DOL

Homework 1

Parisa Mohammadi
810101509

Contents

Q1) Determine the curvature of the functions below. Your responses can be: affine, convex, concave, and none (meaning, neither convex nor concave).

(a) $f(x) = \min\{2, x, \sqrt{x}\}$, **with dom** $f = \mathbb{R}_+$

$$\begin{aligned} f_1(x) &= 2 \implies f_1''(x) = 0 \text{ (concave)} \\ f_2(x) &= x \implies f_2''(x) = 0 \text{ (concave)} \\ f_3(x) &= x^{1/2} \implies f_3''(x) = -\frac{1}{4}x^{-3/2} \leq 0 \text{ (concave)} \\ f(x) &= \min\{f_1, f_2, f_3\} \text{ (minimum of concave functions)} \\ &\implies \text{concave} \end{aligned}$$

(b) $f(x_1, x_2) = x_1/x_2$, **with dom** $f = \mathbb{R}_{++}$

$$\begin{aligned} \nabla f(x) &= \begin{bmatrix} 1/x_2 \\ -x_1/x_2^2 \end{bmatrix} \\ H = \nabla^2 f(x) &= \begin{bmatrix} 0 & -1/x_2^2 \\ -1/x_2^2 & 2x_1/x_2^3 \end{bmatrix} \\ \det(H) &= (0)(2x_1/x_2^3) - (-1/x_2^2)^2 = -1/x_2^4 \\ \det(H) &< 0 \text{ on } \mathbb{R}_{++} \implies H \text{ is indefinite.} \\ &\implies \text{none} \end{aligned}$$

(c) $f(x, y) = \sqrt{x \min\{y, 2\}}$, **with dom** $f = \mathbb{R}_+^2$

$$\begin{aligned} g(y) &= \min\{y, 2\} \text{ (concave)} \\ h(x) &= x \text{ (concave)} \\ G(a, b) &= \sqrt{ab} \text{ (geometric mean, concave and non-decreasing)} \\ f(x, y) &= G(h(x), g(y)) \text{ (composition)} \\ &\implies \text{concav} \end{aligned}$$

(d) $f(x, y) = (\sqrt{x} + \sqrt{y})^2$, **with dom** $f = \mathbb{R}_+^2$

$$\begin{aligned} f(x, y) &= x + y + 2\sqrt{xy} \\ g(x, y) &= x + y \text{ (affine, thus concave)} \\ h(x, y) &= 2\sqrt{xy} \text{ (geometric mean, concave)} \\ f &= g + h \text{ (sum of concave functions)} \\ &\implies \text{concave} \end{aligned}$$

(e) $f(x_1, x_2) = x_1^{\alpha} x_2^{1-\alpha}$, **where** $0 \leq \alpha \leq 1$, **on** $\mathbb{R}_{++}^2$

$$\begin{aligned} \text{Exponents: } a = \alpha \geq 0, \quad b = 1 - \alpha \geq 0 \\ \text{sum of exponents: } a + b = \alpha + (1 - \alpha) = 1 \\ \text{since } a + b \leq 1 \implies \text{concave} \end{aligned}$$

(f) $f(x) = -\exp(-g(x))$ **where** $g : \mathbb{R}^n \to \mathbb{R}$ **has a convex domain and satisfies**

$$\begin{bmatrix} \nabla^2 g(x) & \nabla g(x) \\ \nabla g(x)^T & 1 \end{bmatrix} \succeq 0$$

$$\begin{aligned}
\text{Given: } & \begin{bmatrix} \nabla^2 g(x) & \nabla g(x) \\ \nabla g(x)^T & 1 \end{bmatrix} \succeq 0 \\
& \implies \nabla^2 g(x) - \nabla g(x)(1)^{-1} \nabla g(x)^T \succeq 0 \\
& \implies \nabla^2 g(x) - \nabla g(x) \nabla g(x)^T \succeq 0 \\
\text{hessian of } f(x) : & \\
\nabla^2 f(x) &= \exp(-g(x)) \left[\nabla^2 g(x) - \nabla g(x) \nabla g(x)^T \right] \\
\text{since } \exp(-g(x)) &> 0 \text{ and } [\dots] \succeq 0 \text{ (given)} \\
\nabla^2 f(x) &\succeq 0 \\
& \implies \text{convex}
\end{aligned}$$

Q2) Consider the problem of projecting a point $a \in \mathbb{R}^n$ on the unit ball in ℓ_1-norm:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|x-a\|_2^2 \\ \text{subject to} & \|x\|_1 \le 1. \end{array}$$

Derive the dual problem and describe an efficient method for solving it. Explain how you can obtain the optimal x from the solution of the dual problem.

1. the dual problem

The primal problem has $f_0(x) = \frac{1}{2}\|x-a\|_2^2$ and $f_1(x) = \|x\|_1 - 1 \le 0$.

Step 1: the lagrangian We introduce one dual variable $\lambda \ge 0$.

$$L(x,\lambda) = f_0(x) + \lambda f_1(x) = \frac{1}{2}\|x-a\|_2^2 + \lambda(\|x\|_1 - 1)$$

$$L(x,\lambda) = \left(\frac{1}{2}\sum_{i=1}^n (x_i - a_i)^2 + \lambda \sum_{i=1}^n |x_i|\right) - \lambda$$

Step 2:the dual function $g(\lambda)$ The dual function $g(\lambda) = \inf_x L(x,\lambda)$ is separable.

$$g(\lambda) = -\lambda + \sum_{i=1}^n \inf_{x_i}\left(\frac{1}{2}(x_i - a_i)^2 + \lambda|x_i|\right)$$

$p_i(\lambda) = \inf_{x_i}\left(\frac{1}{2}(x_i-a_i)^2 + \lambda|x_i|\right)$. The minimizing x_i^* is found by setting the subgradient to 0:

$$0 \in (x_i^* - a_i) + \lambda\partial|x_i^*| \implies a_i - x_i^* \in \lambda\partial|x_i^*|$$

The solution is the soft-thresholding operator $x_i^*(\lambda) = S_\lambda(a_i) = \text{sign}(a_i)\max(0, |a_i| - \lambda)$.

We find the $p_i(\lambda)$ by substituting x_i^* in:

- If $|a_i| \le \lambda$: $x_i^* = 0$, so $p_i(\lambda) = \frac{1}{2}(0-a_i)^2 + \lambda|0| = \frac{1}{2}a_i^2$.
- If $|a_i| > \lambda$: $x_i^* = a_i - \lambda \cdot \text{sign}(a_i)$. $p_i(\lambda) = \frac{1}{2}(\lambda^2) + \lambda(|a_i| - \lambda) = \lambda|a_i| - \frac{1}{2}\lambda^2$.

