MATH-UA 252/MA-UY 3204 - Fall 2022 - Homework #3

This homework only has one problem, but it is a bit long, so start early!

The Thomson problem. Consider a system of N particles at positions $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ (where $\boldsymbol{x}_i \in \mathbb{R}^3$), where particle i has charge q_i. The electrostatic potential energy stored in this system is:

$$U_E(N) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j \neq i}^{N} k_e \frac{q_i q_j}{r_{ij}}, \tag{1}$$

where $r_{ij} = |\boldsymbol{x}_i - \boldsymbol{x}_j|$. If we normalize the units so $k_e = 1$ and assume that $q_i = 1$ for all i, then $U_E(N)$ simplifies to:

$$U_E(N) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j \neq i}^{N} \frac{1}{r_{ij}}. \tag{2}$$

Now, assume that we have N electrons which are *constrained to lie on the surface of the unit sphere in* $\mathbb{R}^3$. The *Thomson problem* is the following:

<u>For $N > 0$, find the configuration of electrons on the unit sphere such that $U_E(N)$ is minimized.</u>

That is, solve:

$$\begin{array}{ll} \text{minimize} & U_E(N) \\ \text{subject to} & |\boldsymbol{x}_i| = 1, \qquad i = 1, \ldots, N. \end{array} \tag{3}$$

For this problem, you should write a version of PGD to solve the Thomson problem. Comments:

- You will need to use line search to get your iteration to converge. It's recommended to just use backtracking line search.
- You will need a reasonable way of distributing points on the sphere. This is simple:
  1. For each i, choose $\boldsymbol{x}_i = (x_i, y_i, z_i)$ randomly such that $x_i, y_i, z_i \sim \mathcal{N}(0, 1)$ independently and identically. On Tuesday (9/27) we will discuss line search in more detail.
  2. Set $\boldsymbol{x}_i := \boldsymbol{x}_i / |\boldsymbol{x}_i|$.

  You can do this in numpy as follows:

  ```
  X = np.random.randn(N, 3)
  X /= np.sqrt(np.sum(X**2, axis=1)).reshape(-1, 1)
  ```

To visualize your solution, you have three options:

1. Each point can be written in spherical coordinates:

   $$(x_i, y_i, z_i) = (\cos(\phi_i)\sin(\theta_i), \sin(\phi_i)\sin(\theta_i), \cos(\theta_i)), \tag{4}$$

   where $0 \leq \phi_i < 2\pi$ and $0 \leq \theta_i \leq \pi$. You can use matplotlib to make a 2D scatter plot of the (ϕ_i, θ_i) coordinates.

2. You can use matplotlib's mplot3d to make a 3D plot of the (x_i, y_i, z_i) coordinates (see this link for some ideas).

3. You can use PyVista.

All three options are fine—it is your choice.

Now, once your solver works, do the following:

1. Solve the Thomson problem for several choices of N where $3 \leq N \leq 14$, and verify that your results match the picture here:

   `https://tracer.lcc.uma.es/problems/thomson/thomson.html`

2. The URL above also gives the following fit to the value of the potential for different N:

$$U_{\text{approx}}(N) = \frac{N^2}{2}\left(1 - aN^{-1/2} + bN^{-3/2}\right), \tag{5}$$

   where $a = 1.10461$ and $b = 0.137$. Try to solve the Thomson problem for a few choices of N greater than $N = 14$. The Thomson problem is *nonconvex* and has numerous local minima. If you re-run your optimization algorithm for different initializations, you may get different minimizing values. Make a plot where:

   - The horizontal axis is N, and the vertical axis is U—both axes should be linear (so, use `plt.plot`).
   - Plot U_{approx} using the formula above.
   - Make a scatter plot of your results for comparison.