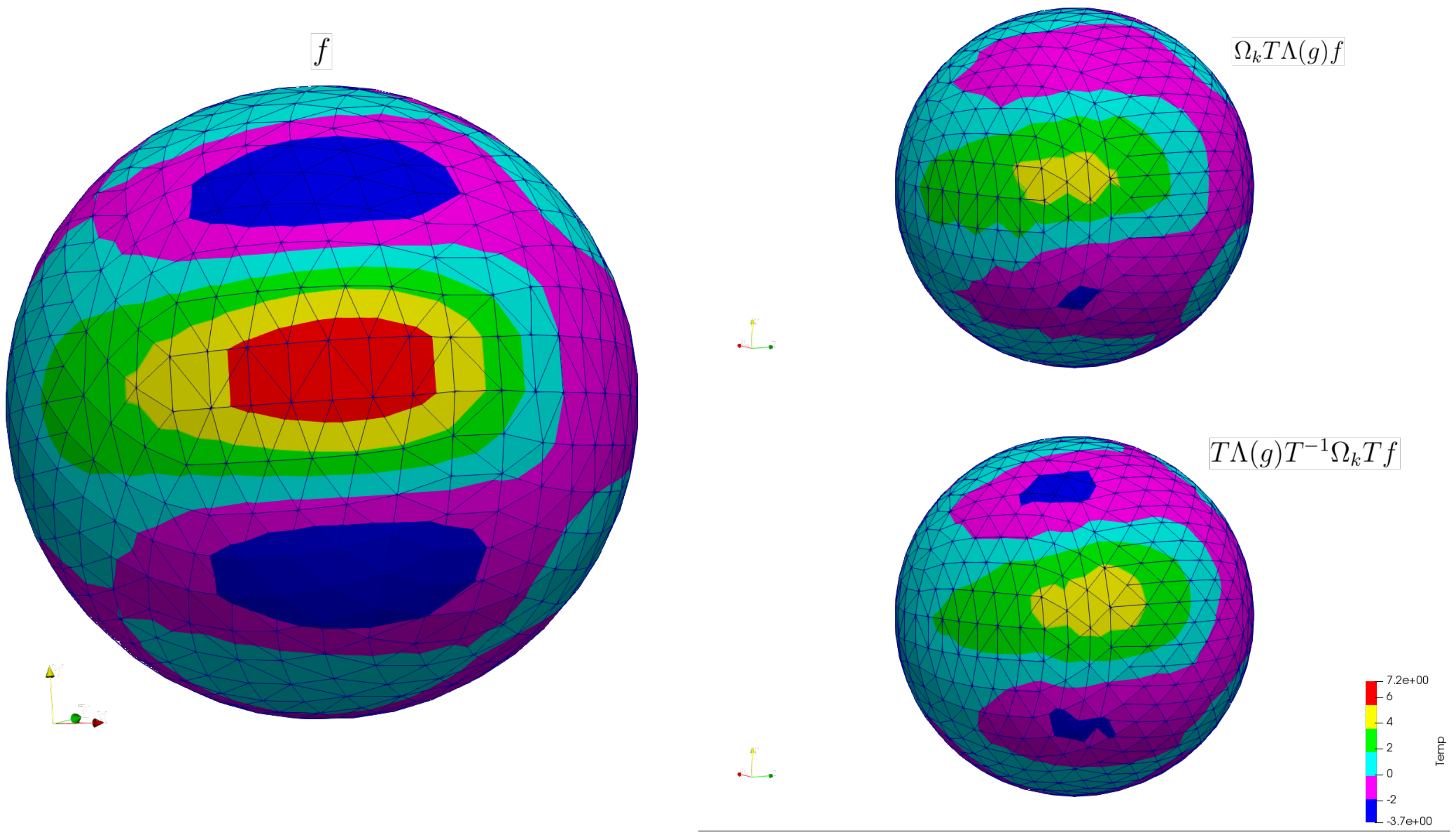
GRAPH LAPLACIANS FOR ROTATION EQUIVARIANT NEURAL NETWORKS

Master thesis in Computational Science and Engineering

School of Basic Sciences
Department of Mathematics
École Polytechnique Fédérale de Lausanne