So, $g(\lambda) = -\lambda + \sum_{i=1}^n p_i(\lambda)$.

Step 3: the dual problem

$$\max_{\lambda \ge 0} \quad g(\lambda) = \max_{\lambda\ge 0}\left(-\lambda + \sum_{i:|a_i|\le\lambda} \frac{1}{2}a_i^2 + \sum_{i:|a_i|>\lambda}\left(\lambda|a_i| - \frac{1}{2}\lambda^2\right)\right)$$

2. solving the dual

$g(\lambda)$ is a 1D concave function. We can find its maximum by finding $\lambda^* \geq 0$ where the derivative $g'(\lambda) = 0$. By $g'(\lambda) = \frac{\partial L(x^*(\lambda),\lambda)}{\partial \lambda} = \|x^*(\lambda)\|_1 - 1$.

$$g'(\lambda) = \|S_\lambda(a)\|_1 - 1 = \sum_{i=1}^{n} \max(0, |a_i| - \lambda) - 1$$

Let $h(\lambda) = g'(\lambda)$. $h(\lambda)$ is a continuous, monotonically decreasing function. We need to find $\lambda^* \geq 0$ such that $h(\lambda^*) = 0$.

Algorithm:

1. Check $h(0) = \sum \max(0, |a_i|) - 1 = \|a\|_1 - 1$.
2. If $h(0) \leq 0$ (i.e., $\|a\|_1 \leq 1$), the point is already in the ball. The gradient is negative (or zero) and decreasing, so the max is at the boundary. The solution is $\lambda^* = 0$.
3. If $h(0) > 0$ (i.e., $\|a\|_1 > 1$), the point is outside. Since $h(0) > 0$ and $h(\lambda) \to -1$ as $\lambda \to \infty$, there is a unique root $\lambda^* > 0$. We must solve $h(\lambda^*) = 0$:

$$\sum_{i=1}^{n} \max(0, |a_i| - \lambda^*) = 1$$

   This is a 1D root finding problem for a monotonically decreasing function. It can be solved using a **bisection search** for λ.

Bisection Search: This is an efficient root-finding algorithm.

- We have an interval $[L, R] = [0, \max |a_i|]$.
- We know $h(L) = h(0) > 0$ (positive) and $h(R) = h(\max |a_i|) \approx -1$ (negative).
- The algorithm repeatedly cuts the interval in half, always keeping the half where the function's sign changes, until it converges to the root λ^* where $h(\lambda^*) = 0$.

3. the Optimal x^*

From the KKT conditions, the primal optimal x^* and dual optimal λ^* must satisfy the stationarity condition. This condition states that x^* must minimize the lagrangian $L(x, \lambda)$ when $\lambda = \lambda^*$.

$$x^* = \arg\min_x L(x, \lambda^*) = \arg\min_x \left(\frac{1}{2}\|x - a\|_2^2 + \lambda^* \|x\|_1 \right)$$

This is the exact same separable minimization subproblem we solved in Step 2 of the dual derivation to find $g(\lambda)$.

In that step, we found that the general formula for the minimizer $x(\lambda)$ is given by the soft-thresholding operator:

$$x_i(\lambda) = S_\lambda(a_i) = \text{sign}(a_i) \max(0, |a_i| - \lambda)$$

Therefore, to find the optimal x^*, we simply take the optimal λ^* (found by our efficient bisection search) and put it into this formula:

$$x_i^* = S_{\lambda^*}(a_i) = \text{sign}(a_i) \max(0, |a_i| - \lambda^*)$$

Q3) A convex problem in which strong duality fails.

(a) **Verify that this is a convex optimization problem. Find the optimal value.**

checking convexity

- objective function: $f_0(x, y) = e^{-x}$

$$\nabla^2 f_0(x, y) = \begin{bmatrix} e^{-x} & 0 \\ 0 & 0 \end{bmatrix} \succeq 0 \text{ (since } e^{-x} > 0\text{). } f_0 \text{ is convex.}$$

- constraint function: $f_1(x, y) = x^2/y$ on $\mathcal{D} = \{(x, y) \mid y > 0\}$

  To check for convexity, we compute the Hessian matrix:

$$\nabla^2 f_1(x, y) = H = \begin{bmatrix} 2/y & -2x/y^2 \\ -2x/y^2 & 2x^2/y^3 \end{bmatrix}$$

  We can test for positive semidefiniteness using Sylvester's criterion (checking principal minors).

  - The first principal minor (top-left element) is $M_1 = 2/y$. Since $(x, y) \in \mathcal{D}$, we have $y > 0$, so $M_1 = 2/y > 0$.
  - The other first principal minor (bottom-right) is $M_2 = 2x^2/y^3$. Since $y > 0$ and $x^2 \geq 0$, we have $M_2 \geq 0$.
  - The second principal minor (the determinant) is:

$$\det(H) = \left(\frac{2}{y}\right)\left(\frac{2x^2}{y^3}\right) - \left(\frac{-2x}{y^2}\right)^2$$

$$\det(H) = \frac{4x^2}{y^4} - \frac{4x^2}{y^4} = 0$$

  Since all principal minors are non-negative ($M_1 \geq 0$, $M_2 \geq 0$, $\det(H) \geq 0$), the Hessian H is positive semidefinite. Therefore, $f_1(x, y)$ is convex.

- domain: $\mathcal{D} = \{(x, y) \mid y > 0\}$ is an open half-plane, which is a convex set.

The problem is a convex optimization problem.

optimal value (p^*)

- feasible set: Constraint $x^2/y \leq 0$ with domain $y > 0$. Since $y > 0$ and $x^2 \geq 0$, the constraint is only satisfied if $x^2 = 0 \implies x = 0$. Feasible set $\mathcal{F} = \{(x, y) \mid x = 0, y > 0\}$.
- primal problem:

$$p^* = \inf_{(x,y)\in\mathcal{F}} f_0(x, y) = \inf_{x=0,y>0} e^{-x} = e^{-0} = 1$$

$$p^* = 1$$

(b) **Give the Lagrange dual problem, and find the optimal solution λ^* and optimal value d^* of the dual problem. What is the optimal duality gap?**

lagrangian and dual function

For $\lambda \geq 0$:

$$L(x, y, \lambda) = f_0(x, y) + \lambda f_1(x, y) = e^{-x} + \lambda \frac{x^2}{y}$$

dual function:

$$g(\lambda) = \inf_{(x,y)\in\mathcal{D}} L(x, y, \lambda) = \inf_{y>0,x} \left(e^{-x} + \frac{\lambda x^2}{y} \right)$$

- case $\lambda = 0$:

$$g(0) = \inf_{y>0,x} (e^{-x}) = \lim_{x\to\infty} e^{-x} = 0$$

- case $\lambda > 0$: We are looking for the infimum of $L(x, y, \lambda) = e^{-x} + \lambda x^2/y$. In the domain $\mathcal{D}$, we have $y > 0$. Also $\lambda > 0$, $x^2 \geq 0$, and $e^{-x} > 0$. This means both terms are non-negative, so $L(x, y, \lambda) \geq 0$ for all $(x, y) \in \mathcal{D}$. Therefore the infimum $g(\lambda)$ must be $g(\lambda) \geq 0$.

