Midterm

Hyuntaek Oh

`ohhyun@oregonstate.edu`

Due: May 21, 2025

ECE599/ AI539 Nonlinear Optimization (Spring 2025)
Midterm
(Due: 11:59pm on May 21, Wednesday.)

Instruction: Students should provide enough detail of the logical procedure of deriving answers. Answers without sufficient justification will receive partial or no credit. For questions involving MATLAB experiments, provide codes with comments. You are allowed to use **Python** instead of MATLAB. The maximum possible score is 100.

1. **Convergence of Gradient Descent with Armijo's Rule: Convex Optimization [30 points]**.

   Suppose that f is in C^2, i.e., twice continuously differentiable, and there exist $A, a > 0$ such that for every $\mathbf{x} \in \mathbb{R}^n$,

$$a \leq \text{all eigenvalues of } \mathbf{F}(\mathbf{x}) \leq A, \tag{1}$$

   where $\mathbf{F}(\mathbf{x})$ denotes, as usual, the Hessian of f at $\mathbf{x}$. This implies that f is a strictly convex function.

   Consider the unconstrained optimization problem:

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \tag{2}$$

   In this question, we will analyze convergence of the Gradient Descent algorithm employing Armijo's rule as the line search method.

   Recall the Armijo's rule with parameters $\epsilon \in (0, 1/2)$ and $\eta > 1$: in the k-th iteration, we choose the step size $\alpha_k \geq 0$ such that it satisfies

$$\begin{array}{c}\text{(i) } f(x_k - \alpha_k \mathbf{g}(\mathbf{x}_k)) \leq f(\mathbf{x}_k) - \epsilon \cdot \alpha_k \mathbf{g}(\mathbf{x}_k)^T \mathbf{g}(\mathbf{x}_k), \quad \text{and} \\ \text{(ii) } f(x_k - \eta\alpha_k \mathbf{g}(\mathbf{x}_k)) > f(\mathbf{x}_k) - \epsilon \cdot \eta\alpha_k \mathbf{g}(\mathbf{x}_k)^T \mathbf{g}(\mathbf{x}_k)\end{array} \tag{3}$$

   where $\mathbf{g}(\mathbf{x}_k)$ denotes $\nabla f(\mathbf{x}_k)^T$. Now, let $\{\mathbf{x}_k\}$ denote the sequence generated by Gradient Descent with Armijo's rule. Solve the following questions related to convergence of this sequence.

   (Pages 240 and 241 of the textbook contain the convergence analysis of Gradient Descent with Exact Line Search and Gradient Descent with Armijo's rule. I strongly recommend you to ***read those two pages*** before you start working on this question. You will be able to get some hints. Note, however, that you are required to provide full derivations in your answers with enough justification, even for those results that were derived in pages 240-241)

(a) Prove that when $0 \leq \alpha \leq 1/A$, $f(\mathbf{x}_k - \alpha \mathbf{g}(\mathbf{x}_k)) \leq f(\mathbf{x}_k) - \epsilon\alpha||\mathbf{g}(\mathbf{x}_k)||^2$.

According to the textbook page 240-241, given a point $\mathbf{x}_k$ for any α, exact line search is:

$$f(\mathbf{x}_k - \alpha \mathbf{g}(\mathbf{x}_k)) \leq f(\mathbf{x}_k) - \alpha \mathbf{g}(\mathbf{x}_k)^T \mathbf{g}(\mathbf{x}_k) + \frac{A\alpha^2}{2}||\mathbf{g}(\mathbf{x}_k)||^2.$$

The inequality above is derived from second-order Taylor expansion with a Hessian upper bound. The right side in the inequality would be compared to the upper bound of Armijo's rule (i):

$$f(\mathbf{x}_k) - \alpha \mathbf{g}(\mathbf{x}_k)^T \mathbf{g}(\mathbf{x}_k) + \frac{A\alpha^2}{2}\mathbf{g}(\mathbf{x}_k)^T \mathbf{g}(\mathbf{x}_k) \leq f(\mathbf{x}_k) - \epsilon\alpha \mathbf{g}(\mathbf{x}_k)^T \mathbf{g}(\mathbf{x}_k)$$

where $\mathbf{g}(\mathbf{x}_k)^T \mathbf{g}(\mathbf{x}_k) = ||\mathbf{g}(\mathbf{x}_k)||^2$.
Simplifying the inequality above:

$$-\alpha + \frac{A\alpha^2}{2} \leq -\epsilon\alpha$$

$$-1 + \frac{A\alpha}{2} \leq -\epsilon \Rightarrow \alpha \leq \frac{2(1-\epsilon)}{A}$$

The upper bound of the condition $0 \leq \alpha < 1/A$ can be compared to the inequality above with respect to α, meaning that:

$$\frac{1}{A} \leq \frac{2(1-\epsilon)}{A}$$

This implies:

$$\alpha \in \left[0, \frac{1}{A}\right] \subseteq \left[0, \frac{2(1-\epsilon)}{A}\right]$$

So as long as $\epsilon \in (0, \frac{1}{2})$, then $\frac{1}{A} \leq \frac{2(1-\epsilon)}{A}$, and thus the inequality is satisfied.
Therefore, for any $\alpha \in [0, \frac{1}{A}]$, the inequality below is satisfied:

$$f(\mathbf{x}_k - \alpha \mathbf{g}(\mathbf{x}_k)) \leq f(\mathbf{x}_k) - \epsilon\alpha||\mathbf{g}(\mathbf{x}_k)||^2 \text{ for any } \epsilon \in (0, \frac{1}{2})$$

(b) Using the result of part (a), prove that $\eta\alpha_k \geq \frac{1}{A}$.

