1) $\mathcal{H} = \{f : \mathcal{X} \to \mathbb{R}\}$, $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $X = \{x_1, \dots, x_m\} \subset \mathcal{X}$

$\mathcal{H}$ RKHS $\Rightarrow \exists u^* \in \underset{u \in \mathcal{H}}{\arg\min}\, J(h)$ s.t. $u^* = \sum_{j=1}^{m} a_j u_j$ if $\underset{u \in \mathcal{H}}{\min}\, J(h)$ exists by Rep. Th'm

Consider $L(u, a) = \|u\|^2 + \langle a, u(X) - y \rangle \Rightarrow \frac{\partial L}{\partial u} = 2u + \langle a, u'(X) \rangle$ for some a

$\langle f, u' \rangle = \langle a, u'(X) \rangle \ \forall u' \in \mathcal{H} \Rightarrow f = aK_X$ since $\langle aK_X, u' \rangle = au'(X) = \langle a, u'(X) \rangle$

$\Rightarrow \frac{\partial L}{\partial u} = 2u + aK_X \overset{\sim}{=} 0 \Rightarrow u^* = -a/2\, K_X$

$u^*(X) = -a/2\, K(X, X) \overset{\sim}{=} y \Rightarrow a = -2K(X, X)^{-1} y \Rightarrow u^* = K_X K(X, X)^{-1} y$ ■

2a) $\varphi \in \mathcal{H}^*$ is BLF on $\mathcal{H}$ $\Rightarrow$ φ bounded on $\mathcal{H}_0$

$\Rightarrow \exists! \hat{\varphi} \in \mathcal{H}_0$ s.t. $\varphi(f) = \langle \hat{\varphi}, f \rangle \; \forall f \in \mathcal{H}_0$

Taking $\varphi(f) = f(x) \; \forall x \in \mathcal{X} \Rightarrow \langle \hat{\varphi}, f \rangle = \langle K_x, f \rangle$

$\Rightarrow \hat{\varphi} \equiv K_\varphi$ by uniqueness of Riesz Rep. $\Rightarrow \varphi(f) = \langle K_\varphi, f \rangle$ ∎

b) Consider $f \perp \text{span}\{K_{\varphi_1}, \dots, K_{\varphi_m}\} \Rightarrow \langle f, K_{\varphi_1} \rangle = \dots = \langle f, K_{\varphi_m} \rangle = 0$

$\Rightarrow \varphi_1(f) = \dots = \varphi_m(f) = 0$ by (a) $\Rightarrow \text{span}\{K_{\varphi_1}, \dots, K_{\varphi_m}\}^\perp = \{f \in \mathcal{H} : \varphi_j(f) = 0 \; \forall j \in [m]\}$ ∎

c) $\vec{\varphi}(u) = \sum_{j=1}^{m} \langle K_{\varphi_j}, u \rangle$ by (b)

Consider $L(u, \lambda) = \|u\|^2 + \langle \lambda, \vec{\varphi}(u) - y \rangle = \|u\|^2 + \sum_{j=1}^{m} \lambda_j (\langle K_{\varphi_j}, u \rangle - y_j)$

$\langle K_{\varphi_j}, f \rangle = K_{\varphi_j} \; \forall f \in \mathcal{H} \Rightarrow \langle K_{\varphi_j}, K_x \rangle = K_{\varphi_j}(x) \; \forall x \in \mathcal{X}$

$\Rightarrow \frac{dL}{du} = 2u + \sum_{j=1}^{m} \lambda_j \langle K_{\varphi_j}, u' \rangle = 2u + \sum_{j=1}^{m} \lambda_j K_{\varphi_j} \equiv 0 \Rightarrow u^* = -\frac{1}{2} \sum_{j=1}^{m} \lambda_j K_{\varphi_j}$

$\Rightarrow \varphi(u^*) = \varphi(-\frac{1}{2} \sum_{j=1}^{m} \lambda_j K_{\varphi_j}) = -\frac{1}{2} \sum_{j=1}^{m} \lambda_j \varphi(K_{\varphi_j}) = -\frac{1}{2} \sum_{j=1}^{m} \lambda_j \Theta_j \equiv \vec{y}$