  Now we check if we can make L arbitrarily close to 0. We need to find a sequence of points $(x_k, y_k) \in \mathcal{D}$ such that $L(x_k, y_k, \lambda) \to 0$ as $k \to \infty$. This requires both $e^{-x_k} \to 0$ and $\lambda x_k^2/y_k \to 0$.

  - To make $e^{-x_k} \to 0$, we must have $x_k \to \infty$. Let's pick $x_k = k$.
  - Now we need to make the second term $\lambda k^2/y_k \to 0$.
  - We must choose y_k such that $y_k > 0$ and $y_k \to \infty$ faster than k^2.
  - Let's choose $y_k = k^3$. This satisfies $y_k > 0$ (for $k \geq 1$) and $k^2/y_k = k^2/k^3 = 1/k$.

  We now test this sequence of points $(x_k, y_k) = (k, k^3)$. This path is in the domain $\mathcal{D}$ for $k \geq 1$.

$$\lim_{k\to\infty} L(k, k^3, \lambda) = \lim_{k\to\infty} \left(e^{-k} + \frac{\lambda k^2}{k^3} \right) = \lim_{k\to\infty} \left(e^{-k} + \frac{\lambda}{k} \right)$$

  As $k \to \infty$, $e^{-k} \to 0$ and $\lambda/k \to 0$.

$$\lim_{k\to\infty} L(k, k^3, \lambda) = 0 + 0 = 0$$

  Since we found a path in the domain $\mathcal{D}$ along which $L \to 0$, and we already know $L \geq 0$ everywhere, the infimum $g(\lambda)$ must be 0.

Therefore, $g(\lambda) = 0$ for all $\lambda \geq 0$.

dual problem

$$\begin{array}{rl} \max & g(\lambda) \\ \text{s.t.} & \lambda \geq 0 \end{array}$$

This is $\max_{\lambda \geq 0} 0$.

dual optimal solution and value

$$d^* = 0$$

Any $\lambda^* \geq 0$ is an optimal solution.

optimal duality gap

$$\text{gap} = p^* - d^* = 1 - 0 = 1$$

$$\text{duality gap} = 1$$

(c) **Does Slater's condition hold for this problem?**

- Slater's condition requires the existence of a point $(\tilde{x}, \tilde{y})$ in the relative interior of the domain $\mathcal{D}$ such that all inequality constraints are strictly satisfied, $f_1(\tilde{x}, \tilde{y}) < 0$.
- The relative interior of a set $\mathcal{D}$ is its interior relative to the smallest flat set (affine hull) containing it.
- For a full dimensional set in $\mathbb{R}^n$ (one that isn't just a line or plane), the relative interior is the same as the standard interior.
- An open set, like our domain $\mathcal{D} = \{(x, y) \mid y > 0\}$, is full-dimensional in $\mathbb{R}^2$ and consists only of interior points.
- Therefore, the relative interior of $\mathcal{D}$ is the set $\mathcal{D}$ itself
- So, we must find a point $(x, y) \in \mathcal{D}$ such that $f_1(x, y) = x^2/y < 0$.
- On the domain $\mathcal{D}$, we are given $y > 0$. We also know $x^2 \geq 0$ for any real x.
- This implies $f_1(x, y) = x^2/y \geq 0$ for all $(x, y) \in \mathcal{D}$.
- It is impossible to find a point where $x^2/y < 0$. The strictly feasible set is empty.

No, Slater's condition does not hold.

Q4) Consider a convex optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le 0, \quad i = 1, \dots, m, \\ & Ax = b, \end{array}$$

with variable $x \in \mathbb{R}^n$, that satisfies Slater's constraint qualification. Determine whether each of the statements below is true or false. True means it holds with no further assumptions.

(a) **The primal and dual problems have the same objective value.**

- answer: true
- reasoning: The problem is given as convex, and it is given that Slater's condition holds. Slater's condition implies that strong duality holds. Strong duality means the primal optimal value (p^*) is equal to the dual optimal value (d^*).

$$p^* = d^*$$

(b) **The primal problem has a unique solution.**

- answer: false
- reasoning: Strong duality does not guarantee a unique primal solution. The problem only states $f_0(x)$ is convex, not strictly convex. For example, if $f_0(x) = c$ (a constant is convex), and the feasible set has more than one point, every feasible point is an optimal solution.

(c) **The dual problem is not unbounded.**

- answer: true
- reasoning: Slater's condition implies the primal problem is feasible, so p^* is either finite or $-\infty$. The dual problem is a maximization problem, and unbounded would mean $d^* \to +\infty$. By strong duality (from part a), we have $d^* = p^*$. Since p^* cannot be $+\infty$, d^* cannot be $+\infty$. Therefore, the dual problem is not unbounded.

(d) **Suppose x^* is optimal, with $f_1(x^*) = -0.2$. Then for every dual optimal point (λ^*, ν^*), we have $\lambda_1^* = 0$.**

- answer: true
- reasoning: Since the problem is convex and Slater's condition holds, the KKT conditions are necessary. Any primal optimal x^* and dual optimal (λ^*, ν^*) must satisfy the complementary slackness condition:

  $$\lambda_i^* f_i(x^*) = 0 \quad \text{for } i = 1, \dots, m$$

  We are given the values for $i = 1$:

  $$f_1(x^*) = -0.2$$

Putting this into the complementary slackness equation:

$$\lambda_1^* \cdot (-0.2) = 0$$

The only solution to this equation is $\lambda_1^* = 0$. This must hold for every dual optimal point.