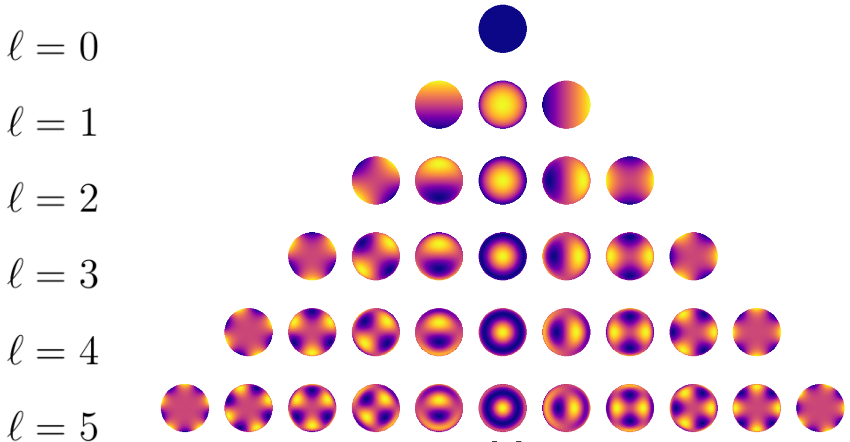
Supervised by: Michaël Defferrard (EPFL)
Nathanaël Perraudin (SDSC – ETH)
Piercesare Secchi (Politecnico di Milano)
Pierre Vandergheynst (EPFL)

Rotation Equivariant Filtering



Fourier analysis on the sphere and the spherical harmonics

$$\Delta Y_\ell^m = -\ell(\ell+1)Y_\ell^m$$



[1]

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

[1] *Starry: analytic occultation light curves*, Luger R. et al., https://rodluger.github.io/starry/v0.3.0/tutorials/basics1.html

Convolution on the sphere

Definition: $$k * f(\omega) = \int_{g \in SO(3)} k(g\eta) f\left(g^{-1}\omega\right) dg\,.$$

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 [2]

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

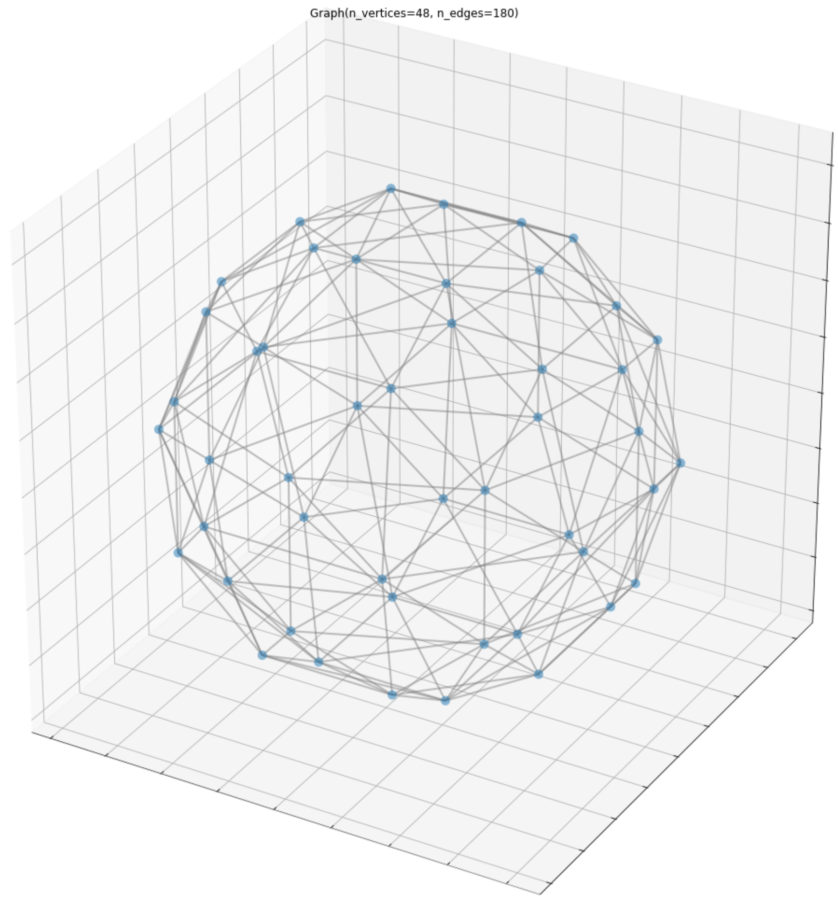
- Convolutions can be done in the spectral domain as usual:

$$f * h = \mathcal{F}^{-1}(\hat{h}\mathcal{F}(f))$$

- Complexity of FFT algorithms is $\mathcal{O}(n^{3/2})$.
- Convolutions are **rotation equivariant** operations

[2] *Computing Fourier Transforms and Convolutions on the 2-Sphere*, Driscoll J.R. and Healy D.M., 1994.

Our tool to perform fast spherical convolutions: graphs.