As described, the result from (a) is used for contradiction. To achieve Armijo's sufficient decrease condition, by applying $\eta\alpha_k$ to second-order Taylor expansion from (a), it would be:

$$f(x_k - \eta\alpha_k \mathbf{g}(\mathbf{x}_k)) \leq f(\mathbf{x}_k) - \epsilon \cdot \eta\alpha_k \mathbf{g}(\mathbf{x}_k)^T \mathbf{g}(\mathbf{x}_k) + \frac{A(\eta\alpha)^2}{2}||\mathbf{g}(\mathbf{x}_k)||^2$$

$\eta\alpha_k$, where $\eta > 1$, does not satisfy the Armijo's condition since a step size $\eta\alpha_k$ is too large a move. Suppose Armijo's condtion fails at $\eta\alpha_k$, so:

$$f(x_k - \eta\alpha_k \mathbf{g}(\mathbf{x}_k)) > f(\mathbf{x}_k) - \epsilon \cdot \eta\alpha_k \mathbf{g}(\mathbf{x}_k)^T \mathbf{g}(\mathbf{x}_k)$$

Then, it compared to the upper bound from Taylor:

$$-\epsilon\eta\alpha_k + \frac{A(\eta\alpha_k)^2}{2} > -\epsilon\eta\alpha_k$$

This means:

$$\frac{A(\eta\alpha_k)^2}{2} > 0$$

By comparing Armijo's rule (ii) with Taylor expansion:

$$f(\mathbf{x}_k) - \epsilon \cdot \eta\alpha_k \mathbf{g}(\mathbf{x}_k)^T \mathbf{g}(\mathbf{x}_k) \leq f(\mathbf{x}_k) - \epsilon \cdot \eta\alpha_k \mathbf{g}(\mathbf{x}_k)^T \mathbf{g}(\mathbf{x}_k) + \frac{A(\eta\alpha)^2}{2} ||\mathbf{g}(\mathbf{x}_k)||^2$$

By simplifying:

$$-\epsilon\eta \leq -\eta + \frac{A\alpha_k\eta^2}{2}$$

$$-\epsilon \leq -1 + \frac{A\alpha_k\eta}{2}$$

$$\frac{2(1-\epsilon)}{A} \leq \eta\alpha_k$$

Since $\epsilon < \frac{1}{2} \Rightarrow 2(1-\epsilon) > 1$, this implies:

$$\frac{1}{A} \leq \eta\alpha_k$$

Thus, $\eta\alpha_k \geq \frac{1}{A}$ holds.

(c) Using the result of part (b) and the fact that α_k satisfies (3), prove that

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \frac{\epsilon}{\eta A} ||\mathbf{g}(\mathbf{x}_k)||^2, \tag{4}$$

or equivalently,

$$f(\mathbf{x}_{k+1}) - f^* \leq f(\mathbf{x}_k) - f^* - \frac{\epsilon}{\eta A} ||\mathbf{g}(\mathbf{x}_k)||^2. \tag{5}$$

where $f^* := \min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$.

The result of part (b) can be modified:

$$\eta\alpha_k \geq \frac{1}{A} \Rightarrow \alpha_k \geq \frac{1}{\eta A}$$

From the Armijo's rule condition, the inequality is:

$$f(\mathbf{x}_k - \alpha\mathbf{g}(\mathbf{x}_k)) \leq f(\mathbf{x}_k) - \epsilon\alpha||\mathbf{g}(\mathbf{x}_k)||^2$$

The left side of the Armijo's condition means:

$$f(\mathbf{x}_k - \alpha\mathbf{g}(\mathbf{x}_k)) = f(\mathbf{x}_{k+1})$$

Then, by applying the result of part (b) to the Armijo's rule condition to obtain a uniform lower bound on the amount of function decrease:

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \epsilon\alpha||\mathbf{g}(\mathbf{x}_k)||^2 \leq f(\mathbf{x}_k) - \frac{\epsilon}{\eta A}||\mathbf{g}(\mathbf{x}_k)||^2$$

The inequalities above show that the bound $\alpha_k \geq \frac{1}{\eta A}$ is used as the worst-case (least) decrease when α_k is exactly $\frac{1}{\eta A}$.
According to the textbook and (b), the second part of the stopping criterion states that $\eta\alpha$ does not satisfy the first criterion and thus the final α must satisfy $\alpha \geq \frac{1}{\eta A}$. There fore, the inequality of the first part of the criterion implies:

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \frac{\epsilon}{\eta A}||\mathbf{g}(\mathbf{x}_k)||^2$$

The inequality above means the function value at the new point is strictly smaller than the previous one, minus some guaranteed drop.
Subtracting f^* from both sides,

$$f(\mathbf{x}_{k+1}) - f^* \leq f(\mathbf{x}_k) - f^* - \frac{\epsilon}{\eta A}||\mathbf{g}(\mathbf{x}_k)||^2$$

As noted, f^* is a minimum point, so the error terms of left and right side indicate how far the point is from optimality at step k and $k+1$.
Thus, $f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \frac{\epsilon}{\eta A}||\mathbf{g}(\mathbf{x}_k)||^2$ and $f(\mathbf{x}_{k+1}) - f^* \leq f(\mathbf{x}_k) - f^* - \frac{\epsilon}{\eta A}||\mathbf{g}(\mathbf{x}_k)||^2$ hold.