Q5) The problem

$$\begin{array}{ll} \text{minimize} & -3x_1^2 + x_2^2 + 2x_3^2 + 2(x_1 + x_2 + x_3) \\ \text{subject to} & x_1^2 + x_2^2 + x_3^2 = 1, \end{array}$$

is nonconvex but still strong duality holds. In fact, a more general result holds: strong duality holds for any optimization problem with a quadratic objective and one quadratic inequality constraint, provided Slater's condition holds. Derive the KKT conditions. Find all solutions x, ν that satisfy the KKT conditions. Which pair corresponds to the optimum?

1. lagrangian

The objective and constraint are:

$$f_0(x) = -3x_1^2 + x_2^2 + 2x_3^2 + 2x_1 + 2x_2 + 2x_3$$

$$h_1(x) = x_1^2 + x_2^2 + x_3^2 - 1 = 0$$

The lagrangian $L(x, \nu) = f_0(x) + \nu h_1(x)$ is:

$$L(x, \nu) = (-3x_1^2 + x_2^2 + 2x_3^2 + 2x_1 + 2x_2 + 2x_3) + \nu(x_1^2 + x_2^2 + x_3^2 - 1)$$
$$L(x, \nu) = (\nu - 3)x_1^2 + 2x_1 + (\nu + 1)x_2^2 + 2x_2 + (\nu + 2)x_3^2 + 2x_3 - \nu$$

2. KKT conditions

The KKT conditions are stationarity ($\nabla_x L = 0$) and primal feasibility ($h_1(x) = 0$).

1. **stationarity:** $\nabla_x L(x, \nu) = 0$

$$\text{(a)}\ \frac{\partial L}{\partial x_1} = 2(\nu - 3)x_1 + 2 = 0 \implies x_1 = \frac{-1}{\nu - 3} = \frac{1}{3 - \nu}$$
$$\text{(b)}\ \frac{\partial L}{\partial x_2} = 2(\nu + 1)x_2 + 2 = 0 \implies x_2 = \frac{-1}{\nu + 1}$$
$$\text{(c)}\ \frac{\partial L}{\partial x_3} = 2(\nu + 2)x_3 + 2 = 0 \implies x_3 = \frac{-1}{\nu + 2}$$

(These equations imply $\nu \notin \{3, -1, -2\}$).

2. **primal feasibility:**

$$\text{(d)}\ x_1^2 + x_2^2 + x_3^2 = 1$$

3. all solutions (x, ν)

We substitute the stationarity results (a, b, c) into the primal feasibility condition (d):

$$\left(\frac{1}{3 - \nu}\right)^2 + \left(\frac{-1}{\nu + 1}\right)^2 + \left(\frac{-1}{\nu + 2}\right)^2 = 1$$

$$\frac{1}{(3-\nu)^2} + \frac{1}{(\nu+1)^2} + \frac{1}{(\nu+2)^2} = 1$$

Any real root ν_k of this equation, paired with the corresponding $x^{(k)} = (x_1(\nu_k), x_2(\nu_k), x_3(\nu_k))$, is a solution (x, ν) that satisfies the KKT conditions.

4. which pair corresponds to the optimum value ?

To find the minimum, we check the second-order sufficient condition. A KKT point is a local minimum if the hessian of the lagrangian, $\nabla_x^2 L(x, \nu)$, is positive definite.

$$\nabla_x^2 L(x, \nu) = \begin{bmatrix} 2(\nu-3) & 0 & 0 \\ 0 & 2(\nu+1) & 0 \\ 0 & 0 & 2(\nu+2) \end{bmatrix}$$

For this diagonal matrix to be positive definite, all its diagonal entries must be positive:

- $2(\nu-3) > 0 \implies \nu > 3$
- $2(\nu+1) > 0 \implies \nu > -1$
- $2(\nu+2) > 0 \implies \nu > -2$

All three conditions are only satisfied if $\nu > 3$.

Let's analyze the function $g(\nu) = \frac{1}{(3-\nu)^2} + \frac{1}{(\nu+1)^2} + \frac{1}{(\nu+2)^2}$ on the interval $(3, \infty)$.

- As ν approaches 3 from the right, $g(\nu) \to \infty$ (because $\frac{1}{(3-\nu)^2} \to \infty$).
- As ν approaches ∞, $g(\nu) \to 0 + 0 + 0 = 0$.

Since $g(\nu)$ is continuous on $(3, \infty)$ and goes from ∞ to 0, by the Intermediate Value Theorem, there must be exactly one root $\nu^* > 3$ that satisfies $g(\nu) = 1$.

The other roots of $g(\nu) = 1$ will correspond to $\nu < 3$, which do not satisfy the second-order sufficient condition for a minimum.

The pair corresponding to the optimum (minimum) is the unique solution (x^*, ν^*) for which $\nu^* > 3$.

Q6) Consider the function $f(x_1, x_2) = (x_1 + x_2^2)^2$. At the point $x^T = (1, 0)$ we consider the search direction $p^T = (-1, 1)$. Show that p is a descent direction and find all minimizers of the following problem at each iteration t:

$$\min_{\alpha>0} f(x^t + \alpha p^t)$$

(quick note: we will use x and p for x^t and p^t as given in the first part).

Part 1: p is a descent direction

A direction p is a descent direction for f at x if the directional derivative is negative: $\nabla f(x)^T p < 0$.

1. **the gradient of $f(x)$:**

$$\begin{aligned} f(x_1, x_2) &= (x_1 + x_2^2)^2 \\ \frac{\partial f}{\partial x_1} &= 2(x_1 + x_2^2) \cdot 1 = 2(x_1 + x_2^2) \\ \frac{\partial f}{\partial x_2} &= 2(x_1 + x_2^2) \cdot (2x_2) = 4x_2(x_1 + x_2^2) \\ \nabla f(x) &= \begin{bmatrix} 2(x_1 + x_2^2) \\ 4x_2(x_1 + x_2^2) \end{bmatrix} \end{aligned}$$

2. **evaluating the gradient at $x = (1, 0)$:**

$$\nabla f(1, 0) = \begin{bmatrix} 2(1 + 0^2) \\ 4(0)(1 + 0^2) \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

3. **the dot product $\nabla f(x)^T p$:** Given $p = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$.

$$\nabla f(x)^T p = \begin{bmatrix} 2 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = (2)(-1) + (0)(1) = -2$$

Since $\nabla f(x)^T p = -2 < 0$, p is a descent direction.

Part 2: all minimizers of $\min_{\alpha>0} f(x + \alpha p)$

What is a Line Search? This problem, $\min_{\alpha>0} f(x + \alpha p)$, is called a line search. The idea is: We are at a point $x = (1, 0)$ and know $p = (-1, 1)$ is a good direction to move in. But we don't know *how far* to move.

