CS395T: Numerical Optimization for Graphics and AI: Homework II

1 Guideline

- Please complete **4** problems out of **8** problems, and please complete at least one problem in the theory session.
- You are welcome to complete more problems.

2 Programming

Each problem in this section counts as two.
Problem 1 and Problem 2. In this problem, we are interested in solving the following shape deformation problem using various optimization techniques. As discussed in class, we consider the setup where we have n points $\boldsymbol{p}_i$ in $\mathbb{R}^3$ and an edge set that connects adjacent points. The rest state of each point is denoted as $\boldsymbol{p}_i^{\text{rest}}$. With $\mathcal{H} \subset \{1, \cdots, n\}$ we denote the set of handles, where we want to move each $\boldsymbol{p}_i \in \mathcal{H}$ to their target location $\boldsymbol{h}_i$. This is formulated as solving the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{p}_1,\cdots,\boldsymbol{p}_n,R_1,\cdots,R_n}{\text{minimize}} \quad & \sum_{j \in \mathcal{N}(i)} \|R_i(\boldsymbol{p}_i^{\text{rest}} - \boldsymbol{p}_j^{\text{rest}}) - (\boldsymbol{p}_i - \boldsymbol{p}_j)\|^2 + \lambda \sum_{\boldsymbol{p}_i \in \mathcal{H}} \|\boldsymbol{p}_i - \boldsymbol{h}_i\|^2 \\ \text{subject to} \quad & R_i \in SO(3), 1 \leq i \leq n. \end{aligned} \tag{1}$$

where $\mathcal{N}(i)$ denotes the neighboring vertices of vertex i in edge set $\mathcal{E}$. Please apply at least two optimization techniques (e.g., Gradient Descent, Alternating Minimization, Gauss-Newton and Newton method) and compare their performance. Note that to turn this problem into unconstrained optimization, you are recommended to use the parameterization

$$R = \exp\begin{pmatrix} 0 & -c_z & c_y \\ c_z & 0 & -c_x \\ -c_y & c_x & 0 \end{pmatrix}.$$

To generate the dataset, you can start with a 20×20 grid and partition each cell into two triangles (so 800 triangles in total). The handles are placed at the corners of this grid and the grid center (5 handles in total). Please test our implementations on a 100×100 grid (20000 triangles in total).

Problem 3 and Problem 4. We are interested in finding the peaks (local maximums) of an un-normalized density function of form

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \exp(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2}{2\sigma^2}).$$

This can be done by starting from each one of the data points and apply coordinate ascent to maximize the value of the objective function $f(\boldsymbol{x})$. Note that the objective function is highly non-convex, so there may be multiple peaks, and it is important to start from many (or even all) of the input points.

- **Euclidean space $\mathbb{R}^3$.** In this case, the input points $\boldsymbol{x}_i$ are given by points in $\mathbb{R}^d$. We will test the case where the input points sampled from a mixture of Gaussians (with 2-4 mixture components).

- **Orthogonal matrices $SO(3)$.** In this case, the input points $\boldsymbol{x}_i$ are given by rotation matrices in $SO(3)$. We again consider the parameterization $R = exp(C)$, where $C = \begin{pmatrix} 0 & -c_z & c_y \\ c_z & 0 & -c_x \\ -c_y & c_x & 0 \end{pmatrix}$. The input points are generated by sampling $\boldsymbol{c} = (c_x, c_y, c_z)$ from a Gaussian distribution in $\mathbb{R}^3$. Again we will consider 2-4 mixture components (you may choose a small variance for each mixture component). Specifically, if we have two mixture components and use 25 samples per mixture component, then we have $n = 50$.

In both cases, we please try at least two methods, e.g., Steepest Ascent, Newton Method and Quasi-Newton Method. Matlab is preferred programming language for this assignment.