(d) Prove that

$$-||\mathbf{g}(\mathbf{x}_k)||^2 \leq 2a(f^* - f(\mathbf{x}_k)). \tag{6}$$

There exists $V \in [x_k, x^*]$ such that

$$f(x^*) = f(x_k) + g(x_k)^T(x^* - x_k) + \frac{1}{2}(x^* - x_k)^T \boldsymbol{F}(V)(x^* - x_k)$$

$$f^* - f(x_k) = g(x_k)^T(x^* - x_k) + \frac{1}{2}(x^* - x_k)^T \boldsymbol{F}(V)(x^* - x_k)$$

This is based on the second-order Taylor expansion and strong convexity property $\nabla^2 f(x) \geq aI$.
By the Mean Value Theorem, the last term would be:

$$\frac{1}{2}a||x^* - x_k||^2 \leq \frac{1}{2}(x^* - x_k)^T \boldsymbol{F}(V)(x^* - x_k) \leq \frac{1}{2}A||x^* - x_k||^2$$

As described in the textbook, if f is strongly convex with constant $a > 0$, then for all $x, y \in \mathbb{R}^n$, this inequality holds:

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{a}{2}||y - x||^2$$

This is a quadratic lower bound on the function, and it could be applied by optimal point:

$$f(\mathbf{x}) \geq f(x_k) + \nabla f(x_k)^T(\mathbf{x} - x_k) + \frac{a}{2}||\mathbf{x} - x_k||^2$$

Then, a quadratic approximation (lower bound) of $f(y)$ can be defined as:

$$\phi(y) := f(x_k) + \nabla f(x_k)^T(y - x_k) + \frac{a}{2}||y - x_k||^2$$

To minimize the equation above, taking gradient of $\phi(y)$ with respect to y:

$$\nabla_y \phi(y) = \nabla f(x_k) + a(y - x_k)$$

When setting derivative to zero:

$$\nabla f(x_k) + a(y - x_k) = 0 \Rightarrow y = x_k - \frac{1}{a}\nabla f(x_k)$$

$$\rightarrow \bar{x} = x_k - \frac{1}{a}\nabla f(x_k)$$

By plugging into $\phi(\bar{x})$:

$$\phi(\bar{x}) = f(x_k) + \nabla f(x_k)^T(-\frac{1}{a}\nabla f(x_k)) + \frac{a}{2}||\frac{1}{a}\nabla f(x_k)||^2$$

Simplifying:

$$\phi(\bar{x}) = f(x_k) - \frac{1}{a}||g(x_k)||^2 + \frac{1}{2a}||g(x_k)||^2 = f(x_k) - \frac{1}{2a}||g(x_k)||^2$$

Therefore, since $f^* \geq \phi(\bar{x})$, it would be:

$$f^* \geq f(x_k) - \frac{1}{2a}||g(x_k)||^2$$

The inequality above can be modified:

$$f^* \geq f(x_k) - \frac{1}{2a}||g(x_k)||^2$$
$$\Rightarrow f^* - f(x_k) \geq -\frac{1}{2a}||g(x_k)||^2$$
$$\Rightarrow 2a(f^* - f(x_k)) \geq -||g(x_k)||^2$$

Thus, $-||\mathbf{g}(\mathbf{x}_k)||^2 \leq 2a(f^* - f(\mathbf{x}_k))$ holds.

(e) Combine the above results to prove that

$$f(\mathbf{x}_{k+1}) - f^* \leq \left(1 - \frac{2\epsilon a}{\eta A}\right)(f(\mathbf{x}_k) - f^*). \tag{7}$$

Based on the above inequality, discuss the convergence of the Gradient Descent algorithm with Armijo's rule when applied to solve the convex optimization problem.

The results from (c) and (d) is combined to prove (e):

$$(c) : f(\mathbf{x}_{k+1}) - f^* \leq f(\mathbf{x}_k) - f^* - \frac{\epsilon}{\eta A}||\mathbf{g}(\mathbf{x}_k)||^2$$

$$(d) : -||\mathbf{g}(\mathbf{x}_k)||^2 \leq 2a(f^* - f(\mathbf{x}_k)) \rightarrow ||\mathbf{g}(\mathbf{x}_k)||^2 \geq 2a(f(\mathbf{x}_k) - f^*)$$
$$\Rightarrow f(\mathbf{x}_{k+1}) - f^* \leq f(\mathbf{x}_k) - f^* - \frac{\epsilon}{\eta A}2a(f(\mathbf{x}_k) - f^*)$$

Rewrite the inequality above:

$$\Rightarrow f(\mathbf{x}_{k+1}) - f^* \leq (f(\mathbf{x}_k) - f^*) - \frac{2\epsilon a}{\eta A}(f(\mathbf{x}_k) - f^*)$$

$$\Rightarrow f(\mathbf{x}_{k+1}) - f^* \leq (1 - \frac{2\epsilon a}{\eta A})(f(\mathbf{x}_k) - f^*)$$

This is identical to part (e).

The gradient descent algorithm with Armijo's rule shows global and linear convergence when applied to minimize a strongly convex function. The inequality implies that the sub-optimal $f(x_k) - f^*$ decreases by a fixed ratio $1 - \frac{2\epsilon a}{\eta A}$ at every iteration. The rate depends on the strong convexity parameter a, the constant A of the gradient, and the Armijo parameters $\epsilon \in (0, 1/2)$ and $\eta > 1$. A larger a, smaller A, larger ϵ, or smaller η results in faster convergence. This demonstrates that even with an inexact line search strategy, gradient descent using Armijo's rule guarantees efficiency for solving strongly convex unconstrained optimization problems.