To find the perfect step size, we define a new, one-dimensional function $g(\alpha)$ which is a slice of our original function f along the line defined by x and p.

$$g(\alpha) = f(x + \alpha p)$$

Our goal is to find the $\alpha > 0$ that minimizes this simple 1D function. We use standard calculus: find the derivative $g'(\alpha)$ and set it to zero. The resulting α is our optimal step size.

solving the Line Search

1. **the new function $g(\alpha)$:** Find the point $x + \alpha p$:

$$x + \alpha p = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \alpha \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1-\alpha \\ \alpha \end{bmatrix}$$

2. **writing $g(\alpha)$ by substituting this point into f:** Let $x_1 = 1 - \alpha$ and $x_2 = \alpha$.

$$g(\alpha) = f(1-\alpha, \alpha) = ((1-\alpha) + (\alpha)^2)^2$$
$$g(\alpha) = (\alpha^2 - \alpha + 1)^2$$

3. **minimizing $g(\alpha)$:** Take the derivative with respect to α and set to 0.

$$g'(\alpha) = 2(\alpha^2 - \alpha + 1) \cdot (2\alpha - 1)$$

4. **the critical points by setting $g'(\alpha) = 0$:**

$$2(\alpha^2 - \alpha + 1)(2\alpha - 1) = 0$$

   This gives two possibilities:

   - $\alpha^2 - \alpha + 1 = 0$: The discriminant is $\Delta = (-1)^2 - 4(1)(1) = -3$. Since $\Delta < 0$, this part has no real roots.
   - $2\alpha - 1 = 0$: This gives the solution $\alpha = 1/2$.

5. **verifing the minimizer:** The only critical point is $\alpha = 1/2$, which satisfies $\alpha > 0$. We check the second derivative:

$$g''(\alpha) = \frac{d}{d\alpha}[2(\alpha^2 - \alpha + 1)(2\alpha - 1)] = 12\alpha^2 - 12\alpha + 6$$

   At $\alpha = 1/2$:

$$g''(1/2) = 12(1/2)^2 - 12(1/2) + 6 = 12(1/4) - 6 + 6 = 3$$

   Since $g''(1/2) = 3 > 0$, the point $\alpha = 1/2$ is a local minimum.

Since this is the only critical point, it is the unique global minimizer.

The minimizer is $\alpha = 1/2$.

Q7) Simulations: Consider the problem of minimizing a piece-wise linear function:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{s.t.} & c^T x \leq d \end{array}$$

which $f(x) = \max_{i=1,\dots,m}(a_i^T x + b_i)$ with the variable $x \in \mathbf{R}^n$ and the constants $a_i \in \mathbf{R}^n$ and $b_i, d \in \mathbf{R}$. Solve this problem with the projected subgradient method for a specific problem instance where a_i and b_i, c and d are generated from a unit normal distribution and plot the objective function value in the following setting.

(a) **Constant step-size rule.**

(b) **Constant step length rule.**

(c) **Non-summable, diminishing step size rule $\alpha_k = \frac{0.1}{\sqrt{k}}$.**

(d) **Square summable but not summable step rule $\alpha_k = \frac{1}{k}$.**

Discuss and compare convergence for the above rules.

Part 1: problem setup

To solve this, I first set up a specific problem instance in Python. I chose a dimension of $n = 10$ and $m = 20$ affine functions. The data A (with rows a_i^T), b, c, and d were all generated using np.random.randn.

The most important step was to find the true optimal value, f^*. We can't know if our algorithm is converging unless we know what it's supposed to converge *to*.

This problem is a convex optimization problem, and it can be perfectly reformulated as a Linear Program (LP):

$$\begin{array}{lll} \text{minimize} & t & \\ \text{s.t.} & a_i^T x + b_i \leq t, & \text{for } i = 1, ..., m \\ & c^T x \leq d & \end{array}$$

I used the ‘cvxpy‘ library to solve this LP exactly. This gave me the ground truth optimal value f^*.

Part 2: the projected subgradient algorithm

The algorithm follows a simple loop for $k = 0, 1, \dots$:

1. **finding a subgradient g_k:** For our function $f(x) = \max_i(a_i^T x + b_i)$, a subgradient g_k is simply the a_j corresponding to the function that wins the max at x_k.

$$j = \text{argmax}_i(a_i^T x_k + b_i) \quad \Longrightarrow \quad g_k = a_j$$

2. **taking a step:** We calculate a temporary new point y_{k+1} using our step size α_k.

$$y_{k+1} = x_k - \alpha_k g_k$$

3. **projecting onto the feasible set:** The point y_{k+1} might not be feasible (i.e., it might violate $c^T y \leq d$). We must project it back onto the feasible set $C = \{x \mid c^T x \leq d\}$ to get our final x_{k+1}.

   The projection $x_{k+1} = P_C(y_{k+1})$ is calculated:

$$x_{k+1} = \begin{cases} y_{k+1} & \text{if } c^T y_{k+1} \leq d \\ y_{k+1} - \frac{c^T y_{k+1} - d}{\|c\|^2} c & \text{if } c^T y_{k+1} > d \end{cases}$$

I implemented this algorithm and ran it for all four step-size rules for 50,000 iterations.

Part 3: simulation results

numerical results After running the simulation for 50,000 iterations, I printed the final objective value $f(x_k)$ for each rule and compared it to the true optimal value f^*. The Gap is $f(x_k) - f^*$.

```
========================================
Final Results after 50000 iterations
True Optimal Value (f_star): 1.04052588
========================================
(a) Constant Step-Size:
 Final f(x): 1.20997680   (Gap: 0.16945092)
(b) Constant Step-Length:
 Final f(x): 1.17929996   (Gap: 0.13877408)
(c) Non-Summable (0.1/sqrt(k)):
 Final f(x): 1.10211266   (Gap: 0.06158679)
(d) Square-Summable (1/k):
 Final f(x): 1.10969210   (Gap: 0.06916622)
========================================
```

Convergence Plot The following plots show the objective value $f(x_k)$ (top) and the optimality gap $f(x_k) - f^*$ on a log scale (bottom) versus the iteration number k.

Part 5: Comparison

The results from the table and the plot clearly show the theoretical properties of the subgradient method.

- **Rules (a) and (b) :** These rules, as expected, do not converge to the true optimal solution. The plot shows that their objective values decrease quickly and then bounce around in a neighborhood above f^*. The final numerical results confirm this: they are stuck with large optimality gaps (0.169 and 0.139). They will never get any closer, no matter how many more iterations we run.