3 Theory

Problem 5. We say f is L-smooth (with constant $L > 0$) on $\mathcal{X}$ if f is continuously differentiable and $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$. Suppose f is convex and L-smooth, let $\boldsymbol{x}^\star$ be an optimal solution. With $\gamma = \frac{1}{L}$, the iterates of gradient descent method

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \gamma \nabla f(\boldsymbol{x}_t)$$

satisfy

$$f(\boldsymbol{x}_t) - \min_{\boldsymbol{x}} f(\boldsymbol{x}) \leq \frac{2L\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2}{t}.$$

Problem 6. Given a C^2 function $f(\boldsymbol{x}_1, \boldsymbol{x}_2)$ with a global minimizer solution $(\boldsymbol{x}_1^\star, \boldsymbol{x}_2^\star)$. Suppose the Hessian matrix

$$H^\star = H_f(\boldsymbol{x}_1^\star, \boldsymbol{x}_2^\star) \succ 0.$$

Consider alternating minimization, which computes the following quantity at each iteration:

$$\begin{aligned} \boldsymbol{x}_1^{(k+1)} &= \underset{\boldsymbol{x}_1}{\operatorname{argmin}}\ f(\boldsymbol{x}_1, \boldsymbol{x}_2^{(k)}) \\ \boldsymbol{x}_2^{(k+1)} &= \underset{\boldsymbol{x}_2}{\operatorname{argmin}}\ f(\boldsymbol{x}_1^{(k+1)}, \boldsymbol{x}_2) \end{aligned} \tag{2}$$

Show that for any $\epsilon > 0$, there exists a converging radius $r(\epsilon)$ so that starting from any initial solution $(\boldsymbol{x}_1^{(0)}, \boldsymbol{x}_2^{(0)})$, where

$$\max\left(\|\boldsymbol{x}_1^{(0)} - \boldsymbol{x}_1\|, \|\boldsymbol{x}_2^{(0)} - \boldsymbol{x}_2\|\right) \leq r(\epsilon),$$

we have for $k = 0, 1, \cdots$,

$$\begin{aligned} \|\boldsymbol{x}_1^{(k+1)} - \boldsymbol{x}_1^\star\| &\leq \left((\frac{1 - \kappa(H^\star)}{1 + \kappa(H^\star)})^2 + \epsilon \right) \cdot \|\boldsymbol{x}_1^{(k)} - \boldsymbol{x}_1^\star\|, \\ \|\boldsymbol{x}_2^{(k+1)} - \boldsymbol{x}_2^\star\| &\leq \left((\frac{1 - \kappa(H^\star)}{1 + \kappa(H^\star)})^2 + \epsilon \right) \cdot \|\boldsymbol{x}_2^{(k)} - \boldsymbol{x}_2^\star\|, \end{aligned} \tag{3}$$

where $\kappa(H^\star) = \lambda_{\min}(H^\star)/\lambda_{\max}(H^\star)$.

Problem 7. Derive similar formula for 'high-order' alternating minimization, which splits the variable $\boldsymbol{x} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$. The alternating minimization formula is given by:

$$\boldsymbol{x}_i^{(k+1)} = \underset{\boldsymbol{x}_i}{\operatorname{argmin}}\ f\big(\boldsymbol{x}_1^{(k+1)}, \cdots, \boldsymbol{x}_{i-1}^{(k+1)}, \boldsymbol{x}_i^{(k)}, \boldsymbol{x}_{i+1}^{(k)}, \cdots, \boldsymbol{x}_n^{(k)}\big), \quad 1 \leq i \leq n,\ k = 0, 1, \cdots \tag{4}$$

Note that for this problem, you need to write down the theorem and then the proof.

Problem 8. Let us go back to the second programming assignment and consider the mixture of Gaussian in 1D, i.e.,

$$f(x) = \sum_{i=1}^{n} \exp(-\frac{(x - x_i)^2}{2\sigma^2}).$$

- If we start from all the input points x_i and perform gradient ascent, do we find all the local maximums? If the answer is Yes, please give a proof. If the answer is No, please give counter examples.
- Does the same argument hold in high dimensions?