Adjacency matrix
$\mathbf{W}$

Graph convolutions

- Given a weighted adjacency matrix $\mathbf{W}$, define the graph Laplacian to be

$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$
$$\mathbf{D}_{ii} = \sum_j w_{ij}$$

- The graph Laplacian is symmetric:

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

- Convolving on the graph a signal $\mathbf{f}$ with a kernel k is defined as:

$$\mathbf{\Omega}_G^k f = \mathbf{V}k(\mathbf{\Lambda})\mathbf{V}^\top\mathbf{f}$$

Notice the similarity with $f * k = \mathcal{F}(\hat{k}\mathcal{F}(f))$

How to build a rotation equivariant graph

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

KEY CONCEPT: **if** $\mathbf{v}^\intercal\mathbf{f} \approx \langle f, Y_\ell^m \rangle_{L^2} = \hat{f}(\ell, m)$ **then** the graph Fourier transform will be rotation equivariant.

Heat Kernel Graph (HKG):

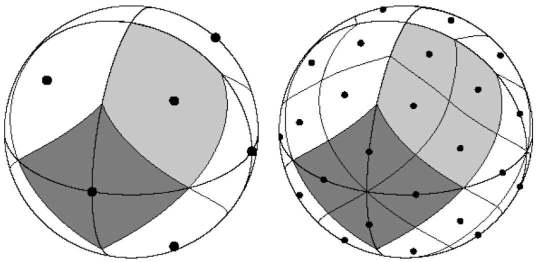
$$w_{ij} = \exp\left(-\frac{||x_i - x_j||^2}{4t}\right)$$

Belkin et al. proved that with a random sampling scheme, in probability [3]

$$\mathbf{Eig}\mathbf{L}_n \xrightarrow{n\to\infty} \mathbf{Eig}\Delta$$

[3] Convergence of *Laplacian Eigenmaps*, Belkin M. and Nyiogi P,, in Advances in Neural Information Processing Systems 19, 2007.

HEALPix: equiarea sampling scheme



- The HKG Laplacian eigenvectors well approximate the spherical harmonics:

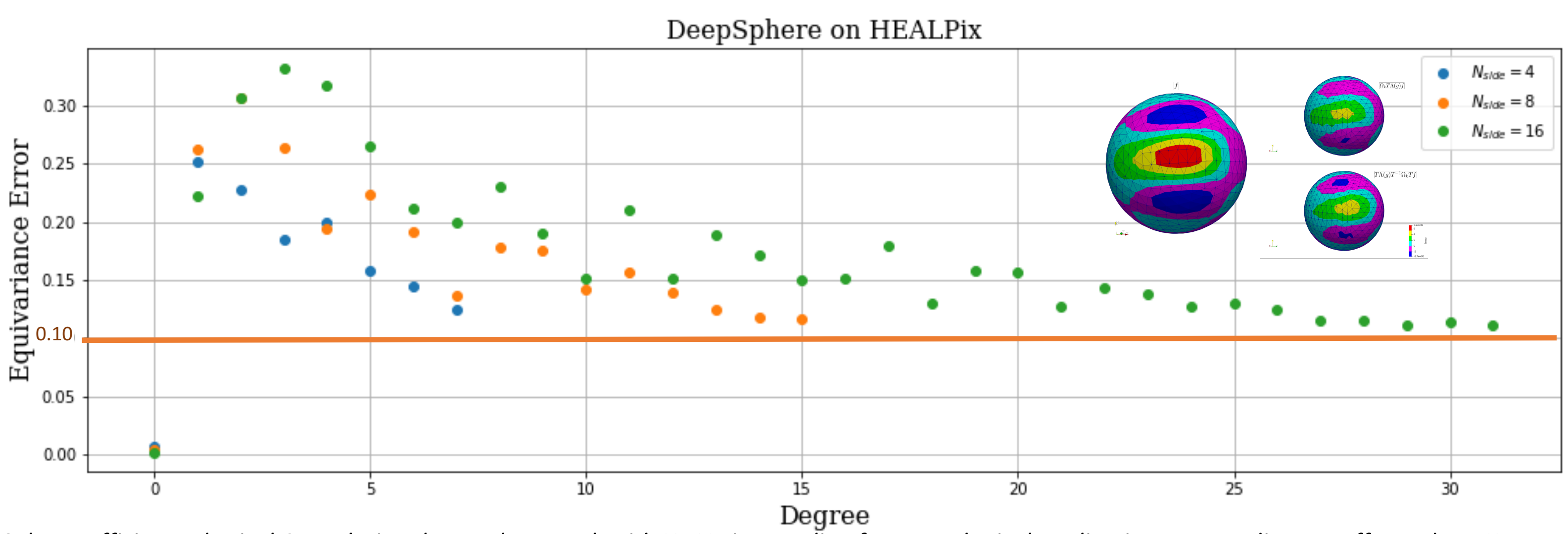
$$\mathbf{v} \approx Y_\ell^m(\mathbf{x})$$