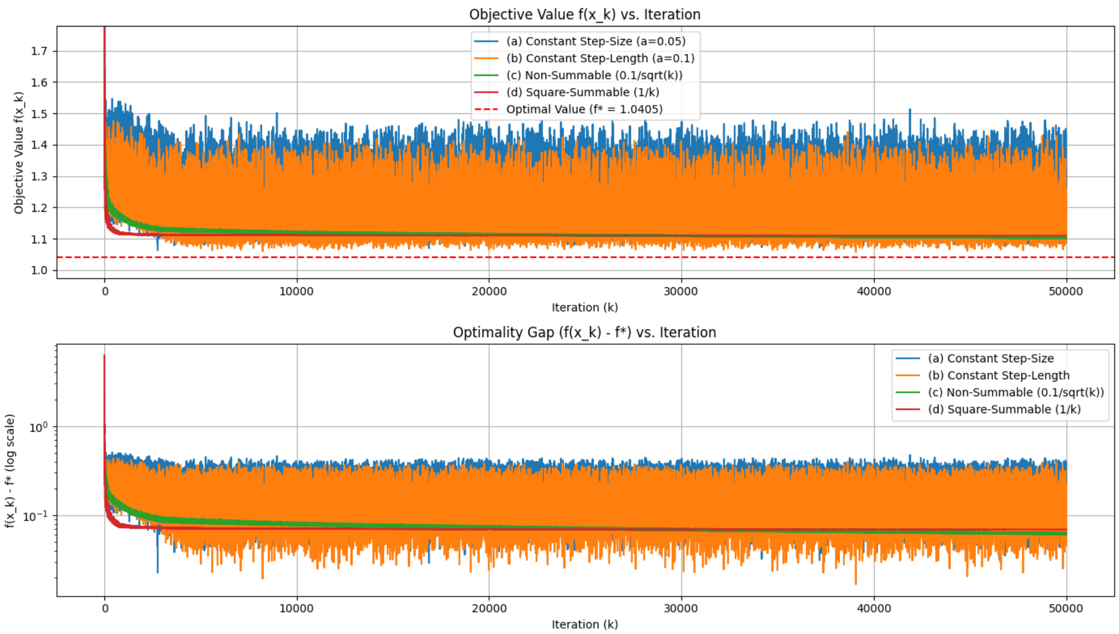

Figure 1: Convergence of the four step-size rules.

- **Rules (c) and (d) :** These rules, $\alpha_k = 0.1/\sqrt{k}$ and $\alpha_k = 1/k$, both satisfy the conditions for convergence ($\sum \alpha_k = \infty$ and $\alpha_k \to 0$). The results clearly show this. Their final gaps (0.062 and 0.069) are less than half the size of the gaps from the constant-step rules. This is a significant difference and proves they are working correctly.
- **On the Slow Convergence:** At first, I was surprised that the gaps for (c) and (d) were still so large (e.g., 0.06) after 50,000 iterations. I expected them to be closer to zero. However, this is the main practical lesson of the subgradient method: it is very slow. Its convergence rate is $O(1/\sqrt{k})$, which means that to get 10 times more accuracy (reduce the gap from 0.06 to 0.006), we would need 100 times more iterations ($50,000 \times 100 = 5,000,000$ iterations).

Q8) Consider the following functions:

$$f_1(x) = \exp(x-3) - \frac{\pi}{2}x - 2$$

$$f_2(x) = -(1+\frac{1}{2^x})\cos(x)\sin(x), \quad -20 \leq x \leq 20$$

$$f_3(x) = 100(x_2 - x_1^2)^2 + (1-x_1)^2$$

For each of these functions, perform the following tasks:

(a) **Compute the gradient and Hessian.**

(b) **Minimize the function with steepest descent method, Newton's method, and a combination of both methods while using backtracking line search.**

(c) **Report the function value, distance to the optimal point, and step-size in three plots for these three algorithms. Explain the results.**

Part 1: gradient and hessian

(a) $f_1(x) = \exp(x-3) - \frac{\pi}{2}x - 2$

gradient $(f_1'(x))$**:**

$$\nabla f_1(x) = \exp(x-3) - \frac{\pi}{2}$$

hessian $(f_1''(x))$**:**

$$\nabla^2 f_1(x) = \exp(x-3)$$

—

(b) $f_2(x) = -(1+\frac{1}{2^x})\cos(x)\sin(x)$

First we simplify the function using $\sin(x)\cos(x) = \frac{1}{2}\sin(2x)$ and $\frac{1}{2^x} = 2^{-x}$.*

$$f_2(x) = -\frac{1}{2}(1+2^{-x})\sin(2x)$$

gradient $(f_2'(x))$**:**

$$\nabla f_2(x) = \frac{\ln(2)}{2}2^{-x}\sin(2x) - (1+2^{-x})\cos(2x)$$

hessian $(f_2''(x))$**:**

$$\nabla^2 f_2(x) = \left(2 + \left(2 - \frac{(\ln 2)^2}{2}\right)2^{-x}\right)\sin(2x) + (2\ln 2)2^{-x}\cos(2x)$$

—

(c) $f_3(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

gradient ($\nabla f_3(x)$)**:**

$$\nabla f_3(x) = \begin{bmatrix} -400x_1(x_2 - x_1^2) - 2(1 - x_1) \\ 200(x_2 - x_1^2) \end{bmatrix} = \begin{bmatrix} 400x_1^3 - 400x_1x_2 + 2x_1 - 2 \\ 200x_2 - 200x_1^2 \end{bmatrix}$$

hessian ($\nabla^2 f_3(x)$)**:**

$$\nabla^2 f_3(x) = \begin{bmatrix} 1200x_1^2 - 400x_2 + 2 & -400x_1 \\ -400x_1 & 200 \end{bmatrix}$$

Part 2: simulation results

(a) $f_1(x)$ Results

explanation of results: The plots show a perfect case for a strongly convex function.

- **steepest descent** shows a straight line on the log-plot, which is the definition of linear convergence. It is slow but steady.
- **Newton's method & combined method** are identical and show a vertical drop, converging in only 5-6 iterations. This is quadratic convergence. They are identical because the Hessian $f_1''(x) = \exp(x - 3)$ is always positive, so the Combined method always defaults to using the Newton step.
- **step size:** The step size alpha for Newton's method becomes 1.0 after the first iteration, showing the full Newton step is accepted, which is key to its speed.