$\Rightarrow -\frac{1}{2} \Theta \lambda = \vec{y} \Rightarrow \lambda = -2\Theta^{-1} \vec{y} = -2\alpha^* \Rightarrow u^* = \sum_{j=1}^{m} \alpha_j^* K_{\varphi_j}$ ∎

3a) $L = D - W \in \mathbb{R}^{n\times n}$, $\tilde{L} = D^{-1/2} L D^{-1/2}$

(λ, v) is eigpair of $\tilde{L}$

$\Rightarrow \lambda v = \tilde{L} v = D^{-1/2}(D-W)D^{-1/2} v = (D^{-1/2} D D^{-1/2} - D^{-1/2} W D^{-1/2}) v = (I - D^{-1/2} W D^{-1/2}) v$

$\Rightarrow \lambda D^{1/2} u = (I - D^{-1/2} W D^{-1/2}) D^{1/2} u = (D^{1/2} - D^{-1/2} W) u$

$\Rightarrow \lambda D u = (D - W) u = L u$

u solves $Lu = \lambda D u$

$\Rightarrow \lambda D^{1/2} v = L D^{-1/2} v = (D-W) D^{-1/2} v = (D^{1/2} - W D^{-1/2}) v = (D-W) D^{-1/2} v = L D^{-1/2} v$

$\Rightarrow \lambda v = D^{-1/2} L D^{-1/2} v = \tilde{L} v \Rightarrow (\lambda, v)$ is an eigpair of $\tilde{L}$ ■

b) $G = \{X, W\} = \bigcup_{j=1}^{M} G_j$ is M-connected $\Rightarrow G_j$ pathwise connected $\forall j \leq M$

$\Rightarrow L_{G_j} 1 = D_{G_j} 1_{G_j} - W_{G_j} 1_{G_j} = d - d = 0 \Rightarrow Nul(L_{G_j}) = 1_{G_j} \quad \forall j \leq M$

$$Lu = \begin{bmatrix} L_{G_1} & & 0 \\ & \ddots & \\ 0 & & L_{G_M} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} L_{G_1} & & 0 \\ & \ddots & \\ 0 & & L_{G_M} \end{bmatrix} \begin{bmatrix} \vec{u}_{G_1} \\ \vdots \\ \vec{u}_{G_M} \end{bmatrix} = 0 \Leftrightarrow L_{G_j} \vec{u}_{G_j} = 0 \quad \forall j \leq M$$

$\Rightarrow Nul(L) = \{u : Lu = 0\} = \{\vec{u}_{G_j} : L_{G_j} \vec{u}_{G_j} = 0\} = span\{1_{G_j} : j \in [M]\}$

$\tilde{L} v = 0 \Leftrightarrow Lu = 0$ by (a)

$\Rightarrow Nul(\tilde{L}) = \{v : \tilde{L} v = 0\} = \{D^{1/2} u : u \in Nul(L)\} = span\{D^{1/2} 1_{G_j} : j \in [M]\}$ ■

cont'd →

c) $D^{-1/2}_{ii} = \begin{cases} d(i)^{-1/2}, & i=j \\ 0 & \text{else} \end{cases}$

For $i \neq j$, $\tilde{L}_{ij} = \vec{D}_i^{-1/2^T}(D-W)_j \, d(j)^{-1/2} = d(i)^{-1/2}(-W_{ij})d(j)^{-1/2}$

$\tilde{L}_{ji} = \vec{D}_j^{-1/2^T}(D-W)_i \, d(i)^{-1/2} = d(j)^{-1/2}(-W_{ji})d(i)^{-1/2}$

No isolated vertices $\Rightarrow W_{ij} = W_{ji} \Rightarrow \tilde{L}_{ij} = \tilde{L}_{ji} \; \forall i,j \Rightarrow \tilde{L}$ Hermitian

$\Rightarrow \lambda_j = \min\limits_{\dim U = j}\{\max\limits_{\|x \in U\|=1}\{\langle x, \tilde{L}x\rangle\}\}$ by Courant-Fischer Weyl

Consider $x \in U$ s.t. $\|x\|=1$ and $U \subset \mathbb{R}^n$ is a subspace

$\langle x, \tilde{L}x\rangle = x^T\tilde{L}x = x^T D^{-1/2}(D-W)D^{-1/2}x = \hat{x}^T L \hat{x} = \langle \hat{x}, L\hat{x}\rangle \geq 0$ for $\hat{x} = D^{-1/2}x$

