Graph Laplacians and discrete Laplace-Beltrami operators on the Sphere for Rotation Equivariant Neural Networks

Martino Milani

Supervised by Michaël Defferrard, Prof. Pierre Vandergheynst, prof. Piercesare Secchi

Cosupervised by Ph.D. Nathanaël Perraudin

Abstract

We want to design computationally efficient algorithms to filter spherical images of the sky, in order to construct a so-called Graph Spherical Convolutional Neural Network. It is then important to design filtering algorithms that are **computationally efficient** and capable of **exploiting the rotational symmetry of the problem**. A way to do so is to construct a sparse **graph** with the vertices corresponding to the pixels of the image and then use a polynomial of the **graph Laplacian** matrix to perform a computationally efficient filtering of the sampled signal. In order to study how well this algorithm respects the symmetry of the problem - i.e., it is **equivariant** to the rotation group SO(3) - it is important to guarantee that the eigenvectors of the graph Laplacian and of the Laplace-Beltrami operator are somewhat *close*.

We show a way to build a graph such that the corresponding graph Laplacian matrix L shows good spectral properties **improving the current state-of-the-art rotation equivariant graphs**.

We then investigate different methods of building the graph Laplacian, 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 we studied a way to **make graph-like filtering inspired from FEM.**

Figure 1: the Nine Year Microwave Sky, an example of the images we want to filter. Source: NASA

1. Convolutions on the Sphere and rotation equivariant graph filtering

Take a sampling $V = \{v_i \in \mathbb{S}^2, i = 0, ..., n\}$ of the sphere, a weighted undirected graph $G(V, E, \mathbf{W})$, a signal $f : \mathbb{S}^2 \to \mathbb{R}$ and the sampling operator $T_V : L^2(\mathbb{S}^2) \supset F \to \mathbb{R}^n,\ T_V(f) = \mathbf{f}$ such that $\mathbf{f} : \quad f_i = f(v_i)$. It's a well known fact that convolutions on the sphere can be performed in the spectral domain as product of the Fourier transformed signals. On a graph we can do the same thing: we can define the graph Laplacian

$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

through the diagonal matrix $\mathbf{D}_{ii} = \sum_j w_{ij}$ and define the **graph Fourier transform** as the projection of a signal $\mathbf{f}$ on its eigenvectors:

$$\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$$
$$\mathcal{F}_G(\mathbf{f}) := \mathbf{V}^T\mathbf{f} = \hat{\mathbf{f}}$$

In this way we can define the **graph convolution** of a signal $\mathbf{f}$ with a kernel $\mathbf{K}$

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

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

The goal of this work is to understand how to make graph convolutions *behave* like spherical convolutions, i.e., **they should commute with any rotation** of the signal. This means that it should be true that

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

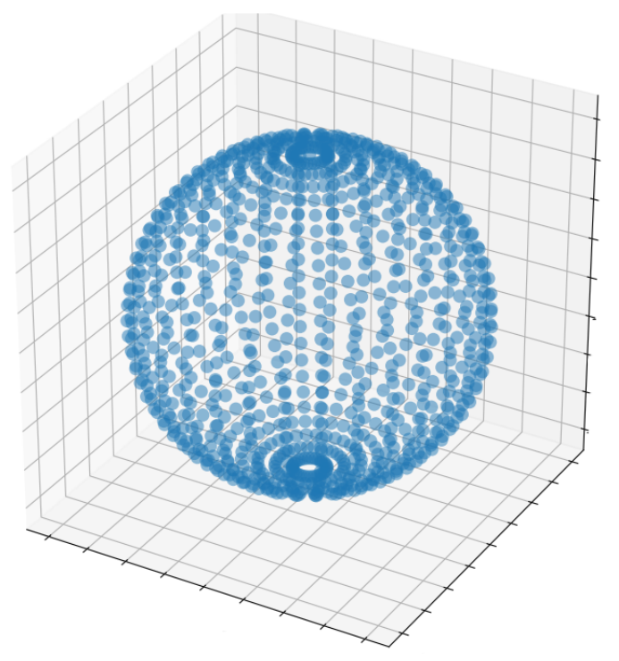


Figure 2: a sampling scheme of the sphere very used in applications to sample continuous spherical signals. These pixels in figure will be connected by suitable weighted edges forming the graph $G(V, E, \mathbf{W})$ that will be used to perform rotation equivariant graph convolutions.

