EEL7680: 3D Shape Analysis
Programming Assignment 1

Max Marks: 100, Due Date:27/02/2025

1. (70 points) Consider a 3D model represented as a triangle mesh. Implement the following.
   (a) Find normal vector at each vertex.
   (b) Find the area of each triangle.
   (c) Find the Laplacian matrix ($\mathsf{L} = \mathsf{A}^{-1}\mathsf{M}$).
   (d) Find the eigenvectors corresponding to the first 30 smallest eigenvalues of the generalized eigenvalue problem $\mathsf{M}\boldsymbol{\phi}_i = \lambda_i \mathsf{A}\boldsymbol{\phi}, i = 1, 2, \ldots, 30$. Plot the eigenfunctions.
   (e) Find the mean curvature at each vertex.
   (f) Implement the mesh smoothing algorithm.

```python
import numpy as np
import scipy.sparse.linalg as sla
from plyfile import PlyData
import polyscope as ps
plydt = PlyData.read("mesh1.ply")
X = np.vstack((plydt['vertex']['x'],plydt['vertex']['y'],plydt['vertex']['z'])).T
tri_data = plydt['face'].data['vertex_indices']
T = np.vstack(tri_data)
#Add your code here to find the noraml vector, nvert=vertnormalfunc(X,T)
#Use below code to visualize the normal vector at each vertex
ps.init()
mymesh=ps.register_surface_mesh("my mesh", X, T)
ps_mesh.add_vector_quantity("nromal vectors", nvert, enabled=True)
ps.show()
#Add your code here to find M and A
n_eig=30
evals, evecs = sla.eigsh(M, n_eig, A, sigma=1e-8) # Use this to answer (c) part.
#Use below code to vizualize the eigenfunctions (install polyscope)
ps.init()
mymesh=ps.register_surface_mesh("my mesh", X, T)
for i in range(n_eig):
    mymesh.add_scalar_quantity("eigenvector_"+str(i), evecs[:,i], enabled=True)
ps.show()
#Write your code here for curvature esitmation
#mcur=yourfunct(laplcian matrix, nomals,etc.)
#Use below code to visualize the curvature
ps.init()
mymesh=ps.register_surface_mesh("my mesh", X, T)
mymesh.add_scalar_quantity("Normal curvature", mcur, enabled=True)
ps.show()
```

2. (30 points) Consider two meshes $\mathcal{M}$ and $\mathcal{N}$ (use meshes **'mesh2.ply'** and **'mesh1.ply'**). Find the point-to-point correspondences between these shapes by matching the spectral embeddings of the points. Visualize the best 100 estimated correspondences.