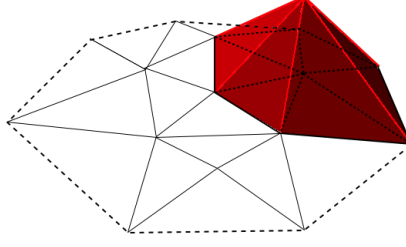


Figure 30: Basis function of the space  $X_h^1$

algebraic eigenvalue problem

Find  $(f, \lambda)$  such that  $\mathbf{A}\mathbf{f} = \lambda\mathbf{B}\mathbf{f}$

$$\begin{cases} (\mathbf{A})_{ij} &= \int_{\mathbb{S}^2} \nabla \phi_i(\mathbf{x}) \cdot \nabla \phi_j(\mathbf{x}) d\mathbf{x} \\ (\mathbf{B})_{ij} &= \int_{\mathbb{S}^2} \phi_i(\mathbf{x}) \phi_j(\mathbf{x}) d\mathbf{x} \\ (\mathbf{f})_i &= f_i : \quad f(\mathbf{x}) = f_0 \phi_0(\mathbf{x}) + \dots + f_{n-1} \phi_{n-1}(\mathbf{x}) \end{cases} \quad (43)$$

$\mathbf{A}$  is called the *stiffness* matrix, and  $\mathbf{B}$  is called the *mass* matrix. Observe that being the Laplace-Beltrami operator self-adjoint, we have that  $\mathbf{A} = \mathbf{A}^\top$ ,  $\mathbf{B} = \mathbf{B}^\top$ . Being  $\mathbf{B}$  non singular, the system (43) is equivalent to the eigenvalue problem

$$\mathbf{B}^{-1}\mathbf{A}\mathbf{f} = \lambda\mathbf{f} \quad (44)$$

It can be shown [25] that even though the matrix  $\mathbf{B}^{-1}\mathbf{A}$  is not symmetric, its eigenvalues are still real and its eigenvectors are such that  $\mathbf{V}\mathbf{B}\mathbf{V}^\top = \mathbf{I}$ , where  $\mathbf{I}$  is the identity matrix. The solution  $\mathbf{f}$  that is the vector of the coefficients of the function  $f_h$  in the basis  $\phi_i$  corresponds exactly to the values of the function  $f_h$  in the vertices.

### 4.3 How to filter a signal with the linear FEM

Calculating the discrete Fourier transform with the linear FEM means projecting the Fourier transform into the subspace  $V_h$ :

$$\hat{f}_{FEM}(\ell, m) = \int_{\eta \in \mathcal{T}_h} f_h(\eta) v_{i(\ell, m)}(\eta) d\eta = \mathbf{v}_{i(\ell, m)}^\top \mathbf{B}\mathbf{f} \quad (45)$$

where  $v_{i(\ell, m)}$  is the solution to the eigenvalue problem (43),  $f_h$  is the projection

of  $f$  on  $V_h$ , and  $\mathbf{B}$  is the mass matrix. It follows that the filtering of a discretized signal  $\mathbf{f}$  in the spectral domain is done by the following matrix  $\Omega_K^{FEM}$ :

$$\Omega_K^{FEM} := (\mathbf{B}\mathbf{V}^\top)^{-1}\mathbf{K}\mathbf{V}^\top\mathbf{B} \quad (46)$$

where  $\mathbf{V}^\top\mathbf{B}$  is the FEM Fourier transform matrix,  $\mathbf{K}$  is the diagonal matrix that represent the chosen kernel for this filter, and  $(\mathbf{V}^\top\mathbf{B})^{-1}$  is the matrix representing the inverse FEM Fourier transform. We can already notice a fundamental difference between this way of filtering a signal and the way of filtering a signal with a graph. The FEM filtering uses not only the eigenvectors of the FEM Laplacian, but also the mass matrix  $\mathbf{B}$ . We computed the usual mean equivariance error per spherical harmonic degree for the diffusion filter  $\mathbf{K} = \exp(-\mathbf{\Lambda})$  on the equiangular sampling, and we show it in figure 31. We

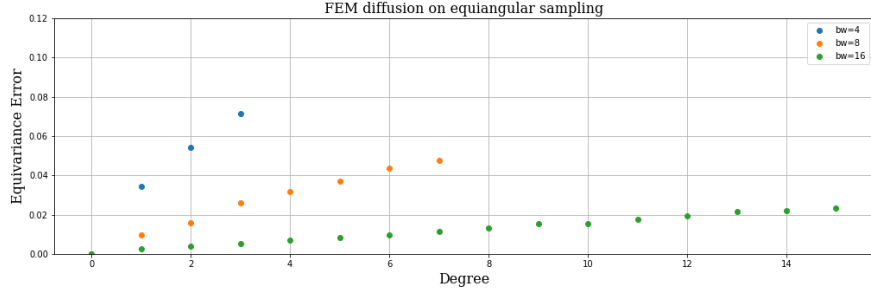


Figure 31: Equivariance error of the linear FEM diffusion on the equiangular sampling by spherical harmonic degree.

can see that the FEM filtering works better than the state-of-the-art graph of Khasanova-Frossard, but the matrix  $\Omega_K^{FEM}$  is full, and its computation requires the expensive inversion of the Fourier matrix  $\mathbf{V}^\top\mathbf{B}$ , making it not efficient if used in a CNN.

#### 4.3.1 A confront between FEM filtering and graph filtering

Due to the fact that  $\mathbf{V}^\top\mathbf{B}\mathbf{V} = \mathbf{I}$ , we have that

$$\begin{aligned} (\mathbf{V}^\top\mathbf{B})^{-1} &= \mathbf{V}, \\ \mathbf{V}^\top\mathbf{B} &= \mathbf{V}^{-1}. \end{aligned}$$

So, the FEM filter matrix  $\Omega_K^{FEM}$  can be rewritten as

$$\Omega_K^{FEM} = \mathbf{V}\mathbf{K}\mathbf{V}^{-1} \quad (47)$$

Thus a polynomial filter

$$k(\lambda) = P_\kappa(\lambda) = \sum_{k=0}^{\kappa-1} \theta_k \lambda^k$$

would be implemented with a matrix

$$\Omega_K^{FEM} = \mathbf{V} \left( \sum_{k=0}^{\kappa-1} \theta_k \mathbf{\Lambda}^k \right) \mathbf{V}^{-1} = \sum_{k=0}^{\kappa-1} \theta_k (\mathbf{B}^{-1} \mathbf{A})^k = P_\kappa(\mathbf{B}^{-1} \mathbf{A}).$$