2. Graph Laplacian on the Sphere

By defining the following edge weights

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

Belkin et al. [1] proved that the spectral convergence of the corresponding graph Laplacian $\mathbf{L}_n^t$ to the continuous Laplace-Beltrami operator on the sphere, making it a good candidate for rotation equivariant graphs:

Theorem 2.1. (Belkin et al., [3]) Let $\lambda_{n,i}^t$ be the ith 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 λ_i and v_i be the corresponding eigenvalue and eigenfunction of Δ respectively. Then there exists a sequence $t_n \to 0$, such that

$$\lim_{n\to\infty} \lambda_{n,i}^{t_n} = \lambda_i$$
$$\lim_{n\to\infty} \left\|\mathbf{v}_{n,i}^{t_n} - v_i(\mathbf{x})\right\|_2 = 0$$

where the limits are in probability.

However, it has three drawbacks: first, it works only with **equiarea** sampling schemes. Second, in order to make the graph convolution efficient, we need a very **sparse** graph, not a full one. We will need to find a way to make it sparse trying to maintain its spectrum as close as possible to Δ. Finally, we need to carefully choose the **kernel width** t.

3. How to build an efficient rotation equivariant graph for an equiarea sampling scheme

We measure the alignment of the eigenspaces of the graph Laplacian with the eigenspaces of the continuous Laplace-Beltrami operator as we change the **kernel width** t. The next plots shows what happens as we increase the kernel width.

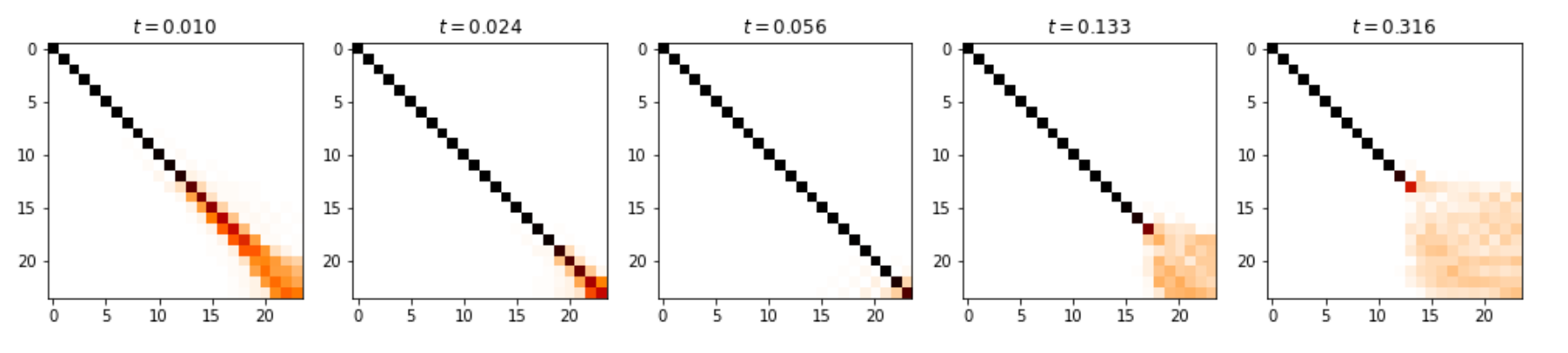


Figure 3: for each value of the kernel width we plot in the entry (i, j) of the matrix the alignment of the ith eigenspace of the graph Laplacian with the jth eigenspace of Δ.