2. **Gauss-Southwell Method [35 points]**. Consider an unconstrained optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$$

where the objective function f is continuously differentiable. A *coordinate descent method* is an iterative descent method where in each iteration, we select one variable and minimize the objective function f over this single variable while fixing all other variables to their values from the previous iteration. Specifically, a coordinate descent method has the following structure:

- Initialize: Set an initial point $\mathbf{x}_0$, and let $k = 0$.
- Iteration: Until the stopping criterion is satisfied, repeat the following steps.
  - *Coordinate Selection*: Determine the coordinate i_k ($i_k \in \{1, 2, ..., n\}$) according to a pre-determined criterion.
  - *Bidirectional Line search*: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \cdot \mathbf{e}_{i_k}$ where $\alpha_k \in \mathbb{R}$ satisfies

$$f(\mathbf{x}_k + \alpha_k \cdot \mathbf{e}_{i_k}) = \min_{\alpha\in\mathbb{R}} f(\mathbf{x}_k + \alpha \cdot \mathbf{e}_{i_k}) \tag{8}$$

  - Increase k by 1.
- Return $\mathbf{x}_k$

In the above, $\mathbf{e}_j$ denotes the vector in $\mathbb{R}^n$ with its j-th entry equal to 1 while all other entries are zeros. Note that in the line search step, we search for an optimal step size in $\mathbb{R}$, i.e., we are not restricting α_k to be nonnegative. Adding $\alpha_k \cdot \mathbf{e}_{i_k}$ to $\mathbf{x}_k$ only changes the i_k-th entry of $\mathbf{x}_k$, so the bidirectional line search step corresponds to minimizing $f(\mathbf{x})$ over $x(i_k)$ alone (which denotes the i_k-the entry of $\mathbf{x}$), while fixing all other entries of $\mathbf{x}$ to be the same as that of $\mathbf{x}_k$. See Section 8.9 of the textbook for more explanation on coordinate descent methods.

In some optimization problems, optimizing f along a single coordinate while fixing all other coordinates turns out to be a simple task (sometimes, a closed-form solution can be obtained.) In such cases, the coordinate descent method is popularly used due to the simplicity of the line search step. There are various types of coordinate descent methods employing different criteria in choosing the coordinate i_k to update in the k-th iteration. For instance, the simplest type is the cyclic coordinate descent method wherein i_k cycles through all coordinates, i.e., the sequence $\{i_k\}$ is

$$1, 2, ..., n, 1, 2, ..., n, 1, 2, ..., n, 1, 2, ..., n, ...$$

In this question, we will analyze a coordinate descent method called *Gauss-Southwell* method wherein the coordinate to update in the k-th iteration is chosen based on the gradient $\mathbf{g}(\mathbf{x}_k) :=$

$\nabla f(\mathbf{x}_k)^T$. Specifically, in the k-th iteration, Gauss-Southwell method sets i_k to be a maximizer of

$$\max_j |g_j(\mathbf{x}_k)| \tag{9}$$

where $g_j(\mathbf{x}_k)$ denotes the j-th entry of the gradient $\mathbf{g}(\mathbf{x}_k)$. In other words, in Gauss-Southwell method, we set i_k to the coordinate at which the magnitude of the gradient entry is maximum. In the rest of this problem, we will focus our attention to a quadratic optimization problem where the objective function is

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} - \mathbf{b}^T\mathbf{x} \tag{10}$$

with $\mathbf{Q}$ being a positive definite matrix. Let A and a denote the largest and smallest eigenvalues of $\mathbf{Q}$, respectively. We will analyze convergence of Gauss-Southwell method when applied to solve this quadratic optimization problem.

(a) Let $\mathbf{g}_k$ denote $\mathbf{g}(\mathbf{x}_k)$. And, let $\bar{g}_k$ denote the vector in $\mathbb{R}^n$ with its i_k-th entry equal to the i_k-th entry of $\mathbf{g}_k$ while all other entries are zeros. Show that the update in the k-th iteration of Gauss-Southwell method is equivalent to the following: $\mathbf{x}_{k+1} = \mathbf{x}_k - \bar{\alpha}_k \cdot \bar{\mathbf{g}}_k$ where $\bar{\alpha}_k$ is a nonnegative step size satisfying $\mathbf{x}_k - \bar{\alpha}_k \cdot \bar{\mathbf{g}}_k = \min_{\alpha \geq 0} \mathbf{x}_k - \alpha \cdot \bar{\mathbf{g}}_k$.