- The dot product of the HKG Laplacian eigenvectors and any sampled signal well approximates the corresponding Fourier coefficient:

$$\mathbf{v}^\intercal \mathbf{f} \approx \langle f, Y_\ell^m \rangle_{L^2} = \hat{f}(\ell, m)$$

DeepSphere 1.0

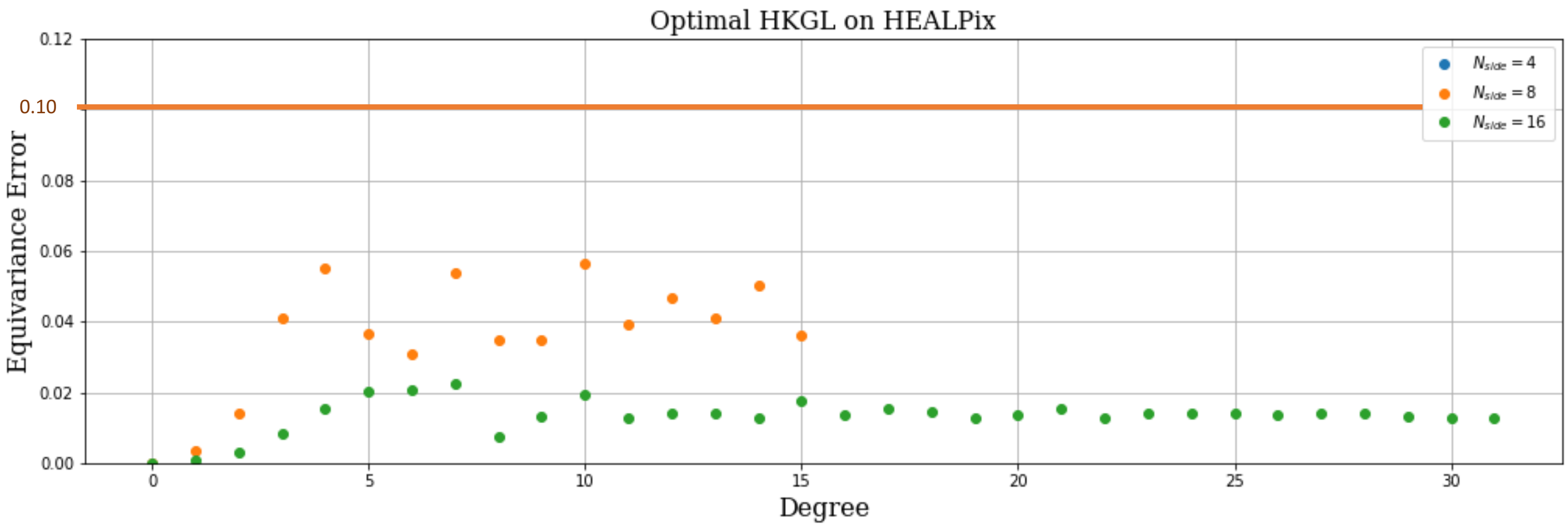
- DeepSphere[4] is a Spherical Graph Convolutional Neural Network
- Each layer implements a **polynomial** filter of a **sparse** HKG Laplacian of a spherical signal sampled with HEALPix.
- Filtering is $\mathcal{O}(n)$



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

DeepSphere 2.0

- DeepSphere 2.0 is an improved version of DeepSphere
- Each layer implements a **polynomial** filter of a sparse (but **not too sparse**) HKG Laplacian of a spherical signal sampled with HEALPix.
- Filtering is $\mathcal{O}(n^{5/4})$