Since $D^{-1/2}$ is pos. def., $\operatorname{argmax}\limits_{x} \langle x, Lx\rangle = \operatorname{argmax}\limits_{\hat{x}} \langle \hat{x}, L\hat{x}\rangle \Rightarrow \|\hat{x}\|=1$

$\langle \hat{x}, L\hat{x}\rangle = \langle \hat{x}, (D-W)\hat{x}\rangle = \sum\limits_{i=1}^{n} d(i)\hat{x}_i^2 - \langle \hat{x}, W\hat{x}\rangle \leq \sum\limits_{i=1}^{n} d(i)\hat{x}_i^2 \leq \sum\limits_{i=1}^{n} d(i)$

$\Rightarrow \max\limits_{\|x \in U\|=1}\{\langle x, \tilde{L}x\rangle\} \leq \sum\limits_{i=1}^{n} d(i)$

$\operatorname{argmin}\limits_{\dim U = j}\{\langle \hat{x}, D\hat{x}\rangle\} \Rightarrow \sum\limits_{i=1}^{n} d(i) = 2 \cdot 1$ (i.e. each node is connected by 1 edge)

Thus, $\lambda_j = \min\limits_{\dim U = j}\{\max\limits_{\|x \in U\|=1}\{\langle x, \tilde{L}x\rangle\}\} \leq 2$ ∎

Following the same reasoning as above for $\min\limits_{\dim U = j}\{\max\limits_{\|x \in U\|=1}\{\langle x, Lx\rangle\}\}$:

$\operatorname{argmin}\limits_{\dim U = j}\{\sum\limits_{i=1}^{n} d(i)\} \Rightarrow \sum\limits_{i=1}^{n} d(i) \leq n-1$ by Gershgorin's Th'm (proven in AMATH 584)

since $L = \begin{bmatrix} d_1 & w_{12} & w_{13} & \\ w_{21} & d_2 & w_{23} & \\ & & & \ddots \end{bmatrix}$ with unnormalized rows, the bound on the eigenvalues can only be as strict as the worst case, where all off-diagonal entries of a row are ones.

Thus, the eigenvalues of L are not uniformly bounded

This computation exercise involves an analysis and comparison between the eigenvectors of the Laplacian matrix and the (vectorized) eigenfunctions of the Laplacian operator. The graph Laplacian is a matrix representation of a weighted graph that captures the geometry and connectivity of the graph. The Laplace operator is a second-order differential operator commonly used in the analysis of partial differential equations. The eigenvectors of the graph Laplacian are compared with the eigenfunctions of the Laplace operator to understand their relationship as we increase the number of evaluation points (and thus the domain of the Laplacian matrix).

<u>Notation</u>

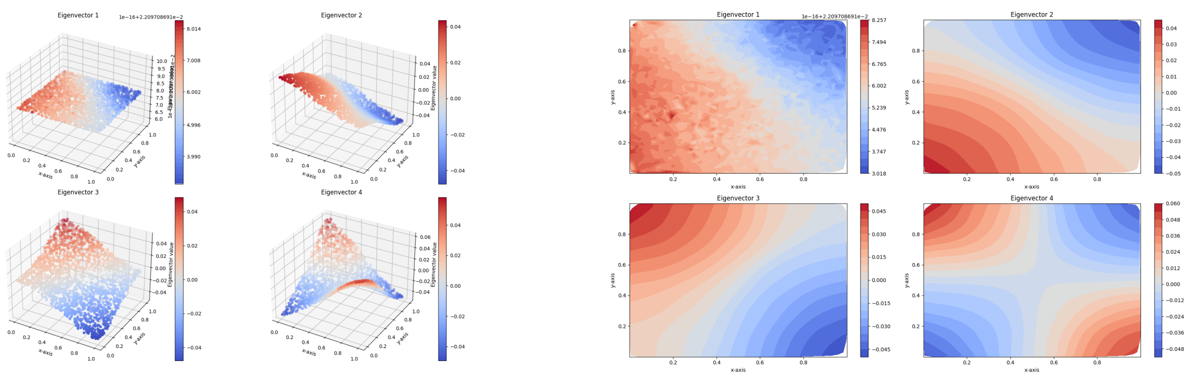
- **Omega**: sample space of evaluation points (subset of R^2)
- **X**: set of m evaluation points uniformly sampled from Omega (2xm real-valued matrix)
- **W**: weight matrix, defined through the kernel specified
- **Epsilon**: bandwidth of the kernel as function of the evaluation set size. Unless otherwise specified, assume the provided implementation scaled by C=1 is employed
- **D**: diagonal degree matrix whose entries are the row sums of W (edges of each node)
- **G**: weighted graph defined for the evaluation points and weights (G={X,W})
- **L**: unnormalized graph Laplacian matrix (L=D-W)
- ***L***: 2D Laplacian operator (negative sum of second partial derivatives, often denoted as nabla^2)

