MATH-UA 253/MA-UY 3204 - Fall 2022 - Homework #5

Problem 1 (the secant condition). Prove that the secant condition holds for the following two formulas:

1. $B_{k+1} = B_k + \frac{(y_k - B_k s_k)v^\top}{v^\top s_k}$, assuming $v^\top s_k \neq 0$.
2. The BFGS update.

Problem 2 (Lennard-Jones potential). The Lennard-Jones potential is:

$$V(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right]. \tag{1}$$

The force due to this potential is just $F(r) = -dV/dr$. If you plot F (or do an image search), you will see that there is net zero force if $r = \sigma$, a very strong repelling force for $r < \sigma$, and a very weak attractive force if $r > \sigma$. In this problem, we will explore doing a simple molecular dynamics simulation in 2D using the Lennard-Jones potential. Consider a set of m particles in $\mathbb{R}^m$ with positions given by (x_i, y_i), $i = 1, \ldots, m$. Let r_{ij} be the distance between particles i and j. The total potential energy for this system of particles is:

$$f(x_1, y_1, \ldots, x_m, y_m) = \sum_{i=1}^{m} \sum_{i \neq j} V(r_{ij}) \tag{2}$$

Do the following:

1. Set $\sigma = \epsilon = 1$, and develop an algorithm for minimizing f using BFGS.
2. Apply this algorithm to systems with $m = 3, 4, \ldots 10$ particles and visualize your results:
   - Make sure to initialize randomly, and note that if your particles are too far apart initially, it will take a while to reach a local minimum because of the weak attractive force. On the other hand, because of the strong repelling forces, it is important to use a backtracking line search.
   - Plot your local minimizers by using a scatter plot for the positions of (x_i, y_i). Additionally, draw a circle of radius $\sigma = 1$ surrounding each particle. It may also help to plot each dot/circle pair using a different color.
   - There are numerous local minimizers for this system. See how many you can find.

Problem 3 (Fermat's principle). Consider a particle traveling with speed $c(x, y)$ meters per second. If we assume that the particle travels between two points $\boldsymbol{r}_0$ and $\boldsymbol{r}_1$ following the minimum travel time trajectory $\boldsymbol{r}(t)$, where $\boldsymbol{r}(0) = \boldsymbol{r}_0$ and $\boldsymbol{r}(1) = \boldsymbol{r}_1$, then $\boldsymbol{r} : [0, 1] \to \mathbb{R}^2$ can be found by minimizing:

$$\text{minimize} \qquad F[\boldsymbol{r}] = \int_0^1 \frac{\|\boldsymbol{r}'(t)\|}{c(\boldsymbol{r}(t))} dt. \tag{3}$$

This is a minimization problem where the variable being minimized over is a function instead of a vector. One approach to solving this problem is to use the calculus of variations, which is outside

the scope of this class. Another approach is to approximate the integral in (3) using a quadrature rule and then solve the resulting finite-dimensional minimization problem.

Consider solving this problem for $(x, y) \in \Omega = [0, 1] \times [0, 1]$, where the particle speed is:

$$c(x, y) = \frac{1}{\sqrt{4 - 3x - y}}. \tag{4}$$

One simple but useful quadrature rule is the composite trapezoid:

$$\int_a^b f(t)dt = \sum_{i=1}^{m} \frac{f(t_{i-1}) + f(t_i)}{2} \Delta t_i + O\left(\max_{1 \le i \le m} \Delta t_i^2 \right), \tag{5}$$

where $a = t_0 < t_1 < \cdots < t_m = b$, and where $\Delta t_i = t_i - t_{i-1} > 0$.

Proceed in the following steps:

1. Use the composite trapezoid rule to approximate the integral in (3) for an arbitrary number of points (i.e., $m + 1$ points, as above). To keep things simple, use uniform spacing—i.e., choose the t_i's such that $\Delta t_i = h$ for all i. Denote your cost function by $f(t_0, \ldots, t_m)$ and write down the new finite-dimensional minimization problem to solve. Compute the gradient of $\nabla f(t_0, \ldots, t_m)$.

2. Let $\boldsymbol{r}_0 = (0, 0)$ and $\boldsymbol{r}_1 = (1, 1)$. Solve your minimization problem for $m = 5, 10, 20, 40, 80, 160$ using gradient descent, SR1, and BFGS. You may need to use a backtracking line search to get your iteration to converge (*Hint*: what conditions do you need to check for each of these methods while backtracking?). You should be able to reach roughly the same minimizers with each method.

3. Plot the level sets of c along with your computed trajectories (doesn't matter from which iteration) on Ω.

4. Let f_m^* be the minimizing value for a given choice of m. Plot $|f_{10}^* - f_5^*|$, $|f_{20}^* - f_{10}^*|$, …, $|f_{160}^* - f_{80}^*|$ versus $m = 5, 10, \ldots, 160$ on a loglog plot. What do you observe?

5. For each m and for each optimization algorithm, plot the magnitude of the step size versus the iteration count on a loglog plot. What do you observe? How do the methods compare?