Non-equiarea sampling schemes

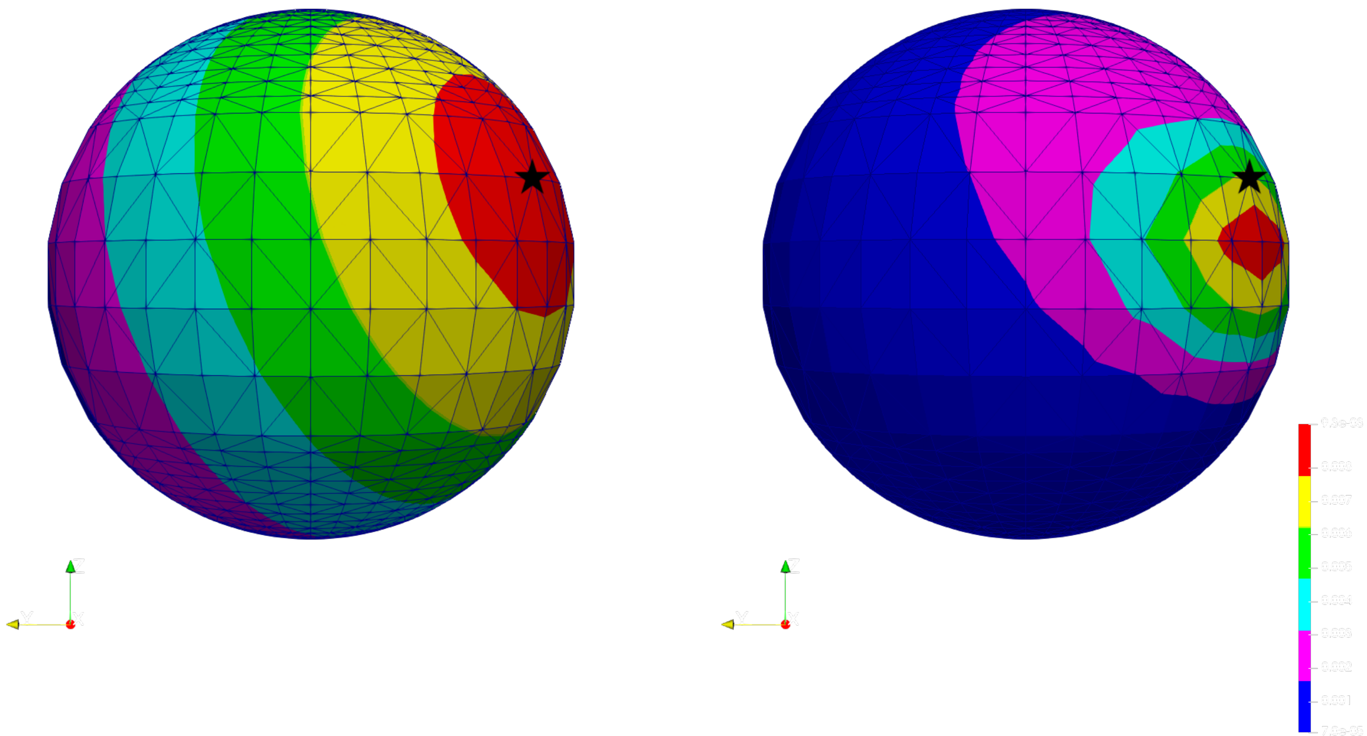


Figure: left, FEM diffusion of a point source signal. Right, diffusion obtained with the HKG Laplacian

Towards the Finite Element Method (FEM)

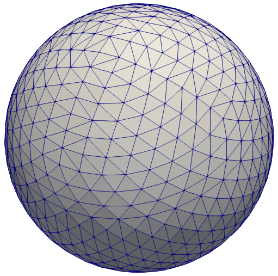
$$\Delta f = -\lambda f$$

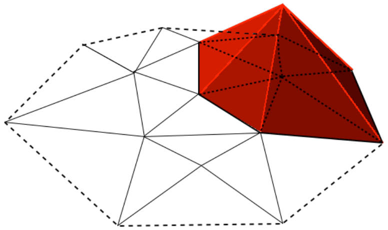
$$\boxed{\langle \nabla f, \nabla v \rangle_{L^2(\mathbb{S}^2)} = \lambda \langle f, v \rangle_{L^2(\mathbb{S}^2)} \quad \forall v \in L^2(\mathbb{S}^2)}$$

The **Galerkin problem** is a discretized version of the weak eigenvalue problem, where the ambient space is finite dimensional:

$$\begin{cases} \langle \nabla f_h, \nabla v_h \rangle_{L^2(\mathbb{S}^2)} = \lambda \langle f_h, v_h \rangle_{L^2(\mathbb{S}^2)} \quad \forall v_h \in V_h \\ V_h \subset L^2(\mathbb{S}^2), \quad V_h = \text{span}\{\phi_0, ..., \phi_{n-1}\} \end{cases}$$

The Finite Element Method (FEM)



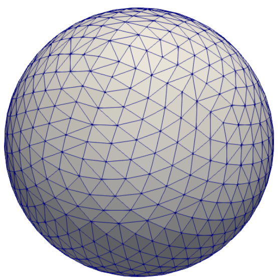


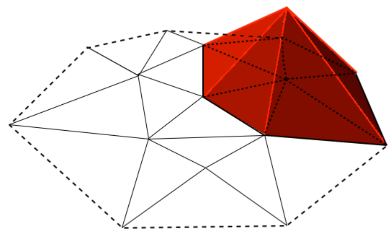
$$f_h(x) = \sum_{i=0}^{n-1} f_i \phi_i(x)$$