It is interesting to notice that FEM filtering (47) looks very much like graph filtering (20) but, given that  $V$  is not orthogonal anymore, it has to replace the  $\mathbf{V}^\top$  with  $\mathbf{V}^{-1}$ . Equation 47 implies that the FEM filtering could be implemented exactly as the graph filtering already implemented in DeepSphere, using the FEM matrix  $\mathbf{B}^{-1} \mathbf{A}$  instead of the usual symmetric HKGL  $\mathbf{L}_n^t$ . Unfortunately, in order to make the evaluation of the polynomial  $P_\kappa(\mathbf{B}^{-1} \mathbf{A})$  efficient, we need the matrix  $\mathbf{B}^{-1} \mathbf{A}$  to be sparse. The only way to do so is to have a *sparse stiffness matrix*  $\mathbf{A}$  and a *diagonal mass matrix*  $\mathbf{B}$ . Unfortunately there is no likely way to have both these conditions satisfied at the same time, since this is one of the most well known trade offs of the FEM [25]. To make  $\mathbf{B}$  diagonal we need to choose  $\phi_i$  to be an orthogonal basis of the finite dimensional functional space  $V_h$ . To do so, we will need to choose more complicated basis functions than the usual ones in figure 30, *most likely with a support that extends to the whole sphere*, making the stiffness matrix  $\mathbf{A}$  not sparse anymore. A common workaround [25] is the so called *lumping* of the mass matrix, that consists in replacing the matrix  $\mathbf{B}$  with the *lumped* diagonal matrix  $\mathbf{D}$  obtained by placing in each diagonal entry  $(\mathbf{D})_{ii}$  the sum of the elements of the  $i$ th row of the mass matrix  $\mathbf{B}$ :

$$\mathbf{D} = \text{diag}\{d_i\}, \quad d_i = \sum_j (\mathbf{B})_{ij} \quad (48)$$

This approximation is well studied in the literature and it is proved to work well in many practical cases [19]. In this way the FEM matrix  $\mathbf{B}^{-1} \mathbf{A}$  would

be approximated by the matrix

$$\mathbf{D}^{-1}\mathbf{A}.$$

This matrix has the big advantage of having the same sparsity pattern of the stiffness matrix  $\mathbf{A}$ , making it very efficient for polynomial filtering. We can take one step further by using the symmetric matrix

$$\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}.$$

To confront FEM and HKGL filtering we filter a unit mass signal  $\mathbf{f} = (0, 0, \dots, 0, 1, 0, \dots, 0)^\top$  with the *diffusion* filter  $\exp(-\tau\mathbf{L})$ . It was chosen a very irregular sampling scheme, shown in figure 32. It can be seen how the full HKGL compresses the signal around the equator where the sampling scheme is more sparse; on the other hand the FEM filtering manages to keep the diffusion homogeneous no matter the asymmetry in the sampling scheme. The mean equivariance error using by spherical harmonic degree can be found in figure 33. It can be seen how  $\mathbf{D}^{-1}\mathbf{A}$  has almost the same equivariance error of the full FEM Laplacian  $\mathbf{B}^{-1}\mathbf{A}$ , while using the symmetric matrix  $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$  makes the equivariance error much worse, especially at lower frequencies.

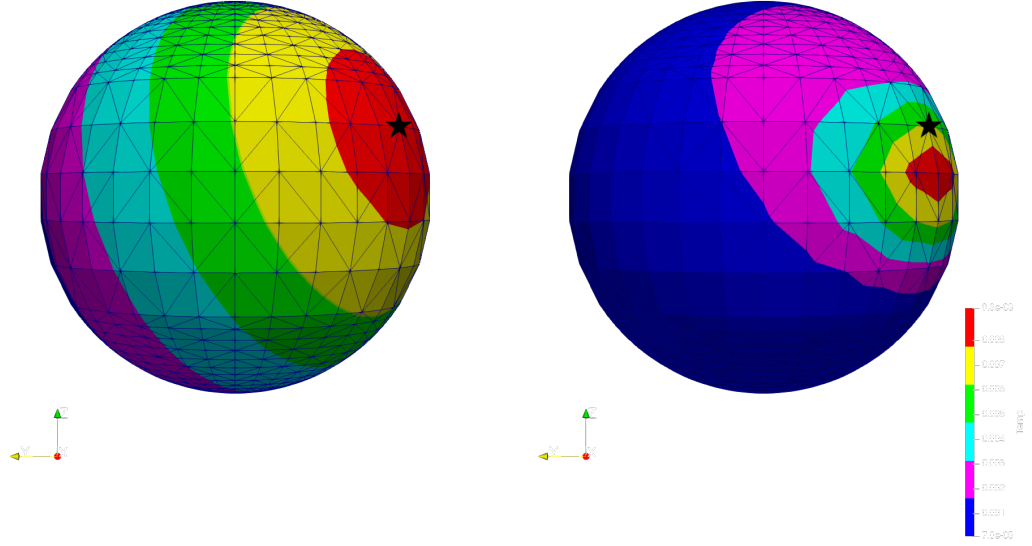


Figure 32: Symmetric lumped linear FEM diffusion and HKGL diffusion on an irregular sampling scheme of the sphere. The position of the filtered source signal is indicated with a black star.

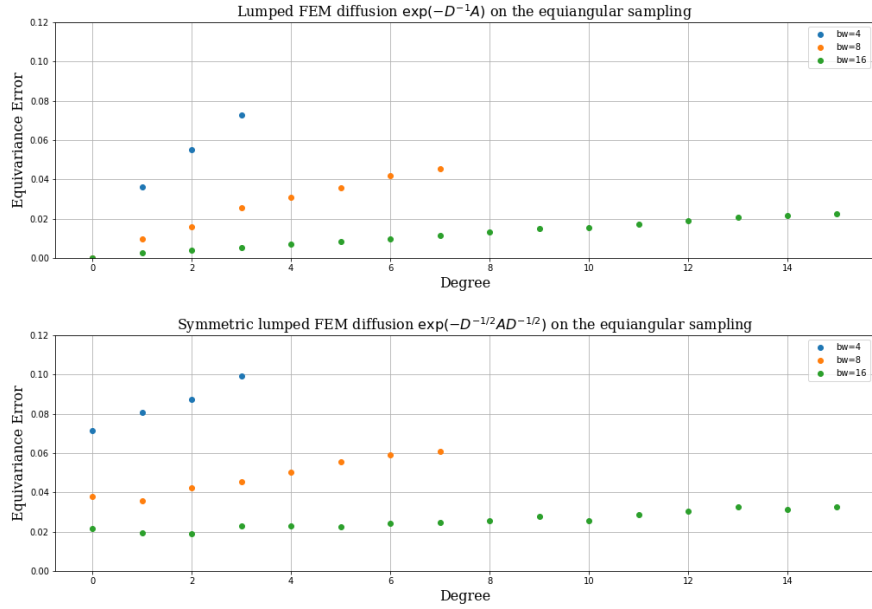


Figure 33: Equivariance error of the lumped FEM Laplacian and of the symmetric lumped FEM Laplacian on the equiangular sampling by spherical harmonic degree  $\ell$ .