Suppose that $\bar{g}_k$:

$$\bar{g}_k = g_{i_k}(x_k)e_{i_k}$$

The new update rule is:

$$x_{k+1} = x_k - \alpha\bar{g}_k = x_k - \alpha g_{i_k}(x_k)e_{i_k}$$

Then, defining $x(\alpha)$ and plugging into $f(x)$:

$$x(\alpha) = x_k - \alpha g_{i_k}(x_k)e_{i_k}$$

$$f(x(\alpha)) = \frac{1}{2}x(\alpha)^T Q x(\alpha) - b^T x(\alpha)$$

This can expand:

$$f(x(\alpha)) = \frac{1}{2}(x_k - \alpha g_{i_k}(x_k)e_{i_k})^T Q(x_k - \alpha g_{i_k}(x_k)e_{i_k}) - b^T(x_k - \alpha g_{i_k}(x_k)e_{i_k})$$

This form is a scalar-valued function of α, and since f is quadratic, this is a quadratic function of α:

$$f(x(\alpha)) = f(x_k) - \alpha \cdot g_{i_k}(x_k) \cdot (Qx_k - b)_{i_k} + \frac{1}{2}\alpha^2 \cdot (g_{i_k}(x_k))^2 \cdot (e_{i_k}^T Q e_{i_k})$$

Note that the term $(Qx_k - b)_{i_k} = g_{i_k}(x_k)$. So, the equation above would be

$$f(x(\alpha)) = f(x_k) - \alpha(g_{i_k}(x_k))^2 + \frac{1}{2}\alpha^2 \cdot (g_{i_k}(x_k))^2 \cdot (e_{i_k}^T Q e_{i_k})$$

, where $(g_{i_k}(x_k))^2 > 0$ and $(g_{i_k}(x_k))^2 \cdot (e_{i_k}^T Q e_{i_k}) > 0$.
This is a convex quadratic in α, minimized at:

$$\bar{\alpha}_k = \frac{(g_{i_k}(x_k))^2}{(g_{i_k}(x_k))^2 \cdot Q_{i_k i_k}} = \frac{1}{(e_{i_k}^T Q e_{i_k})}$$

Since Q is positive definite and $Q > 0$, the term $(e_{i_k}^T Q e_{i_k}) > 0$. Moreover, since $\bar{\alpha}_k > 0$, $\bar{\alpha}_k \in [0, \infty)$. So, this is the minimizer over $\alpha \geq 0$ (unidirectional), and it is also the minimizer over $\alpha \in \mathbb{R}$ (bidirectional). This means:

$$arg \min_{\alpha \geq 0} f(x_k - \alpha \bar{g}_k) = arg \min_{\alpha \in \mathbb{R}} f(x_k - \alpha \bar{g}_k)$$

As a result, the update rule:

$$x_{k+1} = x_k - \bar{\alpha}_k \bar{g}_k, \quad \bar{\alpha}_k = arg \min_{\alpha \geq 0} f(x_k - \alpha \bar{g}_k)$$

This is based on the fact that the descent direction $\bar{g}_k = g_{i_k}(x_k) e_{i_k}$, and the minimizer lies in $\alpha > 0$.

Thus, we can say that the update in the k-th iteration of Gauss-Southwell method is equivalent to the following: $\mathbf{x}_{k+1} = \mathbf{x}_k - \bar{\alpha}_k \cdot \bar{\mathbf{g}}_k$ where $\bar{\alpha}_k$ is a nonnegative step size satisfying $\mathbf{x}_k - \bar{\alpha}_k \cdot \bar{\mathbf{g}}_k = \min_{\alpha \geq 0} \mathbf{x}_k - \alpha \cdot \bar{\mathbf{g}}_k$.

(b) Exercise 19 in Chapter 8 of the textbook is about proving the following fact: if the direction $\mathbf{d}_k$ of an iterative descent method with *exact line search* satisfies (i) $\mathbf{d}_k^T g_k < 0$, and (ii) $(\mathbf{d}_k^T g_k)^2 \geq \beta(\mathbf{d}_k^T \mathbf{d}_k)(g_k^T g_k)$ with $\beta \in (0, 1]$, then

$$\mathbf{E}(\mathbf{x}_{k+1}) \leq \Big(1 - \beta \frac{a}{A}\Big)\mathbf{E}(\mathbf{x}_k) \tag{11}$$

where $\mathbf{E}(\mathbf{x}) := \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)\mathbf{Q}(\mathbf{x} - \mathbf{x}^*)$ ($\mathbf{x}^*$ denotes the unique global minimum point). You can use this result without proving it.

Apply the above result to prove that the sequence $\{\mathbf{x}_k\}$ generated by Gauss-Southwell method satisfies the following inequality:

$$\mathbf{E}(\mathbf{x}_{k+1}) \leq \Big(1 - \frac{a}{nA}\Big)\mathbf{E}(\mathbf{x}_k) \tag{12}$$

(Hint: in part (a), we showed that the k-th iteration of Gauss-Southwell method is equivalent to that of an iterative descent method with exact line search using the direction $\mathbf{d}_k = -\bar{g}_k$.)
Based on this inequality, discuss about global and local convergence of Gauss-Southwell method when applied to the quadratic optimization.

From the hint in part (a), $d_k = -\bar{g}_k$ is applied to condition (i) and (ii). They are:

$$d_k^T g_k = -g_{i_k}(x_k)^2$$

$$d_k^T d_k = g_{i_k}(x_k)^2$$

$$g_k^T g_k = \sum_j^n g_j(x_k)^2$$

By using these, the left and right sides of condition (ii) would be:

$$\textit{left:} \quad (d_k^T g_k)^2 = g_{i_k}(x_k)^4$$

$$\textit{right:} \quad (d_k^T d_k)(g_k^T g_k) = g_{i_k}(x_k)^2 \cdot \sum_j^n g_j(x_k)^2$$