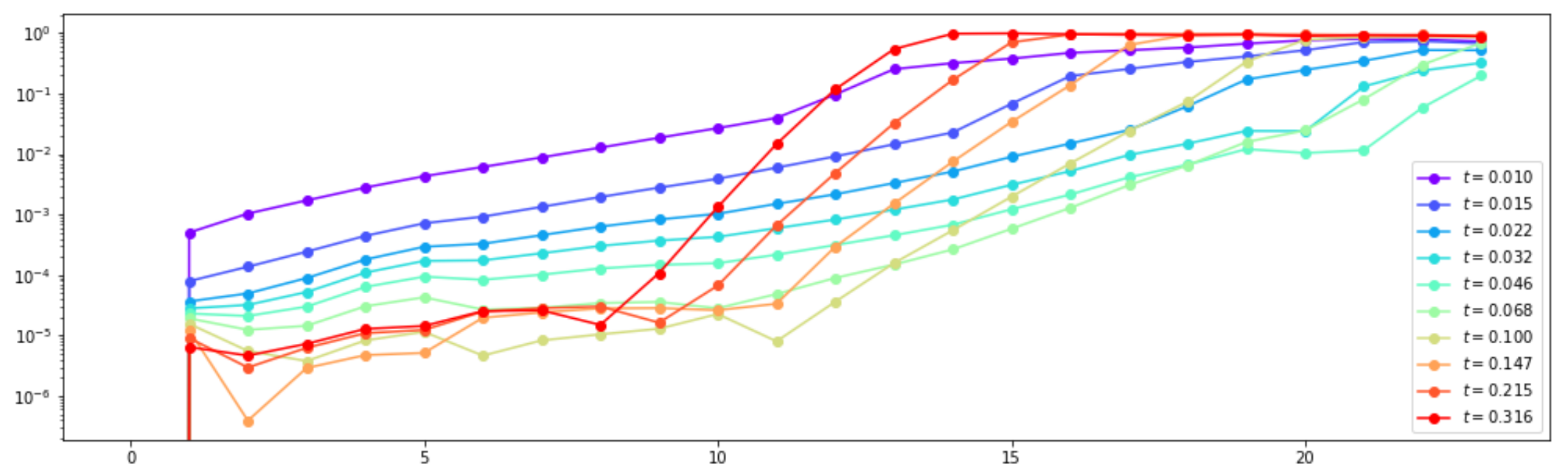


Figure 4: for each eigenspace of Δ we calculate the error in the alignment of the eigenspace of the graph Laplacian with the corresponding eigenspace of Δ.

In both figures we see the existence of an optimum: starting from a small value of $t = 0.01$, as we increase its value, the alignment matrix (figure 3) starts to look more like the identity matrix - meaning that the eigenspaces are perfectly aligned - and the error (figure 4) decreases for all frequencies. When we reach the optimum value around $t = 0.05$, we see that the graph Laplacian is still able to correctly approximate the low frequencies, but start to be unable to *see* the higher ones.

To make the graph Laplacian sparse, we simply **threshold** its weights.

4. Results: mean equivariance error

For a given sampling, a given function f, a given rotation g, and a given filter k we compute the normalized equivariance error

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

And define the **mean normalized equivariance error** $\overline{E}_G$

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

We comparing the state-of-the-art of rotation equivariant graphs DeepSphere [2] with the construction proposed in this work, obtaining the following encouraging results.