**Accuracy of the linear FEM spherical harmonics.** Thanks to the sampling theorem 4.1, we are able to compute the exact SHT (under the hypothesis of band limited signals) of the solutions to the eigenvalue problem (43) and we show in figures 34, 35 the power spectrum of the eigenmodes for both HEALPix sampling and the equiangular sampling, as a measure of the goodness of the linear FEM approach to approximate the spherical harmonics.

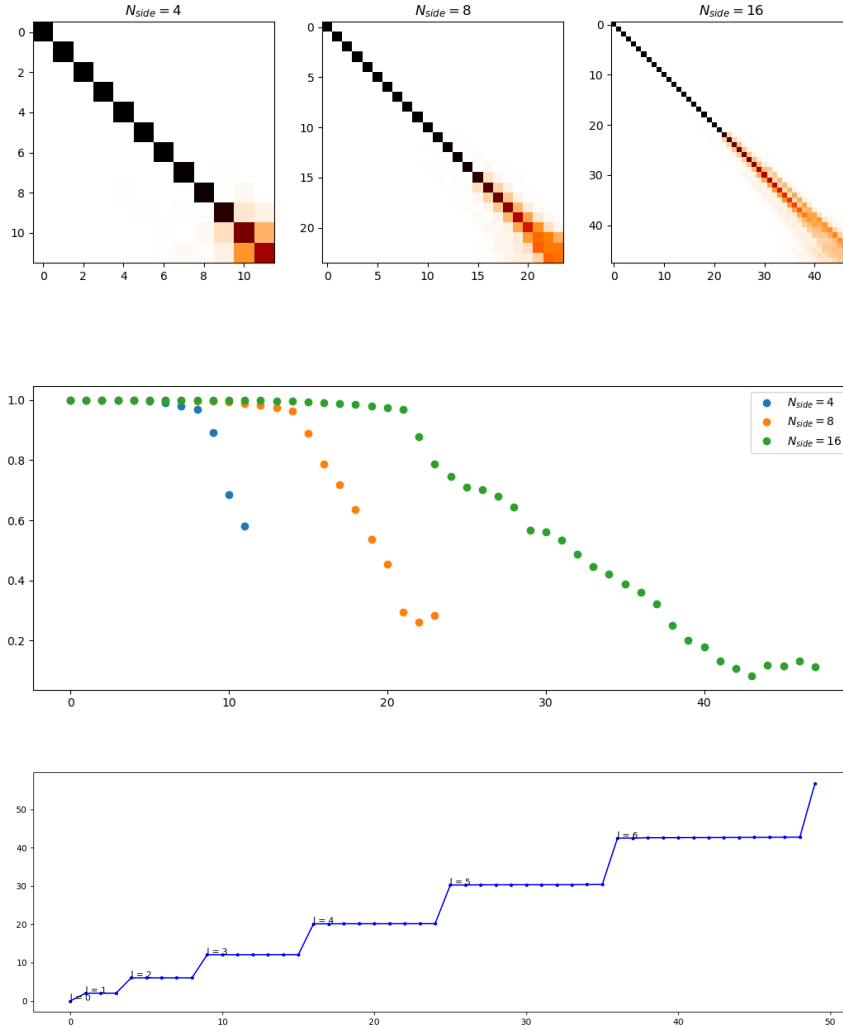


Figure 34: Alignment of eigenspaces of the linear FEM Laplacian on HEALPix, and its spectrum for  $N_{side} = 16$

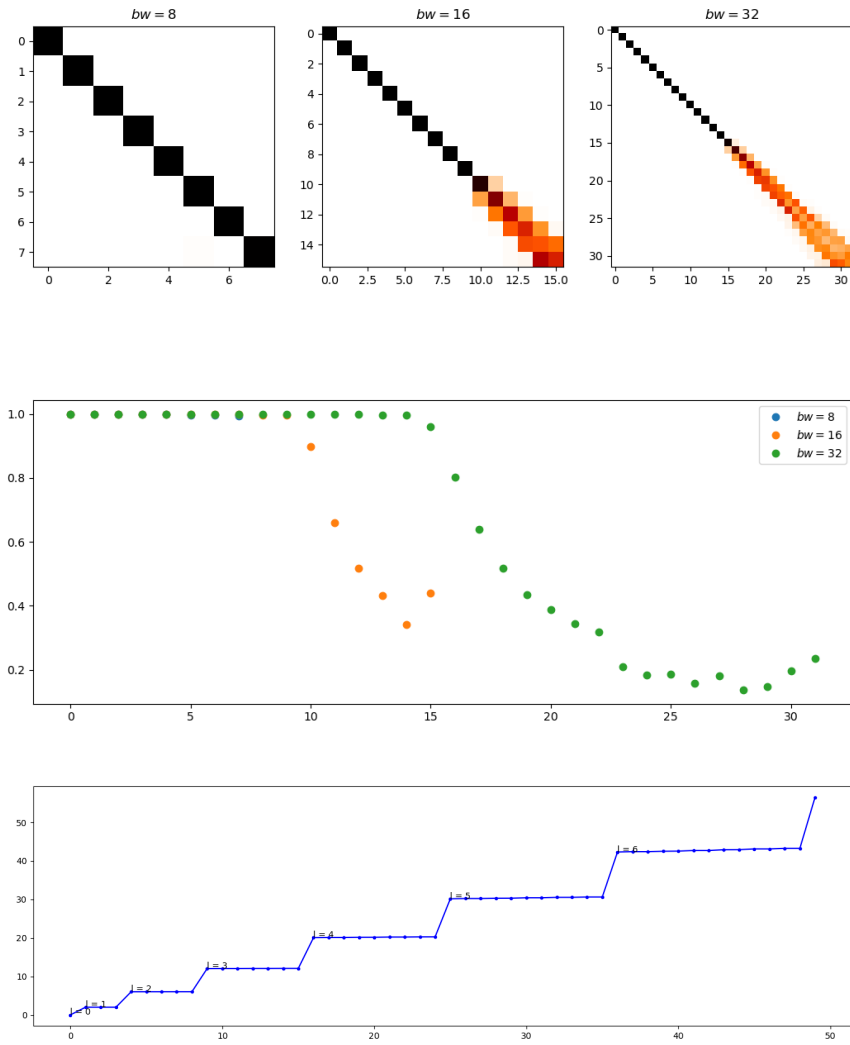


Figure 35: Alignment of eigenspaces of the linear FEM Laplacian on the equiangular sampling, and its spectrum for  $bw = 32$

## 5 Conclusions

### 5.1 Experimental validation: SHREC17

Gusset [11] implemented the graph proposed in Chapter 2 in a GCNN and three other rotation equivariant neural networks on a popular classification problem [22]. The four models tested were the following: the original version of DeepSphere, Deepsphere *Optimal* - obtained implementing the thresholding procedure described in this work in section 3.3 - and the traditional SCNNs of Cohen et al. and of Esteves et al. [4] [8].