The Finite Element Method (FEM) is a method to solve the Galerkin problem where the space V_h is the space of continuous piecewise linear functions defined on a triangulation of the sphere.

$$\begin{cases} \langle \nabla f_h, \nabla v_h \rangle_{L^2(\mathbb{S}^2)} = \lambda \langle f_h, v_h \rangle_{L^2(\mathbb{S}^2)} \quad \forall v_h \in V_h \\ V_h \subset L^2(\mathbb{S}^2), \quad V_h = \text{span}\{\phi_0, ..., \phi_{n-1}\} \end{cases}$$

The Finite Element Method (FEM)

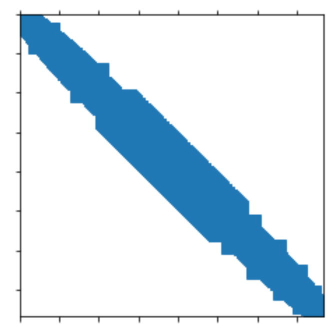




$$f_h(x) = \sum_{i=0}^{n-1} f_i \phi_i(x)$$

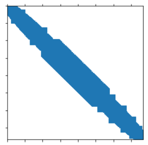
Writing the Galerkin problem n times, each time with $v_h = \phi_i$ we obtain the following **algebraic** generalized eigenvalue problem:

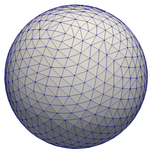
$$\begin{cases} \text{Find } (\mathbf{f}, \lambda) \text{ such that } \mathbf{Af} = \lambda \mathbf{Bf} \\ (\mathbf{A})_{ij} = \int_{\mathbb{S}^2} \nabla \phi_i(\mathbf{x}) \cdot \nabla \phi_j(\mathbf{x}) d\mathbf{x} \\ (\mathbf{B})_{ij} = \int_{\mathbb{S}^2} \phi_i(\mathbf{x}) \phi_j(\mathbf{x}) d\mathbf{x}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

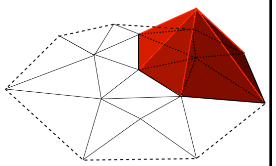


FEM convergence

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







$$\|f - f_h\|_{H^1(\Omega)} \leq C h^r |f|_{H^2}$$

$$|\lambda - \Lambda| \leq C(\lambda) h^2$$

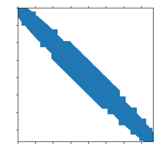
[5]

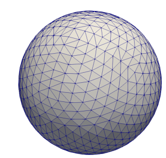
Remark: it's the ***only*** case (together with the random HKG Laplacian) where we have a convergence theorem.

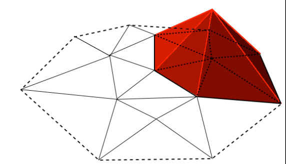
[5] *A priori estimates for the FEM approximations to eigenvalues and eigenfunctions of the Laplace-Beltrami operator*, Bonito et al., 2017.

FEM Fourier transform

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







- Equivalent to finding the decomposition $\mathbf{B}^{-1}\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$
- The eigenvectors are such that:

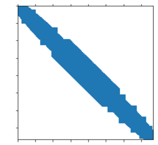
$$\mathbf{V}\mathbf{B}\mathbf{V}^{\intercal} = \mathbf{I}$$

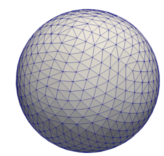
- FEM Fourier transform:

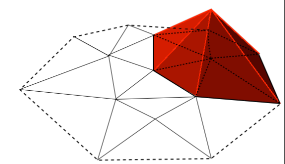
$$\hat{f} \approx \int f_h Y_h = \mathbf{v}^{\intercal}\mathbf{B}\mathbf{f}$$

FEM filtering

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







Filtering a signal $\mathbf{f}$ with a kernel k is defined as the following matrix multiplication, where $\mathbf{V}^\intercal\mathbf{B}$ is the FEM Fourier matrix.