<u>Computational Tools</u>

- **Numpy**: standard scientific computing Python package
- **Scipy.spatial.distance.cdist**: Scipy's pairwise distance implementation (with metric set as 'euclidean')
- **Scipy.sparse.linalg.eigsh**: Scipy's eigendecomposition implementation optimized for sparse Hermitian matrices (note that L and *L* are symmetric positive definite by definition)

<u>Part 1</u>

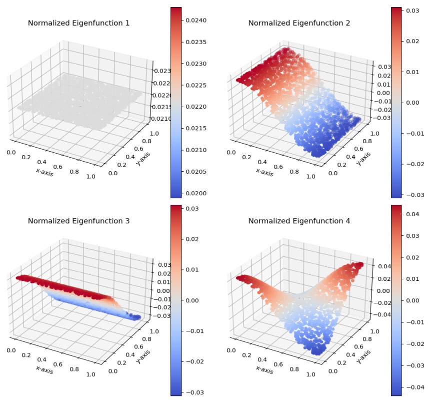
With m=2048 evaluation points, X was constructed by sampling uniformly over Omega=[0,1]^2. W was constructed from X, D from W, and L from D and W by definition. Performing a truncated eigendecomposition for the first 4 pairs on L yielded the following 3D scatter and contour plots:

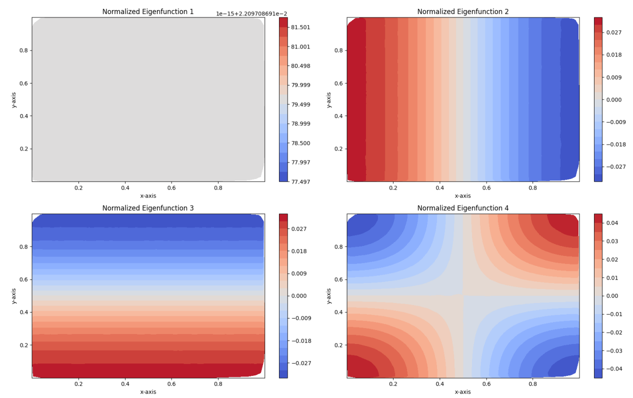


Not much can be said at this point as this is primarily a basis for comparison. Of note, however, is the fact that the plots resemble those of a 0-gradient for eigenvector 1, a 0-gradient saddle point for eigenvectors 2 and 3, and a min/max saddle point for eigenvector 4.

<u>Part 2</u>

Now we consider a similar setting for the Laplacian operator *L* with Neumann boundaries. As silicon-based computers can only handle quantized components at finite-precision values, the eigenvalue problem needs to be vectorized. This was done by pointwise evaluation of the eigenfunctions (i.e. solving the eigenvalue problem for the first 4 eigenvalues at the 2048 points sampled earlier), yielding these plots:

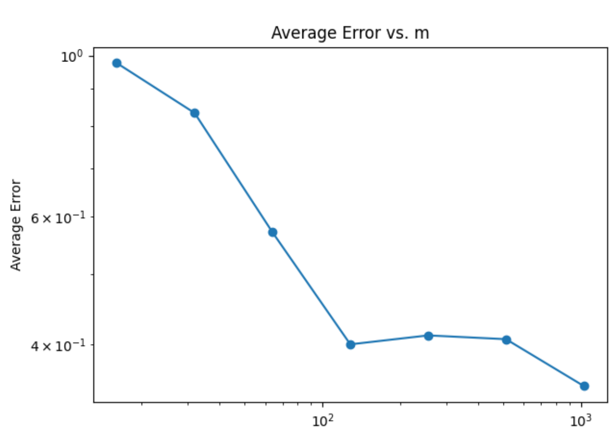




Their structure is nearly identical to the plots in part 1, with only minor differences in raw value and rotation about the z-axis. The plot for the first eigenfunction appears to be an outlier at first glance, however the actual values are similar. The eigenfunctions involve only cosine terms evaluated on rational multiples of pi and the first eigenfunction's arguments are both 0 (and hence exactly 1 for all points). This suggests a strong correspondence between the eigenvalues of L and the eigenfunctions of *L*.