**On the Equiangular sampling.** Gusset compares with two different metrics (accuracy, F1-score) the performances of these four rotation invariant models, while also comparing the speed of inference and of training of each model. Results are shown in table 4. It can be seen how *DeepSphere Optimal* has *always* the highest score between all the rotation equivariant models, no matter the evaluation metric. Furthermore, its performances in terms of speed of inference and training are second only to DeepSphere, remaining by far faster than the other two SCNNs.

Method	performance		size	speed	
	Accuracy	F1-score	params	inference	training
Cohen <i>s2cnn_simple</i>	78.59	78.85	400k	12ms	32h
Esteves <i>sphericalcnn</i>	79.18	79.36	500k	9.8ms	2h52
Deepsphere	73.36	73.67	190k	<b>0.98ms</b>	<b>43m</b>
<b>Deepsphere <i>Optimal</i></b>	<b>80.42</b>	<b>80.65</b>	190k	1.0ms	48m

Table 4: Results from [11]. Performances of four rotation equivariant GCNNs and two SCNNs on the popular classification task SHREC17.

**On HEALPix** Gusset repeated the same test on the same dataset, this time sampled using the HEALPix sampling scheme with  $N_{side} = 32$ . Results can be seen in table 5. Being the new graph of DeepSphere Optimal more equivariant to rotations, we expected to see an improvement in the accuracy, as we did in the equiangular case. The fact that this improvement was not observed means that, with this sampling, the original DeepSphere graph *W* is *already sufficiently equivariant to rotations*.

	DeepSphere	DeepSphere <i>Optimal</i>
accuracy	82.23%	82.76%

Table 5: Results from Gusset et al. Accuracy on the HEALPix sampling

## 5.2 Confront of different Discrete Laplacians on the equiangular sampling

We conclude by showing how the different discrete Laplacians  $\mathbf{L}$  illustrated so far compare in terms of equivariance error and computational time of the filter  $\mathcal{F}(\mathbf{f}) = \mathbf{L}\mathbf{f}$ . We can see how the four sparse discrete Laplacians are one order of magnitude faster than the two full Laplacians. The FEM Laplacian is able to reduce the equivariance error of the HKGL, and it manages to keep it low - around 0.5% - even when using the sparse, lumped approximation  $\mathbf{D}^{-1}\mathbf{A}$  while reducing the computational time of one order of magnitude.  $\mathbf{D}^{-1}\mathbf{A}$  performs really well, and gets close to the performances of the graph Laplacian of Khasanova and Frossard.

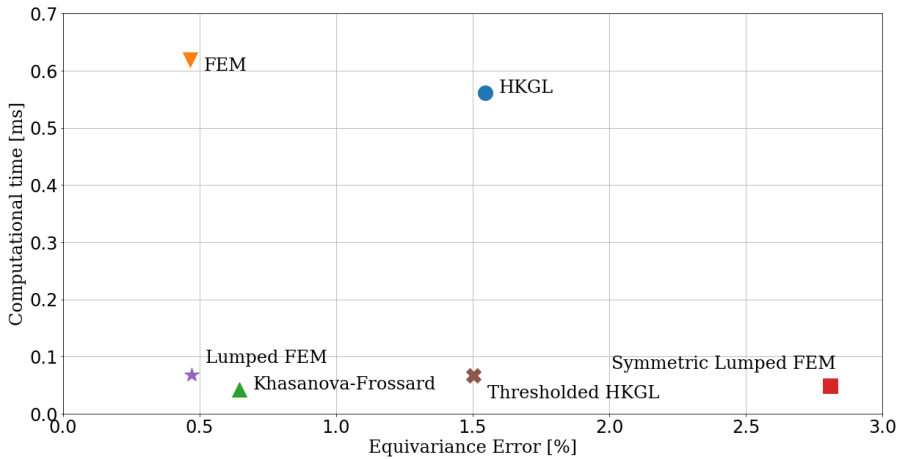


Figure 36: Trade-off between computational time and equivariance error for the filter  $\mathbf{L}$  for different discrete Laplacians on the equiangular sampling

## 5.3 Final considerations and future work

In order not to confuse the notation between the FEM and the Graph approach we will need a more precise notation than in the rest of this work. For this

	FEM	HKGL	Lumped FEM	Symmetric Lumped FEM	Thresholded HKGL	Khasanova Frossard
Equivariance error [%]	0.46	1.55	0.47	2.80	1.50	0.64
Computational time [ $\mu$ s]	619	561	68	49	67	42

Table 6: Results of figure 36

purpose, define a graph  $G$ , and its graph Laplacian by  $L_G$ . Define  $\mathbf{V}_G$ ,  $\mathbf{\Lambda}_G$  to be the solution of the eigenvalue problem

$$\mathbf{L}_G \mathbf{V}_G = \mathbf{V}_G \mathbf{\Lambda}_G.$$

Define the FEM stiffness matrix  $\mathbf{A}$ ,  $(\mathbf{A})_{ij} = \int \nabla \phi_i \nabla \phi_j$  and the FEM mass matrix  $\mathbf{B}$ ,  $(\mathbf{B})_{ij} = \int \phi_i \phi_j$ . Define  $\mathbf{V}_{FEM}$  and  $\mathbf{\Lambda}_{FEM}$  to be the solution to the generalized eigenvalue problem

$$\mathbf{A} \mathbf{V}_{FEM} = \mathbf{B} \mathbf{V}_{FEM} \mathbf{\Lambda}_{FEM}.$$

