**Problem 1: Circle fitting**

*Download the source code at:*

**1. Problem**

Write a program in C++ to fit a circle on the given 3d points

**2. Solution**

The circle fitting method can be split into the following steps:

1. Using SVD (Singular Value Decomposition) find the best fitting plane to the set of mean-centered points.

2. Project the mean-centered points onto the fitting plane in new 2D coords.

3. Using method of least-squares fit a circle in the 2D coords and get circle center and radius.

4. Transform the circle center back to 3D coords. Now the fitting circle is specified by its center, radius and normal vector.

Please see the details in: *https://meshlogic.github.io/posts/jupyter/curve-fitting/fitting-a-circle-to-cluster-of-3d-points/*

*My program implemented the existing algorithm in C++ with STL and Eigen library.*

**3. Usage**

- Run "circle\_fitting.cc". It should read the 3d points data from "data.txt", plot a 2D-view of fitting circle and write the fitting circle points to "fitting\_circle.txt".

**Example**: *g++ test.cc -I /usr/include/python3.8 -lpython3.8 -I /usr/include/eigen3 -std=c++17*

- Run "plot.py". It should plot a 3D-view of fitting circle.

**Example**: *python3 plot.py*

- The results is stored in "Result" dicrectory.

**4. Tools**

- Ubuntu 20.04

- C++ 17

- Eigen 3.4.0

- Python version 3.8

- Matplotlib-cpp

**Note**: Matplotlibcpp is used to plot figures in C++. However, because I can’t find the way to plot 3D-view with Matplotlibcpp, "plot.py" is used as a workaround.