README

1)topo2dg.py- This script generates a scalar field using the function scalar_field_function(x, y). The scalar field is created using a grid with specified grid_size. The grid is created using np.linspace() and np.meshgrid() functions. Numpy library is necessary for this.

2)topo3d.py- This script creates a 3D plot of the scalar field sin(x)cos(y) and marks critical points (local minima and maxima) with red circles. It also connects the critical points sequentially with black lines.For this NumPy, matplotlib and scipy libraries are needed.

3)topoad1.py-this depicts a graph of critical points on the scalar field chosen. Libraries are numpy, matplotlib and scipy.

4)topompl.py-This code gives you a visual representation of the scalar field function in the 2D space defined by the **x** and **y** values. Libraries are numpy and matplotlib.

5)topoms.py- this code gives us the morse-smale visualisation, the libraries required are numpy,vtk and ttk. This code creates a scalar field using the scalar_field_function, then creates a VTK structured points dataset with the scalar field values. The Morse-Smale complex is computed using the TTK library's MorseSmaleComplex class. The output includes the critical points and separatrices, which are printed in the final step. Ideally the code should work with proper environment of libraries, our machines had technical glitches hence the libraries were not installed, although the code should work to best of our knowledge.

6) topoms2d1.py- This code will produce a plot of the scalar field sin(3x)cos(3y) with the Morse-Smale complex overlaid. Local minima are marked with red circles, and local maxima are marked with blue crosses. The Morse-Smale complex is drawn using black lines. Libraries used are numpy, matplotlib, scipy, networkx.

All the files have to be run in python script with all the necessary libraries mentioned for working of the code. Data used has been generated from the range of -2pi to 2pi with a grid size 100.