We saw that both in the graph and in the FEM approach, filtering a sampled signal means approximating the Fourier transform through the multiplication of the signal  $\mathbf{f}$  by a Fourier matrix -  $\mathbf{V}_G^T$  for the graph,  $\mathbf{V}_{FEM}^T \mathbf{B}$  for the FEM - then applying a filter through a diagonal matrix  $\mathbf{K}$ , and then applying the inverse Fourier transform -  $\mathbf{V}_G$  for the graph,  $(\mathbf{V}_{FEM}^T \mathbf{B})^{-1}$  for the FEM -. From these considerations it follows that a polynomial filter  $P_\kappa(\mathbf{\Lambda})$  is implemented in the graph domain by multiplying the signal  $\mathbf{f}$  by a polynomial of the symmetric graph Laplacian  $\mathbf{L}_G$

$$P_\kappa(\mathbf{L}_G),$$

and in the FEM domain (thanks to what explained in section 4.3.1) by a polynomial of the matrix  $\mathbf{B}^{-1} \mathbf{A}$

$$P_\kappa(\mathbf{B}^{-1} \mathbf{A}).$$

Starting from the FEM Laplacian  $\mathbf{B}^{-1} \mathbf{A}$  it is possible to construct a *sparse* Laplacian  $\mathbf{D}^{-1} \mathbf{A}$  that shows almost no difference in its equivariance error compared to the full FEM Laplacian. Levy [14] showed that, by explicitly solving

the integrals  $\int_{\tau} \phi_i \phi_j$ ,  $\int_{\tau} \nabla \phi_i \cdot \nabla \phi_j$ ,

$$\mathbf{D}_{ii} = \frac{A_i}{3},$$

$$\mathbf{A}_{ij} = \frac{1}{2} (\cot(\alpha_{ij}) + \cot(\beta_{ij}))$$

where  $A_i$  is the sum of the area of all the triangles  $\tau$  of the triangulation  $\mathcal{T}_h$  sharing the  $i$ th vertex, proving that the lumped FEM Laplacian  $\mathbf{D}^{-1}\mathbf{A}$  corresponds to the Laplacian of Desbrun et al. [6] introduced in this work in section 2. In this way he connected the FEM approach to Laplacians obtained from Differential Geometry and Discrete Exterior Calculus [26], [15].

The symmetric graph Laplacian  $\mathbf{L}_G$  constrains the graph Fourier matrix  $\mathbf{V}_G^{\top}$  to be orthogonal, while the FEM Laplacian  $\mathbf{B}^{-1}\mathbf{A}$  leaves to its Fourier matrix  $\mathbf{V}_{FEM}^{\top}\mathbf{B}$  more degrees of freedom. The fact that the mass matrix  $\mathbf{B}$  is constructed to *exactly* represent the dot product in the Galerkin subspace  $V_h$  and that the matrix  $\mathbf{V}_{FEM}$  converges to the sampled spherical harmonics makes it possible for the FEM filtering to converge towards the continuous filtering even in cases of non uniform sampling measures [19], while for symmetric Laplacians there is no convergence result available.

How to interpret the fact that the graph approach constrains the Fourier matrix  $\mathbf{V}_G^{\top}$  to be orthogonal is still not clear and will be subject of future work. However, even given these orthogonality constraints it is sometimes possible to design graphs with state-of-the-art performances, like the Khasanova-Frossard graph for the equiangular sampling scheme. Notice that this graph was obtained solving the optimization problem (39) formulated directly in the spatial (vertex) domain, without relying on the spectral interpretation of the graph filtering, that in the case of a non uniform sampling presents the problems discussed above and it is still not clear.

## References

- [1] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM TRANSACTIONS ON MATHEMATICAL SOFTWARE*, 22(4):469–483, 1996.
- [2] Mikhail Belkin and Partha Niyogi. Towards a theoretical foundation for laplacian-based manifold methods. In *Proceedings of the 18th Annual Conference on Learning Theory*, COLT’05, pages 486–500, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3-540-26556-2, 978-3-540-26556-6. doi: 10.1007/11503415\_33. URL [http://dx.doi.org/10.1007/11503415\\_33](http://dx.doi.org/10.1007/11503415_33).
- [3] Mikhail Belkin and Partha Niyogi. Convergence of laplacian eigenmaps. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 129–136. MIT Press, 2007. URL <http://papers.nips.cc/paper/2989-convergence-of-laplacian-eigenmaps.pdf>.
- [4] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *CoRR*, abs/1801.10130, 2018. URL <http://arxiv.org/abs/1801.10130>.
- [5] B.C. Csaji. Approximation with artificial neural networks. *M.S.’Thesis, Dept. Science, Eotvos Lorand Univ., Budapest, Hungary*, 2001. URL <https://ci.nii.ac.jp/naid/20001716508/en/>.
- [6] M. Desbrun, M. Meyer, P. Schroder, and A.H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. *Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow*, 1999. cited By 3.
- [7] J. R. Driscoll and D. M. Healy. Computing fourier transforms and convolutions on the 2-sphere. *Adv. Appl. Math.*, 15(2):202–250, June 1994. ISSN 0196-8858. doi: 10.1006/aama.1994.1008. URL <http://dx.doi.org/10.1006/aama.1994.1008>.
- [8] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. 3d object classification and retrieval with spherical cnns. *CoRR*, abs/1711.06721, 2017. URL <http://arxiv.org/abs/1711.06721>.
- [9] Pascal Frossard and Renata Khasanova. Graph-based classification of omnidirectional images. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 860–869, 2017.
- [10] Krzysztof Gorski, Eric Hivon, AJ Banday, Benjamin Wandelt, F K. Hansen, M Reinecke, and M Bartelman. Healpix: A framework for high-resolution

- discretization and fast analysis of data distributed on the sphere. *apj*, 622, 09 2004. doi: 10.1086/427976.
- [11] Frederik Gusset, Michaël Defferrard, Nathanaël Perraudin, and Pierre Vanderghelynst. Spherical convolutional neural networks: empirical analysis of sc-nns. 2019.
  - [12] Vassilis Kalofolias. How to learn a graph from smooth signals. In *Artificial Intelligence and Statistics*, pages 920–929, 2016.
  - [13] Vassilis Kalofolias and Nathanaël Perraudin. Large scale graph learning from smooth signals. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=ryGkSo0qYm>.
  - [14] Bruno Levy. Laplace-beltrami eigenfunctions: Towards an algorithm that understand s geometry. In *IEEE International Conference on Shape Modeling and Applications, inv ited talk*, 2006.
  - [15] Mark Meyer, Mathieu Desbrun, Peter Schroder, and Alan H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. pages 35–57. Springer-Verlag, 2002.
  - [16] Mark Meyer, Mathieu Desbrun, Peter Schroeder, and Alan H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. pages 35–57. Springer-Verlag, 2002.
  - [17] Bastien Paskdeloup, Vincent Gripon, Grégoire Mercier, Dominique Pastor, and Michael G Rabbat. Characterization and inference of graph diffusion processes from observations of stationary signals. *IEEE transactions on Signal and Information Processing over Networks*, 4(3):481–496, 2018.
  - [18] Nathanaël Perraudin, Michaël Defferrard, Tomasz Kacprzak, and Raphael Sgier. Deepsphere: Efficient spherical convolutional neural network with healpix sampling for cosmological applications. *CoRR*, abs/1810.12186, 2018. URL <http://arxiv.org/abs/1810.12186>.
  - [19] Alfio Quarteroni. *Numerical models for differential problems*. MS&A. Springer, Milano, 2009. doi: 10.1007/978-88-470-1071-0. URL <https://cds.cern.ch/record/1639539>.
  - [20] Martin Reuter, Silvia Biasotti, Daniela Giorgi, Giuseppe Patanas, and Michela Spagnuolo. Discrete laplace beltrami operators for shape analysis and segmentation. *Computers and Graphics*, 33(3):381 – 390, 2009. ISSN 0097-8493. doi:



- <https://doi.org/10.1016/j.cag.2009.03.005>. URL <http://www.sciencedirect.com/science/article/pii/S0097849309000272>. IEEE International Conference on Shape Modelling and Applications 2009.
- [21] Steven Rosenberg. *The Laplacian on a Riemannian Manifold: An Introduction to Analysis on Manifolds*. London Mathematical Society Student Texts. Cambridge University Press, 1997. doi: 10.1017/CBO9780511623783.
- [22] M. Savva, F. Yu, Hao Su, M. Aono, B. Chen, D. Cohen-Or, W. Deng, Hang Su, S. Bai, X. Bai, N. Fish, J. Han, E. Kalogerakis, E. G. Learned-Miller, Y. Li, M. Liao, S. Maji, A. Tatsuma, Y. Wang, N. Zhang, and Z. Zhou. Large-scale 3d shape retrieval from shapenet core55. In *Proceedings of the Eurographics 2016 Workshop on 3D Object Retrieval*, 3DOR '16, pages 89–98, Goslar Germany, Germany, 2016. Eurographics Association. ISBN 978-3-03868-004-8. doi: 10.2312/3dor.20161092. URL <https://doi.org/10.2312/3dor.20161092>.
- [23] Rasoul Shafipour, Santiago Segarra, Antonio G Marques, and Gonzalo Mateos. Network topology inference from non-stationary graph signals. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5870–5874. IEEE, 2017.
- [24] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.*, 30(3):83–98, 2013. URL <http://dblp.uni-trier.de/db/journals/spm/spm30.html#ShumanNF0V13>.
- [25] Gilbert Strang. *Linear algebra and its applications*. Thomson, Brooks/Cole, Belmont, CA, 2006. ISBN 0030105676 9780030105678 0534422004 9780534422004. URL <http://www.amazon.com/Linear-Algebra-Its-Applications-Edition/dp/0030105676>.
- [26] Bruno Vallet and Bruno Lévy. Spectral Geometry Processing with Manifold Harmonics. Technical report, 2007. URL <https://hal.inria.fr/inria-00186931>.
- [27] Ulrike von Luxburg, Mikhail Belkin, and Olivier Bousquet. Consistency of spectral clustering. *Ann. Statist.*, 36(2):555–586, 04 2008. doi: 10.1214/009053607000000640. URL <https://doi.org/10.1214/009053607000000640>.

## 6 Appendix

In this section we provide the necessary definitions and mathematical concepts necessary to properly introduce the weak formulation of a differential problem, the Galerkin method and finally the Finite Element Method.

**Linear operators, functionals and bilinear forms** A *linear operator*  $L : X \rightarrow X$  from the Hilbert space  $(X, \|\cdot\|)$  to itself is a map such that  $L(\alpha x + \beta y) = \alpha Lx + \beta Ly$ . A linear operator on a Hilbert space is *continuous* (or *bounded*) if  $\exists M : \|Lx\| \leq M\|x\|$ .  $(x, \lambda)$  are called respectively *eigenvector* and *eigenvalue* of the linear operator  $L$  if the image of  $x$  through  $L$  is a rescaling of  $x$  of factor  $\lambda$  i.e.  $Lx = \lambda x$ . The operator  $L$  is called *self-adjoint* if  $\langle Lx, y \rangle = \langle x, Ly \rangle \quad \forall x, y \in X$ . Self-adjoint operators have two important properties: their eigenvalues are real, and two eigenvectors  $x, y$  associated to different eigenvalues  $\lambda, \mu$  are orthogonal. Indeed

$$\begin{aligned}\langle Lx, y \rangle &= \langle x, Ly \rangle, \\ \langle \lambda x, y \rangle &= \langle x, \mu y \rangle, \\ \lambda \langle x, y \rangle &= \mu \langle x, y \rangle,\end{aligned}$$

that implies  $\langle x, y \rangle = 0$ . If the eigenvectors of a self-adjoint operator  $L$  span the whole space  $X$ , then  $L$  is called *diagonalisable*. A linear operator from a Hilbert space  $X$  to  $\mathbb{R}$  is called a linear *functional* on  $X$ . A functional is *bounded* if  $\exists M \geq 0 : |Lx| \leq M\|x\|$ . The normed vector space made by the set of all linear bounded functionals on  $X$  endowed with the norm  $\|L\| = \sup_{v \in X} |Lv| / \|v\|$  is called the *dual* space of  $X$  and is indicated with  $X^*$ . An important property of linear functionals is that for every functional  $L \in X^*$ , there exists a unique vector  $u \in X$  such that

$$Lv = \langle u, v \rangle \quad \forall v \in X$$

A *bilinear form* on the vector space  $X$  is a map  $a(\cdot, \cdot) : X \times X \rightarrow F$  that is linear with respect to both arguments. It is said to be *strongly coercive* if  $\exists \alpha > 0 : |a(x, x)| \geq \alpha \|x\|^2$ , and *bounded* if  $\exists M > 0 : |a(x, y)| \leq M \|x\| \|y\|$ .

**The Lax-Milgram theorem** Before introducing the Galerkin method the last thing to do is to state the Lax-Milgram theorem, theorem that is the foundation of the FEM formulation.

*Theorem 6.1 (Lax-Milgram).* If  $a(\cdot, \cdot)$  is a bounded and strongly coercive bilinear form on the Hilbert space  $X$ ,  $L \in X^*$  is a linear bounded functional on  $X$ , there exist a unique solution  $f$  to the following problem:

$$\text{Find } f \in X \text{ such that } a(f, v) = Lv \text{ for all } v \in X \quad (49)$$

For such  $f$  one has  $\|f\| \leq \frac{1}{\alpha} \|L\|$  where  $\alpha > 0$  is the coercive constant  $a(v, v) \geq \alpha \|v\|^2 \forall v \in X$ .

The Finite Element Method takes a PDE problem in strong form (50), reformulates it in the equivalent *weak form* (49) through suitable definitions of  $X, a(\cdot, \cdot), L$  and finally solves it through polynomial interpolation.