Then, the condition (ii) can be divided by $(d_k^T d_k)(g_k^T g_k)$ to obtain β:

$$\frac{(\mathbf{d}_k^T g_k)^2}{(\mathbf{d}_k^T \mathbf{d}_k)(g_k^T g_k)} \geq \beta \Rightarrow \frac{g_{i_k}(x_k)^4}{g_{i_k}(x_k)^2 \cdot \sum_j^n g_j(x_k)^2} = \frac{g_{i_k}(x_k)^2}{\sum_j^n g_j(x_k)^2} \geq \beta$$

This means the ratio of the left side affects β, which is in the right side of the inequality. The numerator is just one variable (Gauss-Southwell method chooses i_k to maximize $|g_j(j)|$), whereas denominator is from one to n, so the ratio of the left side is:

$$\beta \leq \frac{g_{i_k}(x_k)^2}{\sum_j^n g_j(x_k)^2} = \frac{1}{n}$$

This implies that $\beta = \frac{1}{n}$. Then, it apply to given inequality:

$$\boldsymbol{E}(\mathbf{x}_{k+1}) \leq \left(1 - \beta\frac{a}{A}\right)\boldsymbol{E}(\mathbf{x}_k) \Rightarrow \beta = \frac{1}{n} \Rightarrow \boldsymbol{E}(\mathbf{x}_{k+1}) \leq \left(1 - \frac{a}{nA}\right)\boldsymbol{E}(\mathbf{x}_k)$$

Thus, it holds.
$\boldsymbol{E}(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)\boldsymbol{Q}(\mathbf{x} - \mathbf{x}^*) = f(x_k) - f^* > 0$ if f^* is a minimum point. Moreover, the recurrence:

$$\boldsymbol{E}(\mathbf{x}_{k+1}) \leq \left(1 - \frac{a}{nA}\right)\boldsymbol{E}(\mathbf{x}_k)$$

ensures decrease of the $E(x)$. Since $\left(1-\frac{a}{nA}\right) \in (0,1)$, it shrinks when k goes to ∞:

$$E(x_k) \leq \left(1-\frac{a}{nA}\right)^k E(x_0) \to 0 \text{ as } k \to \infty$$

Therefore, $x_k \to x^*$, the global minimizers, meaning that Gauss-Southwell method is globally convergent for strongly convex quadratic functions.

Additionally, as described above, since $\left(1-\frac{a}{nA}\right) \in (0,1)$, the local convergence of the Gauss-Southwell method on quadratic optimization problems is linear. This means that the rate of convergence does not get better or faster when we're close to the minimizer. It continues to shrink $E(x)$ by a constant fraction per step.

(c) (MATLAB experiment) Consider the unconstrained quadratic optimization with

$$\mathbf{Q} = \begin{bmatrix} 1 & 0.5 & 0 \\ 0.5 & 1 & 0.25 \\ 0 & 0.25 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix} \tag{13}$$

Implement Gauss-Southwell method to solve this problem. You can use the stopping criterion $||g(\mathbf{x}_k)|| \leq \epsilon$ with ϵ being a very small positive real number (e.g., 10^{-6}). Plot $||g(\mathbf{x}_k)||$ versus k. Plot $\mathbf{E}(\mathbf{x}_k)$ versus k. When plotting these curves, make the y-axis be in log-scale so that you can observe the order of convergence from the plots. Interpret the plots.

- *Plot $||g(\mathbf{x}_k)||$ versus k and Interpret the plot.*

  As can be seen in Figure 1, the plot of the gradient norm $||g(\mathbf{x}_k)||$ versus the iteration k on a logarithmic scale shows a straight-line pattern. This indicates that the gradient norm decreases exponentially with respect to the iteration number, which is evidence of linear convergence. In the context of the Gauss-Southwell method, this behavior is expected, as the method consistently reduces the most dominant component of the gradient in each iteration. The consistent decline without oscillation or flattening confirms that the method is effectively converging toward the optimal point.

- *Plot $\mathbf{E}(\mathbf{x}_k)$ versus k and Interpret the plot.*

  Figure 2 displays a linear trend on a log scale, supporting that the error decreases at a linear rate. This result matches the theoretical convergence rate of the Gauss-Southwell method for quadratic objectives, $E(x_{k+1}) \leq (1-\frac{a}{nA})E(x_k)$, where a and A are the smallest and largest eigenvalues of the matrix Q, respectively. The straight line nature of the plot in log-scale implies that the convergence order is linear, with

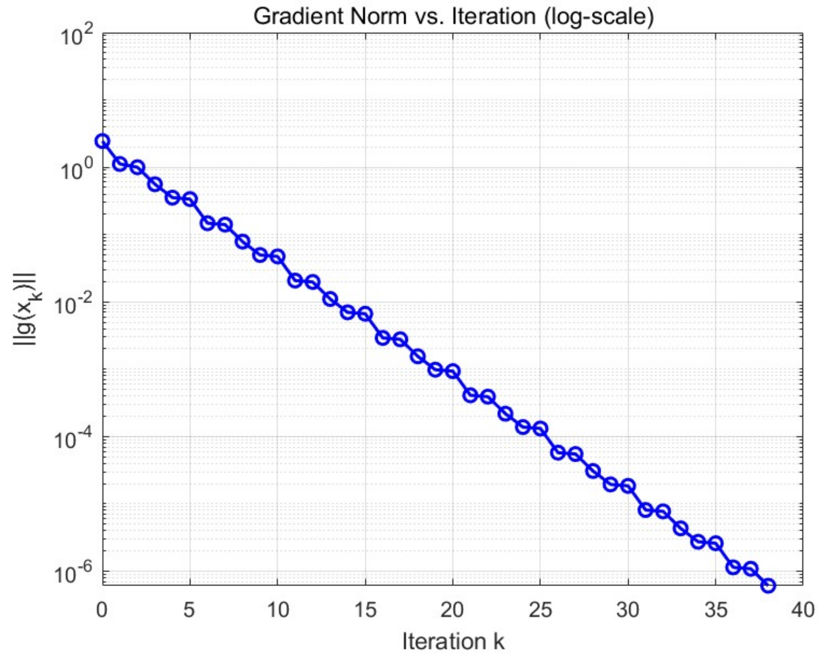

Figure 1: $||g(\mathbf{x}_k)||$ versus Iteration k

the convergence factor approximately matching the predicted value.