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%

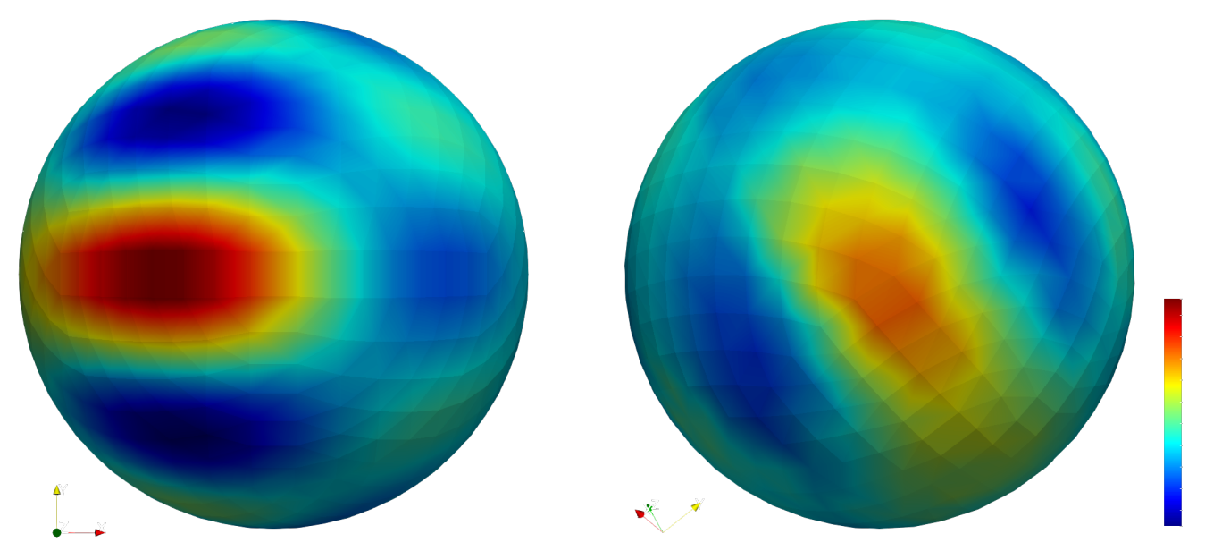


Figure 5: on the left, the signal f. On the right, the signal filtered (diffused) signal and rotated. The mean equivariance error measures the difference between *first* rotating the signal and *then* filtering it, and *first* filtering it and *then* rotating it. In the continuous setting the two operations commute and this error is zero.

5. Results: Experimental validation

Gusset et al. [3] compared the performances of DeepSphere on the SHREC17 perturbed dataset [4]. They modified the available graph of DeepSphere and proposed four different graph architectures to compete in the SHREC17 competition. One of these four architectures - called in the table *DeepSphere Optimal* - was obtained implementing the procedure described in this work. They compared the performances of these four DeepSphere architectures together with the traditional rotation equivariant CNNs of Cohen et al. [4], Esteves et al. [5]. The models were tested in a retrieval task, and evaluated with two different metrics: accuracy and F1-score while also comparing the speed of inference and of training of each model.

	performance		size	speed	
Method	Accuracy	F1-score	params	inference	training
Cohen *s2cnn*	73.10	72.86	1.4M	38ms	65h
Cohen *s2cnn_simple*	78.59	78.85	400k	12ms	32h
Esteves *sphericalcnn*	79.18	79.36	500k	9.8ms	2h52
Deepsphere *Cohen-like*	77.86	77.90	170k	17.5ms	9h43
Deepsphere *local filter*	76.83	76.66	60k	2.8ms	1h37
Deepsphere *equiangular*	73.36	73.67	190k	**0.98ms**	**43m**
Deepsphere *Optimal*	**80.42**	**80.65**	190k	1.0ms	48m

DeepSphere Optimal has the highest score between all the rotation equivariant models. Its performances in terms of speed of influence and training are comparable to the best one. It can be appreciated how all the four Graph CNNs are orders of magnitude faster to train than the traditional Spherical CNNs of Cohen et al. and of Esteves et al.

Bibliography

[1] M. Belkin, P. Nyiogi, *Convergence of Laplacian Eigenmaps*, 2006.