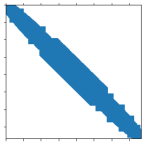
$$\boldsymbol{\Omega}^k_{\text{FEM}}\mathbf{f}, \quad \boldsymbol{\Omega}^k_{\text{FEM}} = (\mathbf{V}^\intercal\mathbf{B})^{-\mathbf{1}} k(\boldsymbol{\Lambda})\mathbf{V}^\intercal\mathbf{B}$$

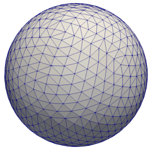
$$\mathbf{V}^\intercal\mathbf{B}\mathbf{V} = \mathbf{I} \implies \begin{cases} \mathbf{V}^\intercal\mathbf{B} = \mathbf{V}^{-\mathbf{1}} \\ (\mathbf{V}^\intercal\mathbf{B})^{-\mathbf{1}} = \mathbf{V} \end{cases}$$

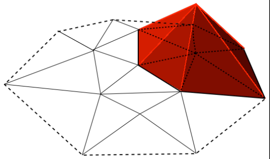
$$\boldsymbol{\Omega}^k_{\text{FEM}} = \mathbf{V}k(\boldsymbol{\Lambda})\mathbf{V}^{-1}$$

FEM polynomial filtering

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





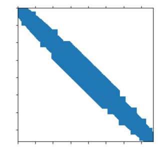


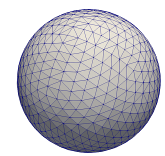
$$k(\mathbf{\Lambda}) = P(\mathbf{\Lambda}) \implies \mathbf{\Omega}^k_{\text{FEM}} = P(\mathbf{B}^{-1}\mathbf{A})$$

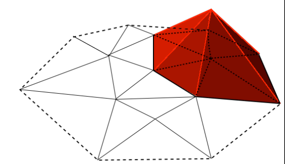
- $\mathbf{B}^{-1}\mathbf{A}$ is not symmetric.
- $\mathbf{B}^{-1}\mathbf{A}$ is full.
- $\mathbf{V}\mathbf{B}\mathbf{V}^\intercal = \mathbf{I}$.

Lumped FEM Laplacian

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







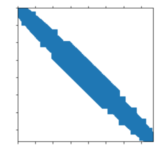
$$P\left(\mathbf{B}^{-1}\mathbf{A}\right) \approx P\left(\mathbf{D}^{-1}\mathbf{A}\right)$$

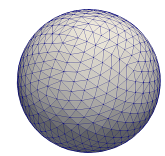
$$\mathbf{d}_{ii} = \sum_j \mathbf{b}_{ij}$$

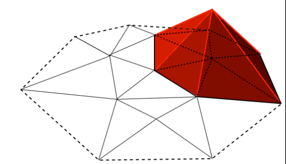
- $\mathbf{D}^{-1}\mathbf{A}$ is not symmetric.
- $\mathbf{D}^{-1}\mathbf{A}$ is sparse.