The order of convergence is to meet the condition based on p:

$$||x_{k+1} - x^*|| \leq C||x_k - x^*||^p$$

Taking log to both sides:

$$log(||x_{k+1} - x^*||) \leq log(C) + p \cdot log(||x_k - x^*||)$$

This implies that the graph would be linear shape if $p = 1$, meaning that the order of convergence is 1.
The decreasing ratio $(1 - \frac{a}{nA})$ is approximately 0.906:

$$1 - \frac{a}{nA} \approx 1 - \frac{0.441}{3 * 1.559} = 0.906$$

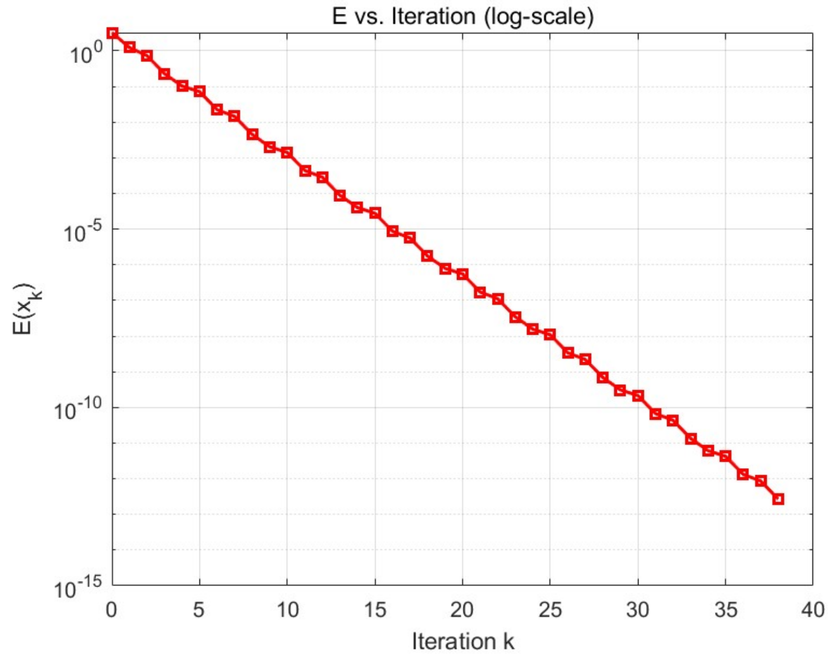

Figure 2: $\mathbf{E}(\mathbf{x}_k)$ versus Iteration k

3. **Application: Power System State Estimation [35 points]**. In controlling and monitoring a power grid, it is of utmost importance to have an accurate estimate of the system state. Consider the simple system with 2 buses in Figure 1. For this system, the system state is consisting of three real variables V_1, V_2, and θ_2, which are bus 1 voltage magnitude, bus 2 voltage magnitude, and bus 2 voltage phase angle respectively (the bus 1 voltage phase angle θ_1 is set to zero, i.e., the reference angle).

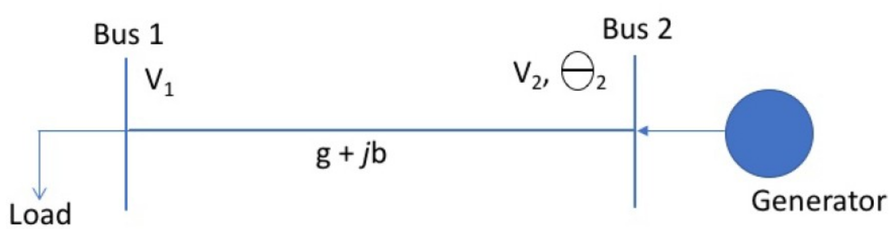

Figure 3: A power system consisting of twp buses

In practice, the system state is unknown and often not measured directly. To gain a good understanding of the system operating condition, we need to estimate the system state $(V_1, V_2, \theta_2) \in \mathbb{R}^3$ based on available measurements, and this procedure is called *state estimation*. Suppose that we have real and imaginary power flow measurements of the line 1-2 in both directions, which are related to the state as follows:

$$\begin{aligned} P_{12} &= g(V_1V_2cos\theta_2 - V_1^2) - bV_1V_2sin\theta_2 =: h_1(V_1, V_2, \theta_2) \\ Q_{12} &= -b(V_1V_2cos\theta_2 - V_1^2) - gV_1V_2sin\theta_2 =: h_2(V_1, V_2, \theta_2) \\ P_{21} &= g(V_2V_1cos\theta_2 - V_2^2) + bV_2V_1sin\theta_2 =: h_3(V_1, V_2, \theta_2) \\ Q_{21} &= -b(V_2V_1cos\theta_2 - V_2^2) + gV_2V_1sin\theta_2 =: h_4(V_1, V_2, \theta_2) \end{aligned} \tag{14}$$

where P_{12} and Q_{12} are real and imaginary parts, respectively, of measurement of the complex power flow from bus 1 to bus 2. Similarly, P_{21} and Q_{21} are real and imaginary parts, respectively, of measurement of the complex power flow from bus 2 to bus 1. In addition, g and b are conductance and susceptance of the line 1 - 2 respectively. In practice, we have measurements corrupted by additive random noise; however, in this question, we consider noiseless measurements for simplicity.