[2] N. Perraudin, M. Defferrard, T. Kacprzak, R. Sgier, *DeepSphere: Efficient spherical Convolutional Neural Network with HEALPix sampling for cosmological applications,* 2018

[3] F. Gusset, *Spherical Convolutional Neural Networks: empirical analysis of SCNNs,* 2019

[4] T.S. Cohen, M. Geiger, J. Köhler, M. Welling, *Spherical CNNs,* 2018

[5] C. Esteves C. Allen-Blanchette, A. Makadia, K. Daniilidis, *3D object classification and retrieval with Spherical CNNs,* 2017

6. Graph filtering inspired by the FEM for non equiarea sampling schemes

To filter a signal sampled on the sphere with the linear Finite Element Method one has to project the weak eigenvalue problem for the Laplace-Beltrami operator

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

on the subspace spanned by piecewise linear basis functions in figure 6. This leads to a filtering of the signal given by

$$B^{-1}V^{-T}KV^TB\mathbf{f}$$

where B is the *mass matrix,* V is matrix of eigenvectors of the *stiffness* matrix A and K is the diagonal matrix representing the filter. Taking inspiration from this we can define the following sparse **symmetric lumped FEM Laplacian**

$$\mathbf{L} = D^{-1/2}AD^{-1/2}$$
$$D = \mathrm{diag}\{d_i\}, \quad d_i = \sum_j (B)_{ij}$$

that is sparse and symmetric, thus a possible good candidate to be used in rotation equivariant graph convolutions.