Lumped FEM Laplacian as a graph Laplacian

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



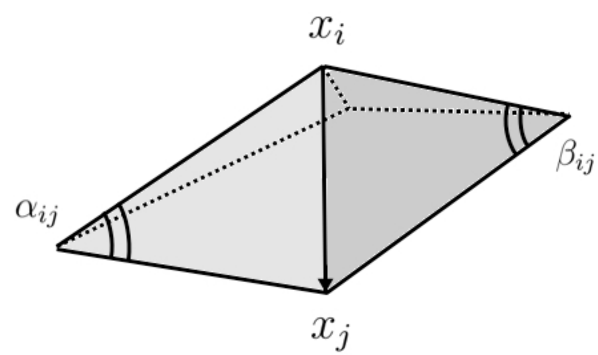




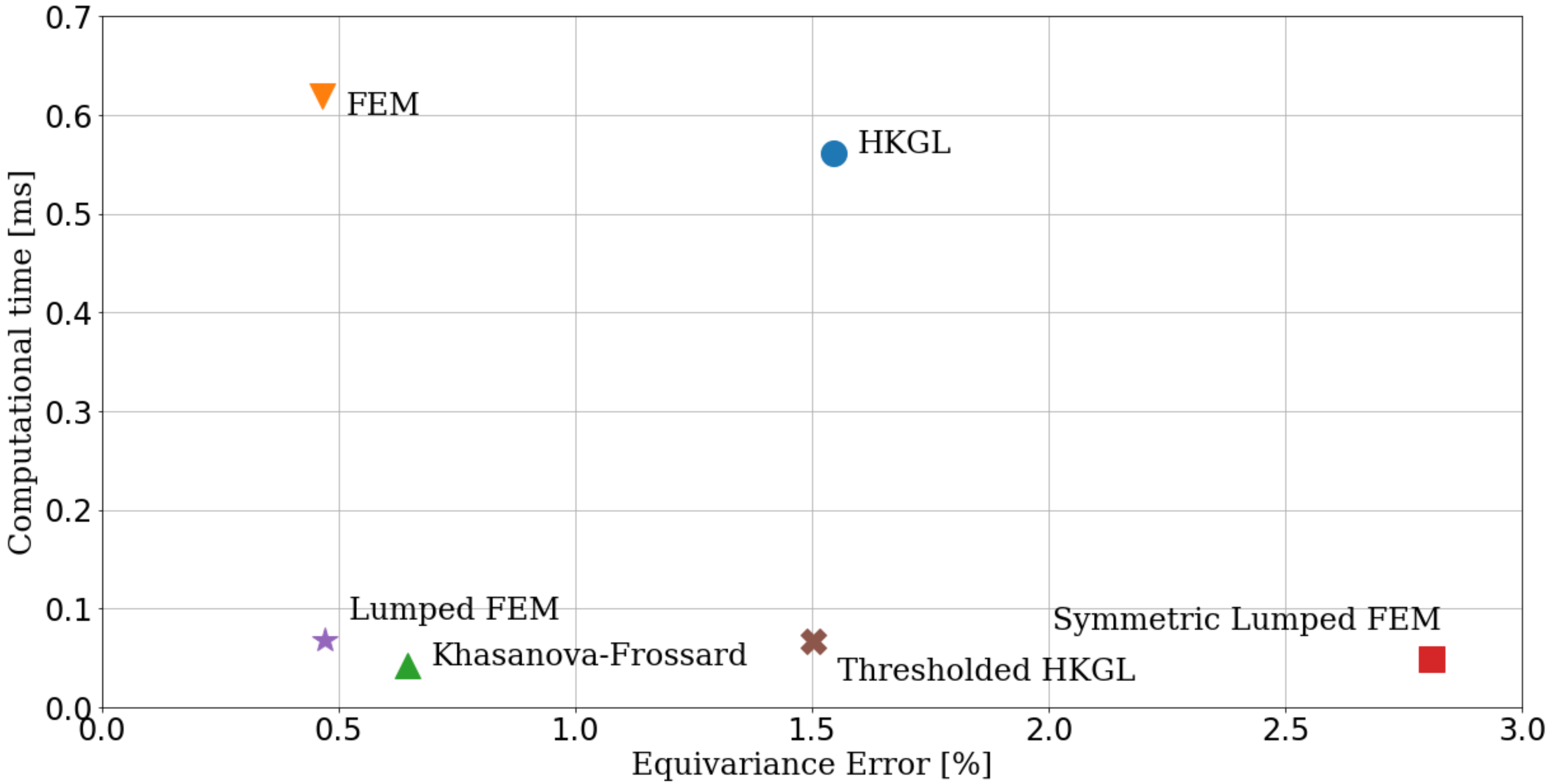
$$\mathbf{D}^{-1}\mathbf{A} :$$

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

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

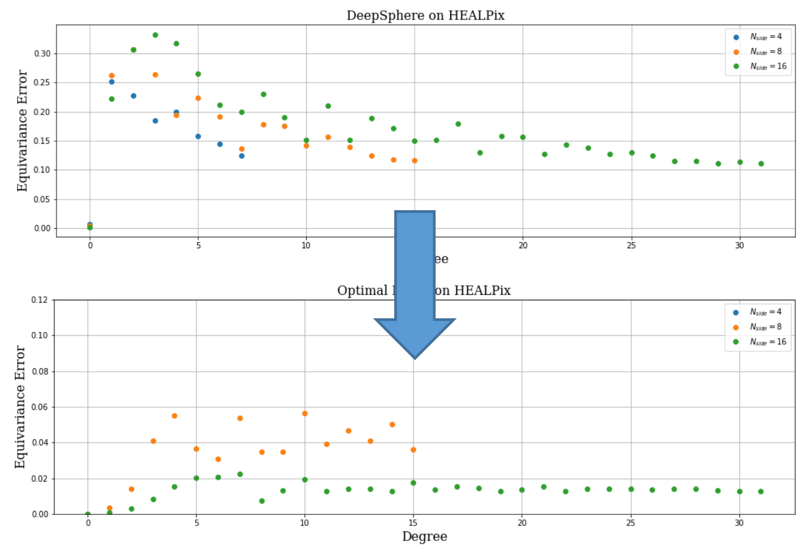


Equivariance error and computational time



Conclusions

1) We gained a better understanding of the HKG.
2) We put that knowledge in practice, improving the Equivariance Error of DeepSphere.



3) We investigated different Discrete Laplacians, from Differential Geometry to Numerical Mathematics
4) We used this knowledge to better understand the advantages and limitations of Graph Laplacians when it comes to non uniform sampling of the sphere.