(b) $f_2(x)$ Results

explanation of results: The algorithms fail to converge to the global minimum, and the plots show them diverging or getting stuck. There are two primary reasons for this failure:

- **non convexity:** $f_2(x)$ is a highly non-convex "wavy" function with many different local minima. Optimization algorithms are only guaranteed to find a local minimum, not the global one. Depending on the start, they can converge to different minima, as seen when Steepest Descent and Newton's method move towards different valleys.
- **numerical instability:** The term $1/2^x$ (or 2^{-x}) in the function definition causes the function value and its derivatives to become very large for negative x (e.g., 2^{20}). This causes a numerical overflow, where the gradient explodes to inf. This breaks the algorithms, causing them to either diverge (as seen in the Distance plot) or get stuck taking tiny, useless steps.

(c) $f_3(x)$ Results

Explanation of Results: These plots perfectly illustrate the weakness of steepest descent and the power of Newton's method.

- **steepest descent** (blue line) fails, showing extremely slow linear convergence. It takes thousands of iterations to make minimal progress. This is due to the fact that the algorithm gets stuck on the steep walls of the narrow valley, and its tiny steps fail to make progress down the valley floor.
- **Newton's Method & combined method** (orange/green lines) show a vertical drop, achieving quadratic convergence in only 15-20 iterations. They are identical because the hessian is positive definite in the valley.
- **step size:** Newton's method uses a step size of alpha = 1.0, showing it is using the full, smart Newton step (which accounts for the valley's curvature) to jump directly toward the minimum.

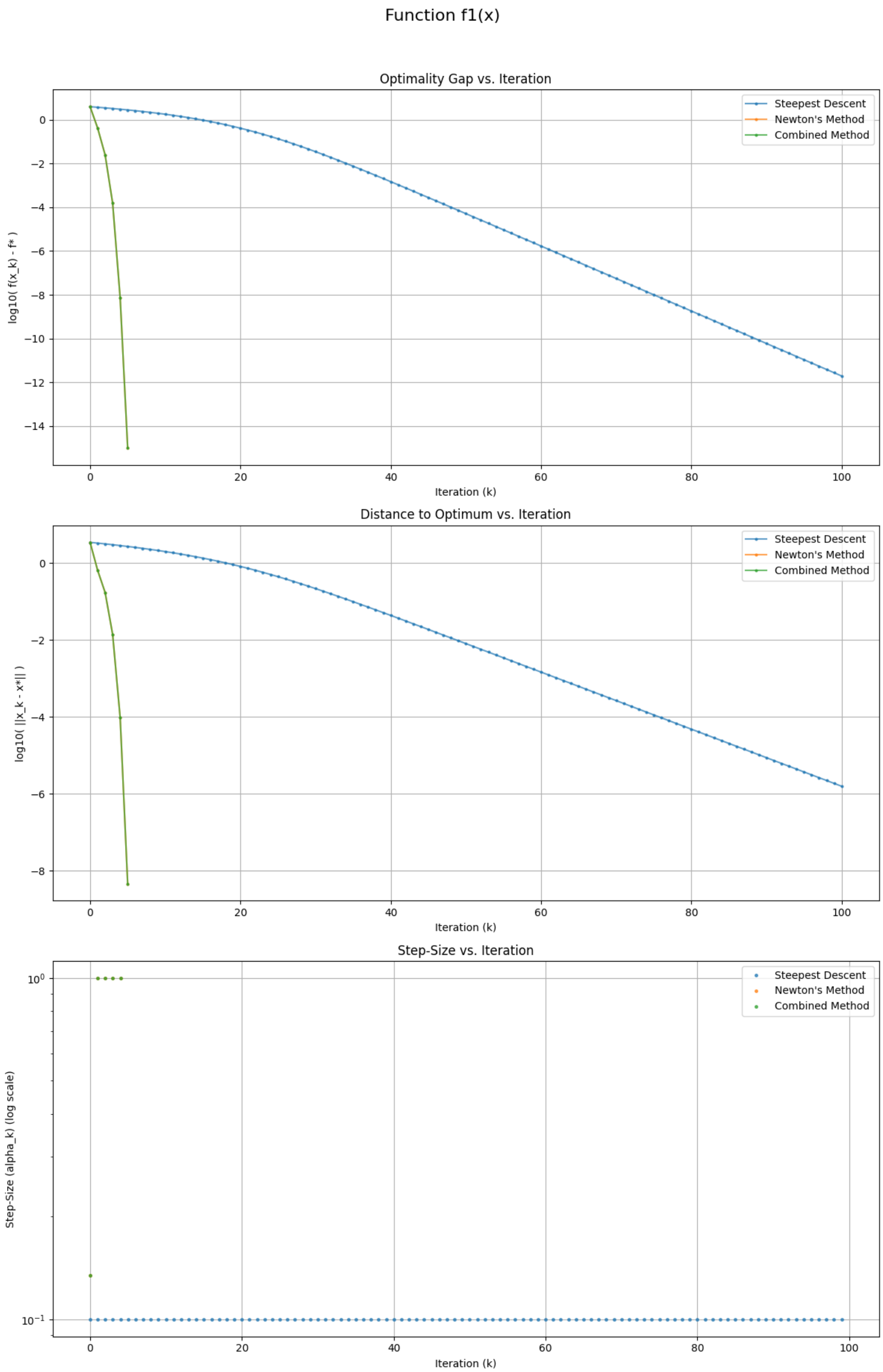

Figure 2: Convergence plots for the convex function $f_1(x)$.