### 6.0.1 Weak formulation of a PDE and Galerkin Method

Galerkin's method is a method to approximate the solution  $f$  of an infinite dimensional problem of the form (49) with the solution  $f_h$  of a finite dimensional problem. Our goal is now to explain how to write a differential problem like

$$\begin{cases} -\partial_{x_1 x_1} f(\mathbf{x}) - \partial_{x_2 x_2} f(\mathbf{x}) = u(\mathbf{x}) & \mathbf{x} \in \Omega \\ f(\mathbf{x}) = 0 & \mathbf{x} \in \partial\Omega \end{cases} \quad (50)$$

in the form 49. Let's multiply the differential equation times a sufficiently regular function  $v$  that vanishes on  $\partial\Omega$  and integrate on  $\Omega$ . We obtain

$$\begin{cases} -\int_{\Omega} \Delta f(\mathbf{x}) v(\mathbf{x}) d\mathbf{x} = \int_{\Omega} u(\mathbf{x}) v(\mathbf{x}) d\mathbf{x} & \mathbf{x} \in \Omega \\ f(\mathbf{x}) = 0 & \mathbf{x} \in \partial\Omega \end{cases} \quad (51)$$

Since the contribution of both  $f$  and  $v$  on the border  $\partial\Omega$  is zero,  
 $\int_{\partial\Omega} \nabla f \cdot \mathbf{n} v d\sigma = 0$  and integrating by parts we get

$$\int_{\Omega} \nabla f(\mathbf{x}) \cdot \nabla v(\mathbf{x}) d\mathbf{x} = \int_{\Omega} u(\mathbf{x}) \cdot v(\mathbf{x}) d\mathbf{x} \quad (52)$$

By defining  $a(f, v) := \int_{\Omega} \nabla f(\mathbf{x}) \cdot \nabla v(\mathbf{x}) d\mathbf{x}$  and  $Lv := \int_{\Omega} u(\mathbf{x}) \cdot v(\mathbf{x}) d\mathbf{x}$  problem 52 can be written in the form of equation 49. It remains to choose a Hilbert space  $(X, \langle \cdot, \cdot \rangle_X)$  such that (i)  $X$  is "big enough" to include those functions such that all the integrals and derivatives in the problem 52 is well defined. This means that  $X$  must include those functions  $f$  such that  $f, \nabla f$  are in  $L^2(\Omega)$  and has to be *complete* with respect to the norm induced by the scalar product  $\langle \cdot, \cdot \rangle_X$ . At the same time  $X$  must be (ii) "small enough" to include only those functions that vanish on the boundary of  $\Omega$ . and (iii) the hypothesis of the Lax-Milgram theorem are satisfied, i.e.  $a(\cdot, \cdot)$  is actually bounded and coercive and  $L$  is bounded and linear. It turns out that such an Hilbert space exists: it is called  $H_0^1(\Omega)$ , it contains all the functions  $v$  such that  $v \in L^2(\Omega), \nabla v \in L^2(\Omega)$  and it is endowed with the scalar product

$$\langle u, v \rangle_{H_0^1(\Omega)} = \int_{\Omega} u(\mathbf{x}) v(\mathbf{x}) d\mathbf{x} + \int_{\Omega} \nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x}) d\mathbf{x}$$

$H_0^1$  contains only those functions that vanish on  $\partial\Omega$  i.e.

$f \in H_0^1(\Omega) \cap C(\Omega) \implies f(\mathbf{x})|_{\partial\Omega} = 0$ . This means that thanks to Lax-Milgram theorem the problem

$$\begin{aligned} &\text{Given } u \in L^2(\Omega) \text{ find } f \in H_0^1(\Omega) \text{ such that} \\ &\int_{\Omega} \nabla f(\mathbf{x}) \cdot \nabla v(\mathbf{x}) d\mathbf{x} = \int_{\Omega} u(\mathbf{x}) \cdot v(\mathbf{x}) d\mathbf{x} \quad \forall v \in H_0^1(\Omega) \end{aligned} \quad (53)$$

*has one and only one solution in  $H_0^1(\Omega)$ .* However, to have this result (existence and uniqueness of the solution) we had to pay the price of looking for the solution  $f$  in  $H_0^1(\Omega)$ , a much bigger space than  $C^2(\Omega)$  that we had in the original, strong form of the problem. This means that our solution  $f \in H_0^1(\Omega)$  to the problem (53) could not be a solution to problem (50), because it could be not regular enough and the second derivative  $\Delta f$  could not exist! Fortunately there are regularity results - that we omit here - that assure that if the forcing

term  $u$  is regular enough, then also the solution  $f$  will be regular and thus the two formulations - strong and weak - of the problem are actually equivalent, and thus solving 53 eventually leads to solving 50.

Now that we know that a solution exists, we need to compute it! Computing it analytically is often impossible; Galerkin's method in a mathematical tool provide us a way to compute an approximation of the solution  $f$ . Take the weak problem 49, but restrict the ambient space to be a finite dimensional subspace of  $X$ , say  $V_h = \text{span}\{\phi_0, \dots, \phi_{n-1}\}$ . We write thus the Galerkin problem

$$\text{Find } f_h \in V_h \text{ such that } a(f_h, v_h) = \langle u_h, v_h \rangle \text{ for all } u_h \in V_h \quad (54)$$

The key property of the Galerkin method is that the error  $f - f_h$  is orthogonal to  $V_h$ ; thus by choosing a sequence of finite dimensional spaces that fill the original space  $X$ , we can get as close as we want to the continuous solution  $f$ . To solve equation 54 we write  $f_h, u_h, v$  as linear combinations of the basis  $\{\phi_0, \dots, \phi_{n-1}\}$

$$\begin{cases} f_h = f_0\phi_0 + f_1\phi_1 + \dots f_{n-1}\phi_{n-1} \\ u_h = u_0\phi_0 + u_1\phi_1 + \dots u_{n-1}\phi_{n-1} \\ v_h = v_0\phi_0 + v_1\phi_1 + \dots v_{n-1}\phi_{n-1} \end{cases} \quad (55)$$

thus obtaining by linearity of the bilinear form and of the scalar product a linear system of equations in the  $n$  coordinates of  $f_h$

$$\text{Find } \mathbf{f} \in \mathbb{R}^n \text{ such that } \sum_{j=0}^{n-1} a(\phi_i, \phi_j) f_j = \sum_{j=0}^{n-1} \langle \phi_i, \phi_j \rangle u_j \text{ for all } i = 0, \dots, n-1 \quad (56)$$

Defining the *stiffness matrix*  $(A)_{ij} = a(\phi_i, \phi_j)$  and the *mass matrix*  $(B)_{ij} = \langle \phi_i, \phi_j \rangle$  we can rewrite problem 56 in the following algebraic form

$$\text{Find } \mathbf{f} \in \mathbb{R}^n \text{ such that } \mathbf{A}\mathbf{f} = \mathbf{B}\mathbf{u} \quad (57)$$

where  $\mathbf{f}, \mathbf{u}$  are the vectors of coordinates of  $f_h, u_h$  with respect to the basis  $(\phi_i)$ .

