4.1 (4 points) Consider the Bregman divergence \( D(x, y) = \sum_i x_i \log(x_i/y_i) - (x_i - y_i) \), which is known as the generalized KL divergence.

Show that the projection with respect to this Bregman divergence on the simplex, i.e., \( \Delta_n = \{ x \in \mathbb{R}_+^n \mid 1^T x = 1 \} \), amounts to a simple renormalization \( y \to y/\|y\|_1 \).

4.2 (4 points) High dimensional problems, mirror descent, and gradient descent. We consider using mirror descent versus projected subgradient descent to solve the non-smooth minimization problem

\[
\minimize f(x) = \max_{i \in \{1, \ldots, m\}} \{ a_i^T x + b_i \} \quad \text{subject to } x \in \Delta_n = \{ z \in \mathbb{R}_+^n \mid z^T 1 = 1 \}.
\]

Implement mirror descent with the choice \( h(x) = \sum_{i=1}^n x_i \log x_i \) and projected subgradient descent for this problem. (You will need to project onto the simplex efficiently for this to be a reasonable method at all.) You will compare the performance of these two methods.

Generate random problem data for the above objective with \( a_i \) drawn as i.i.d. \( N(0, I_{n \times n}) \) (multivariate normals) and \( b_i \) drawn i.i.d. \( N(0, 1) \), where \( n = 500 \) and \( m = 50 \). Solve the problem using CVX (or Convex.jl or CVXPY), then run mirror descent and projected gradient descent on the same data for 100 iterations. Run each method with constant stepsizes \( \alpha \in \{ 2^{-12}, 2^{-11}, \ldots, 2^6, 2^7 \} \). Repeat this 25 times, then plot the average optimality gap \( f(x^k) - f(x^*) \) or \( f_{\text{best}}^k - f(x^*) \) as a function of iteration for the best stepsize (chosen by smallest optimality gaps) for each method. Which method gives the best performance?

4.3 (4 points) Maximum volume ellipsoid vs Chebyshev center method. Consider the convex set

\[
C = \{ x \mid Ax \preceq b \},
\]

where \( A \in \mathbb{R}^{n \times d} \) and \( b \in \mathbb{R}^d \). The data files Amatrix and bvector are available on Canvas.

(a) (2 points) Find the center of the maximum volume ellipsoid in \( C \) and the center of the largest Euclidean ball in \( C \). You may use CVX/CVXPY. Hint: See 364a slides for calculating the maximum volume ellipsoid.
(b) (2 points) Denote the two centers (vectors in $\mathbb{R}^d$) in part (a) by $x_{\text{ellipsoid}}$ and $x_{\text{ball}}$ respectively. Let $g \in \mathbb{R}^d$ be the all-ones vector. We will consider the cuts $g^T(x - x_{\text{ball}}) \geq 0$ and $g^T(x - x_{\text{ellipsoid}}) \geq 0$. Estimate the volume ratios

$$R_{\text{ellipsoid}} := \frac{\text{vol}\left( \{ g^T(x - x_{\text{ellipsoid}}) \geq 0 \} \cap C \right)}{\text{vol}(C)},$$

and

$$R_{\text{ball}} := \frac{\text{vol}\left( \{ g^T(x - x_{\text{ball}}) \geq 0 \} \cap C \right)}{\text{vol}(C)},$$

by generating $M = 10^6$ i.i.d. uniformly distributed random vectors in $[-0.5, +0.5]^d$ (i.e., $x = \text{rand}(d,1)-0.5$ for $M$ trials). Hint: Let $M_C$ be number of random vectors that satisfy $Ax \preceq b$. Let $M_{\text{ellipsoid}}$ be the number of random vectors that satisfy $Ax \preceq b$ and $g^T(x - x_{\text{ellipsoid}}) \geq 0$. Similarly, let $M_{\text{ball}}$ be the number of random vectors that satisfy $Ax \preceq b$ and $g^T(x - x_{\text{ball}}) \geq 0$. The volume ratios can be estimated by

$$R_{\text{ellipsoid}} \approx \frac{M_{\text{ellipsoid}}}{M_C},$$

and

$$R_{\text{ball}} \approx \frac{M_{\text{ball}}}{M_C}.$$

4.4 (5 points) Kelley’s cutting-plane algorithm. We consider the problem of minimizing a convex function $f : \mathbb{R}^n \to \mathbb{R}$ over some convex set $C$, assuming we can evaluate $f(x)$ and find a subgradient $g \in \partial f(x)$ for any $x$. Suppose we have evaluated the function and a subgradient at $x^{(1)}, \ldots, x^{(k)}$. We can form the piecewise-linear approximation

$$\hat{f}^{(k)}(x) = \max_{i=1,\ldots,k} \left( f(x^{(i)}) + g^{(i)T}(x - x^{(i)}) \right),$$

which satisfies $\hat{f}^{(k)}(x) \leq f(x)$ for all $x$. It follows that

$$L^{(k)} = \inf_{x \in C} \hat{f}^{(k)}(x) \leq p^*,$$

where $p^* = \inf_{x \in C} f(x)$. Since $\hat{f}^{(k+1)}(x) \geq \hat{f}^{(k)}(x)$ for all $x$, we have $L^{(k+1)} \geq L^{(k)}$.

In Kelley’s cutting-plane algorithm, we set $x^{(k+1)}$ to be any point that minimizes $\hat{f}^{(k)}$ over $x \in C$. The algorithm can be terminated when $U^{(k)} - L^{(k)} \leq \epsilon$, where $U^{(k)} = \min_{i=1,\ldots,k} f(x^{(i)})$.

(a) (3 points) Use Kelley’s cutting-plane algorithm to minimize the piecewise-linear function

$$f(x) = \max_{i=1,\ldots,m} (a_i^T x + b_i)$$

that we have used for other numerical examples, with $C$ the unit cube, i.e., $C = \{ x \mid \| x \|_{\infty} \leq 1 \}$. Generate the same data we used before using
n = 20; % number of variables
m = 100; % number of terms
randn('state',1);
A = randn(m,n);
b = randn(m,1);

You can start with \( x^{(1)} = 0 \) and run the algorithm for 40 iterations. Plot \( f(x^{(k)}) \), \( U^{(k)} \), \( L^{(k)} \) and the constant \( p^* \) (on the same plot) versus \( k \).

(b) (2 points) Repeat for \( f(x) = \|x - c\|_2^2 \), where \( c \) is chosen from a uniform distribution over the unit cube \( C \). (The solution to this problem is, of course, \( x^* = c \).)