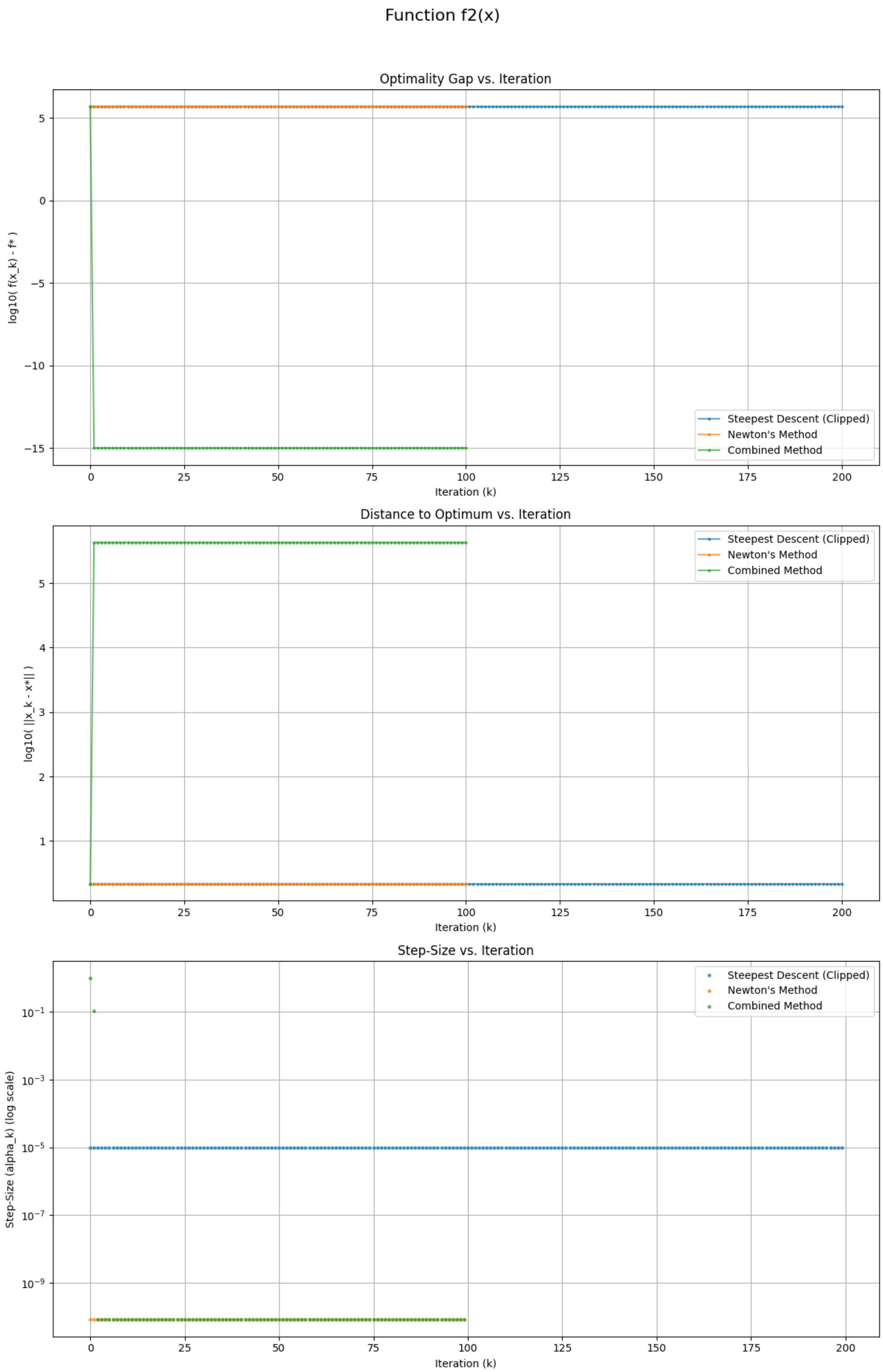

Figure 3: Convergence plots for the non-convex function $f_2(x)$.

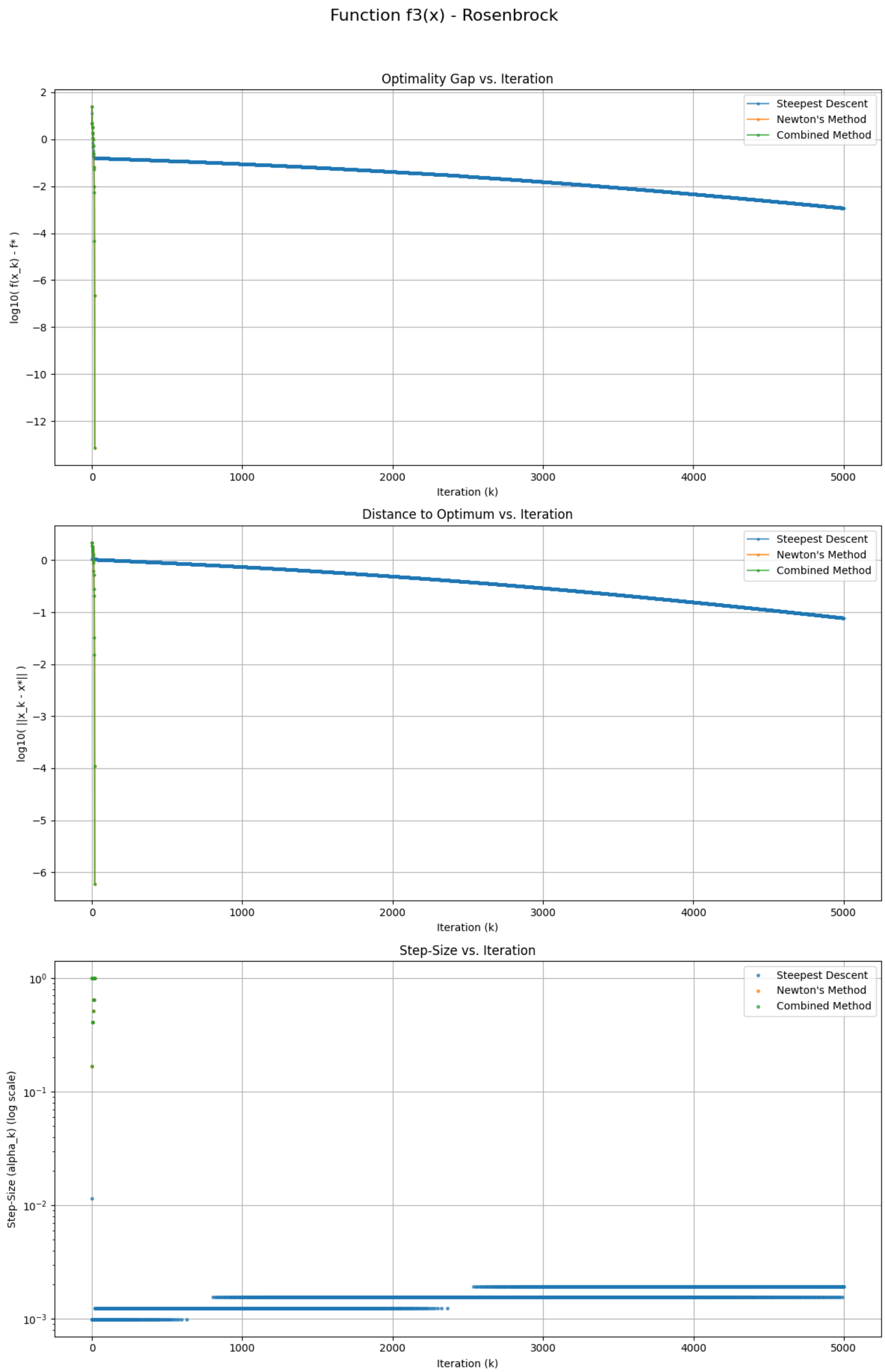

Figure 4: Convergence plots for the function $f_3(x)$.