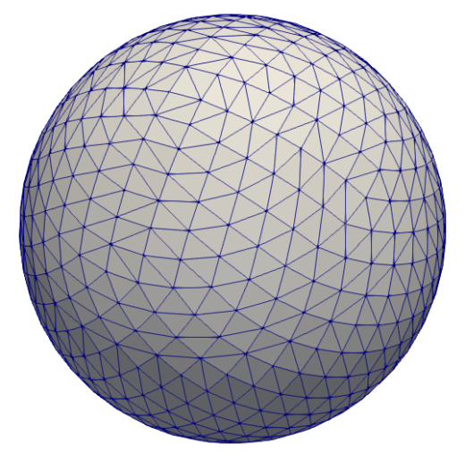


Figure 6: a triangulation of the sphere used for the FEM

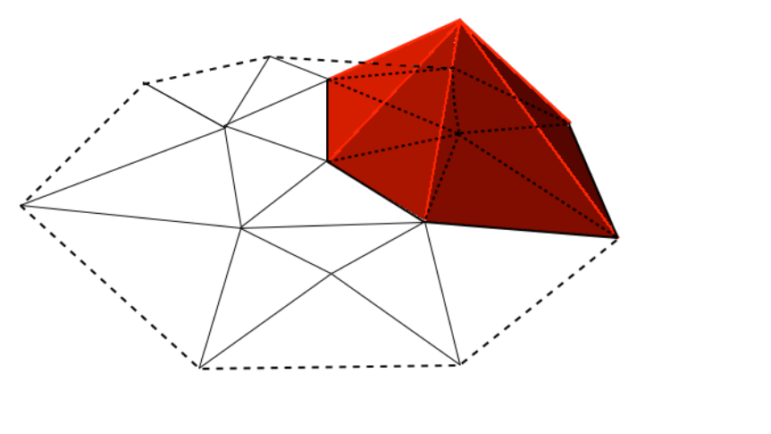
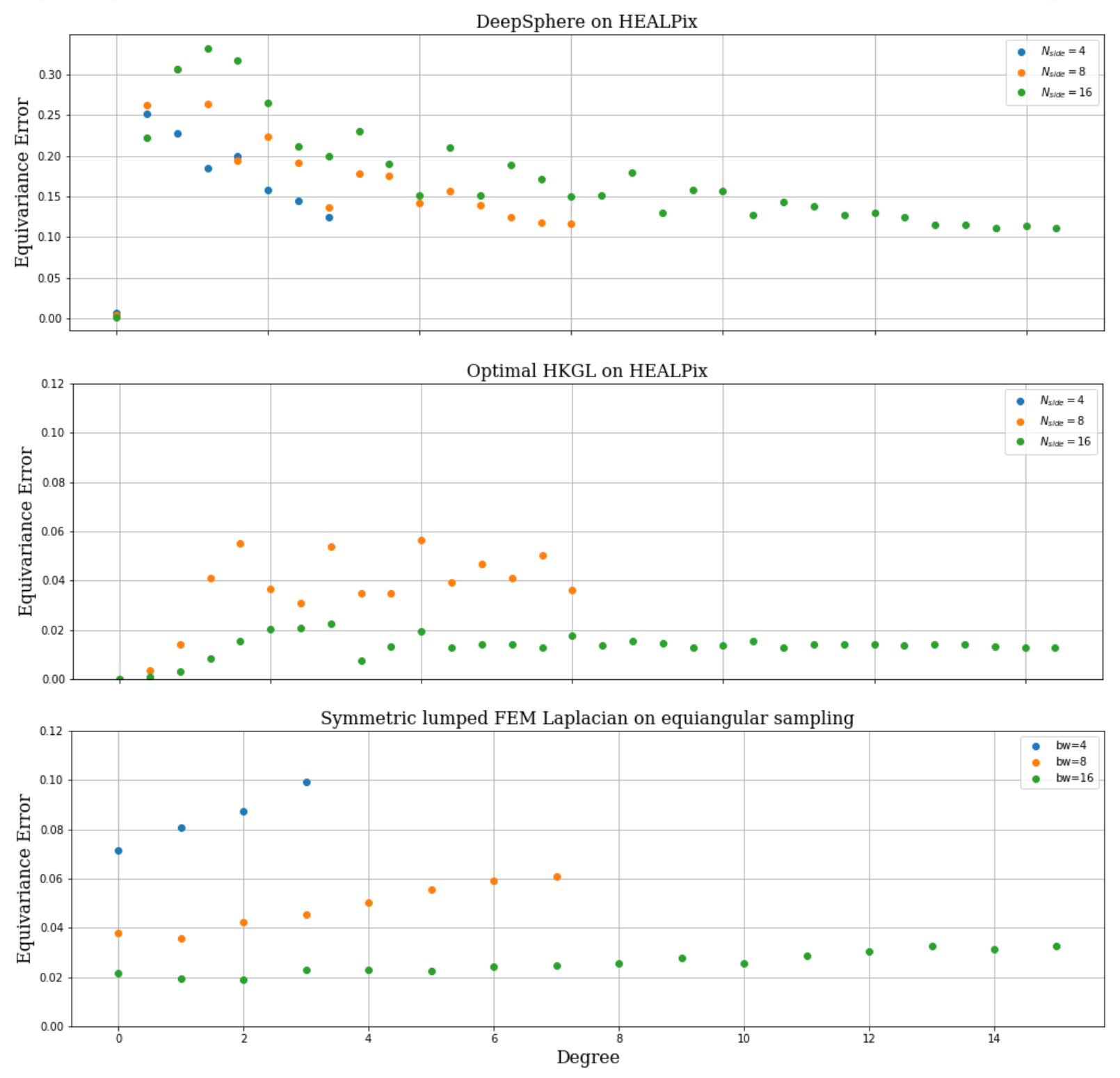


Figure 7: one of the piecewise linear basis functions of the space on which the FEM projects the weak eigenvalue problem

5. Conclusions

We report the plots of the **mean equivariance error** by spherical harmonic degree of the three constructions analyzed in this work: **DeepSphere**, **DeepSphere Optimal** and the **symmetric lumped FEM Laplacian** where the improvements with respect to DeepSphere, the starting point of this work, can be appreciated. Notice the change in the scale of the y axis for the graph of DeepSphere, that show errors up to 30%. N_{side} and bw are parameters proportional to the number of vertices of the graph.



Contacts

EPFL, martino.milani@alumni.epfl.ch
EPFL, michael.defferrard@epfl.ch
EPFL, pierre.vandergheynst@epfl.ch
ETHZ, nathanael.perraudin@sdsc.ethz.ch
Politecnico di Milano, piercesare.secchi@polimi.it