Part 3

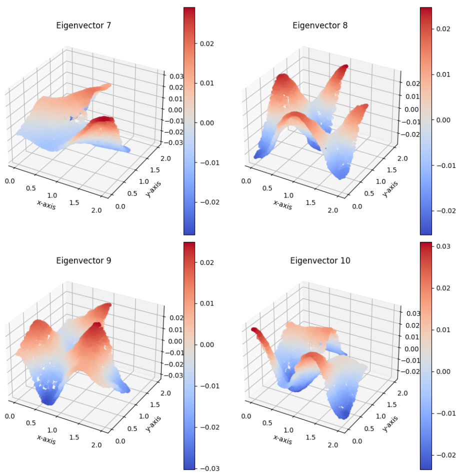
To further investigate the apparent correspondence observed between the results in parts 1 and 2, their spans over the same unit box sample space are compared. This is done through projectors: structure-preserving linear transformations with defining property P*P=P which leaves its image unchanged. First, we define corresponding linear maps --2 matrices with the eigenvectors of L and vectorized eigenfunctions of *L* as columns, respectively. Since the columns are linearly independent (different eigenvalues for each eigenvector), their projectors are computed through a transpose multiple as this guarantees symmetry by the Spectral Theorem. Now, applying the projections to one another and taking their difference provides a basis for quantifying the "closeness" of the respective subspaces. In this exercise, the quantification was done through the matrix extension of the Euclidean norm, the Frobenius norm. Applying this procedure for several uniformly-sampled domains at different values of m, each for 30 trials, yielded the following loglog plot:

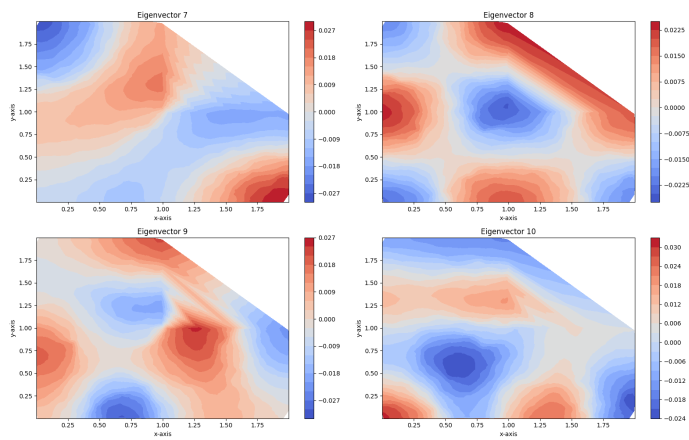


Since the true domain of *L* is infinite-dimensional, increasing m yields an improved approximation in theory. For the most part, the inverse relationship between m and the error affirms the correspondence seen between parts 1 and 2. This suggests convergence of L to *L* as m approaches infinity. I suspect the largest values of m deviating from the trend can be explained by loss of precision during the various operations on the float64 representations of values since the conducting the experiment in trials minimizes the contribution of sampling noise.

<u>Part 4</u>

The domain considered in the prior 3 experiments was a simple unit box. For the final experiment, Omega is taken to be L-shaped: [0,2]^2 without the upper-right quadrant. This shape of domain is commonly considered in the context of PDEs due to the sharp corner inducing a non-smooth, non-convex boundary making the problem significantly more complex and therefore difficult to approximate. Below are the plots for the 7^{th} through 10^{th} eigenvectors of L on m=2^13 points sampled from this space:





The plots are somewhat difficult to interpret, however the eigenvectors appear smooth, self-consistent, consistent with the domain, and consistent with my expectations of the spectrum of *L* for this problem. This indicates that the strength of the approximation of L to *L* is not confined to the unit box domain. Therefore, I conclude that the graph Laplacian is a good approximation the the 2D Laplacian operator and demonstrates converges for a reasonable number samples relative to the complexity of the domain.