

# MATH-UA 252/MA-UY 3204 - Fall 2022 - Homework #1

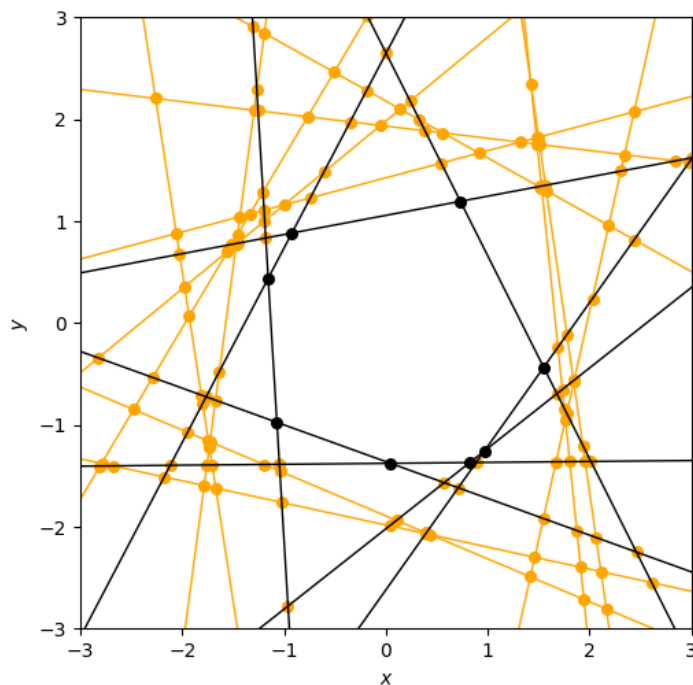
**Problem 1.** A convex polyhedron  $P \subseteq \mathbb{R}^3$  can be written as the set of points which satisfies:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &\leq b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 &\leq b_n \end{aligned} \tag{1}$$

Note that the surface of a convex polyhedron can be split up into vertices, edges, and faces. Solve the following subproblems:

1. Give an algorithm which runs in  $O(n^4)$  time which finds all of the vertices of  $P$ . Explain why your algorithm has this time complexity.
2. Explain how to find the edges of the polyhedron.
3. Note that some of the constraints in (1) may be redundant. Explain how to determine which constraints are redundant as a byproduct of running your algorithm from Part 1.
4. **Bonus.** The faces of  $P$  are convex (why?) and are bounded by three or more of  $P$ 's edges. Each face corresponds to exactly one active constraint. Give an algorithm which finds each face  $F$  of  $P$  in terms of its vertices. Consequently, explain how to determine the active constraint corresponding to  $F$ .

You may find this plot inspirational:



**Problem 2.** Program your algorithm from Problem 1 (Parts 1 and 2). To test it, you can generate a test convex polyhedron centered about the origin using the following code snippet:

```

n = 20
d = 3
A = np.random.randn(n, d)
A /= np.sqrt(np.sum(A**2, axis=1)).reshape(-1, 1)
b = 1 + np.random.random(n)

```

This code with  $d = 2$  was used to generate the plot above. There will be redundant constraints (the orange lines shown above).

1. Make a 3D plot of a few random convex polyhedra using `mplot3d` or `PyVista`. Plot the vertices (using a 3D scatter plot in `mplot3d` or as sphere in `PyVista`) and the edges (using lines in `mplot3d` or as cylinders in `PyVista`). Make sure to plot the axes and make a list of the vertices in each polyhedron. You can come up with some alternate way to plot the vertices and edges provided that the plot is understandable. If in doubt feel free to ask me.
2. **Bonus.** Program your algorithm to find the faces of the polyhedron and plot them. Use a different color for each face.

**Problem 3.** Adapt your algorithm from Problem 1 to the case of  $d = 2$  (a convex polygon). Make a plot of the resulting polygon. Additionally, set up a random cost function  $f(x) = c^\top$  using:

```
c = np.random.randn(2)
```

Plot the level sets of this function over  $P$  and include a colorbar (`plt.colorbar`). How can you use the result of your algorithm to minimize  $f(x)$ ? Plot the resulting minimizer.