*Remark.* If the basis functions  $\phi_i$  are orthonormal, then the mass matrix  $B$  is the identity matrix. Furthermore, the scalar product in the space  $V_h$  of two functions  $u_h, v_h$  is equal to the dot product defined by the mass matrix  $B$  in  $\mathbb{R}^n$  of the coordinate vectors

$$\langle u_h, v_h \rangle = \mathbf{u}^\top B \mathbf{v} \quad (58)$$

The Galerkin method 54 is well posed by a straight-forward application of the Lax-Milgram theorem, and thus also the system 57 admits one and only one solution.

### 6.0.2 The Finite Element Method

The Finite Element Method is a technique that let us construct a particular subspace  $V_h$  in (54) through polynomial interpolation. Let's refer again to the problem 50 defined on the domain  $\Omega \subset \mathbb{R}^2$  in figure 37.

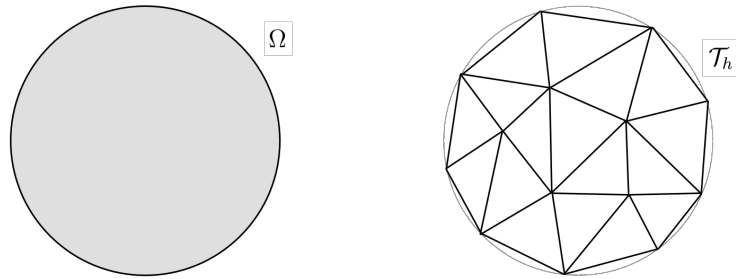


Figure 37: The domain  $\Omega$  and its approximation  $\mathcal{T}_h$

**Approximation of the domain  $\Omega$**  First we need to construct a discretization of the continuous domain  $\Omega$ . In this example we take the triangulation in figure 37  $\mathcal{T}_h = \{\tau_k : 1 \leq k \leq q\}$ , where  $h \in \mathbb{R}$  is a parameter such that every edge of the triangles  $\tau_k \in \mathcal{T}_h$  is smaller than  $h$ .  $\mathcal{T}_h$  is such that

- The elements  $\tau \in \mathcal{T}_h$  are closed subsets of  $\Omega$  with pairwise disjoint interior and  $\Omega_h = \bigcup_{\tau \in \mathcal{T}_h} \tau$
- The triangulation  $\mathcal{T}_h$  has no hanging vertices.

It's clear that by discretizing the continuous domain  $\Omega$  we introduce a first source of errors in the method; however for simplicity we won't take this into

account in the next discussion, and we will identify the domain  $\Omega$  with the domain  $\Omega_h$  covered by  $\mathcal{T}_h$ . The quality of the *mesh*  $\mathcal{T}_h$  is important for a good solution; a good quality mesh should avoid triangles with extreme angles and every triangle of  $\mathcal{T}_h$  should look like as much as possible to an equilateral triangle. In figure 38 we see an example of a bad quality mesh: its triangles look very stretched; there are vertices that are shared by many triangles and vertices that are shared by very few. In figure 37 we see a better mesh: there are no stretched triangles, and every vertex is shared by an almost constant number of triangles.

**Choosing  $V_h$  and the basis functions  $\phi_i$**  Beyond  $X_h^1$ , explained in section ??, other choices for  $V_h$  are possible. A common choice is the space  $X_h^2$  that is the space of all the piecewise second-order polynomials. In this case, since every second order polynomial in  $\mathbb{R}^2$  has 6 degrees of freedom per each triangle  $\tau_k$ , meaning that we need to know its values in at least 6 different points on the triangle  $\tau_k$  to uniquely identify it, the dimension of the space will grow (figure 39). A bigger space means that the approximation  $f_h$  will be better, but we'll need more basis functions to define and thus it will result in a bigger linear system  $Af = Bu$  and higher computational costs.

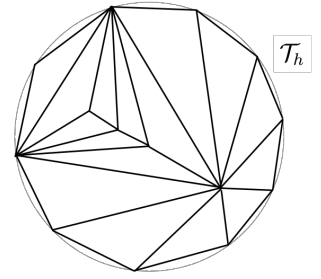


Figure 38: A bad quality mesh

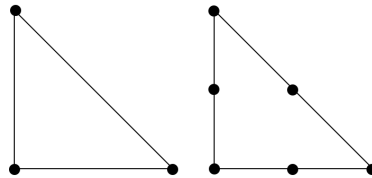


Figure 39: The degrees of freedom (DOF) for  $X_h^1$  and  $X_h^2$  on a reference triangle

**Assembling the stiffness and mass matrices** Once defined the basis functions  $\phi_i$ , the FEM method constructs the stiffness matrix and the mass

matrix  $(A)_{ij} = a(\phi_i, \phi_j)$ ,  $(B)_{ij} = \langle \phi_i, \phi_j \rangle$  and solves the linear system 57 for the coefficients  $f_i$  of the FEM solution  $f_h$ . An important fact to notice is both matrices  $A$  and  $B$  are *sparse* and share the same sparsity pattern. Due to the form of the basis function  $\phi_i$ , the element  $(i, j)$  of these matrices is different from zero only if the supports of the corresponding basis functions  $(\phi_i, \phi_j)$  overlap, meaning that the vertices  $(x_i, x_j)$  are connected by an edge of the mesh  $\mathcal{T}_h$ . In other words, the number of non-null entries of the  $i - th$  row of  $A$  and  $B$  is equal to number of triangles of the mesh  $\mathcal{T}_h$  that share the  $i$ th vertex i.e., the degree of the  $i - th$  vertex.

**About the boundary conditions** Note that the Dirichlet boundary conditions in the strong formulation of the differential problem 50 got transformed in the weak formulation 53 as a condition on the ambient space  $H_0^1(\Omega)$ . Other kind of boundary conditions (Neumann, Robin for example) would not translate into the condition on the ambient space that functions must vanish on the border, but instead they would impose a different formulation of the bilinear form  $a(\cdot, \cdot)$  or of the functional  $L$ . For this reason, Dirichlet boundary conditions are called *essential* since translated into a condition on the ambient space and thus automatically satisfied from the FEM formulation; Neumann boundary conditions are called *natural* since they transform into a different weak formulation through a modification of the bilinear form  $a(\cdot, \cdot)$  and/or the functional  $L$ .