In order to estimate the state (V_1, V_2, θ_2) from the available measurement $(P_{12}, Q_{12}, P_{21}, Q_{21})$, a typical approach is to solve the unconstrained optimization problem

$$\min_{(V_1,V_2,\theta_2)\in\mathbb{R}^3} f(V_1, V_2, \theta_2) \tag{15}$$

where $f(V_1, V_2, \theta_2)$ is defined as the sum of the squared estimation residues:

$f(V_1, V_2, \theta_2)$

$:= (P_{12}-h_1(V_1, V_2, \theta_2))^2+(Q_{12}-h_2(V_1, V_2, \theta_2))^2+(P_{21}-h_3(V_1, V_2, \theta_2))^2+(Q_{21}-h_4(V_1, V_2, \theta_2))^2$

The solution to (15) is taken as the estimate of the system state. Since we are considering the noiseless case, our estimate is expected to be exactly the same as the true state.

Throughout this question, suppose that the line conductance and susceptance are as follows:

$$g = 1, \quad b = -5$$

(a) Obtain the expression of the gradient $\nabla f(V_1, V_2, \theta_2)^T$.

The expression of the gradient $\nabla f(V_1, V_2, \theta_2)^T$ would be:

$$\nabla f(V_1, V_2, \theta_2)^T = \begin{bmatrix} \frac{\partial f}{\partial V_1} \\ \frac{\partial f}{\partial V_2} \\ \frac{\partial f}{\partial \theta_2} \end{bmatrix}$$

The given function $f(V_1, V_2, \theta_2)$ is:

$(P_{12}-h_1(V_1,V_2,\theta_2))^2+(Q_{12}-h_2(V_1,V_2,\theta_2))^2+(P_{21}-h_3(V_1,V_2,\theta_2))^2+(Q_{21}-h_4(V_1,V_2,\theta_2))^2$

By simplifying this function to get a simple form of derivative :

$$f(V_1, V_2, \theta_2) = \sum_i^4 (a_i - h_i)^2, \text{ where } a_i = \{P_{12}, Q_{12}, P_{21}, Q_{21}\}$$

A simple form of derivative is:

$$\nabla f(V_1, V_2, \theta_2) = -2\sum_i^4 (a_i - h_i) \cdot \frac{\partial h_i}{\partial x}, \text{ where } x \in \{V_1, V_2, \theta_2\}$$

By using this, $\frac{\partial f}{\partial V_1}$, $\frac{\partial f}{\partial V_2}$, and $\frac{\partial f}{\partial \theta_2}$ are respectively:

$$\frac{\partial f}{\partial V_1} = -2\left[(P_{12} - h_1) \cdot \frac{\partial h_1}{\partial V_1} + (Q_{12} - h_2) \cdot \frac{\partial h_2}{\partial V_1} + (P_{21} - h_3) \cdot \frac{\partial h_3}{\partial V_1} + (Q_{21} - h_4) \cdot \frac{\partial h_4}{\partial V_1}\right]$$

$$\frac{\partial f}{\partial V_2} = -2\left[(P_{12} - h_1) \cdot \frac{\partial h_1}{\partial V_2} + (Q_{12} - h_2) \cdot \frac{\partial h_2}{\partial V_2} + (P_{21} - h_3) \cdot \frac{\partial h_3}{\partial V_2} + (Q_{21} - h_4) \cdot \frac{\partial h_4}{\partial V_2}\right]$$

$$\frac{\partial f}{\partial \theta_2} = -2\left[(P_{12} - h_1) \cdot \frac{\partial h_1}{\partial \theta_2} + (Q_{12} - h_2) \cdot \frac{\partial h_2}{\partial \theta_2} + (P_{21} - h_3) \cdot \frac{\partial h_3}{\partial \theta_2} + (Q_{21} - h_4) \cdot \frac{\partial h_4}{\partial \theta_2}\right]$$

Then, partial derivative of h_1, h_2, h_3, and h_4 are examined with respect to V_1, V_2, θ_2 given $g = 1$ and $b = -5$:

$$\frac{\partial h_1}{\partial V_1} = \frac{\partial(V_1V_2\cos\theta_2 - V_1^2 + 5V_1V_2\sin\theta_2)}{\partial V_1} = V_2\cos\theta_2 - 2V_1 + 5V_2\sin\theta_2$$

$$\frac{\partial h_2}{\partial V_1} = \frac{\partial(5V_1V_2\cos\theta_2 - 5V_1^2 - V_1V_2\sin\theta_2)}{\partial V_1} = 5V_2\cos\theta_2 - 10V_1 - V_2\sin\theta_2$$

$$\frac{\partial h_3}{\partial V_1} = \frac{\partial(V_2V_1\cos\theta_2 - V_2^2 - 5V_2V_1\sin\theta_2)}{\partial V_1} = V_2\cos\theta_2 - 5V_2\sin\theta_2$$

$$\frac{\partial h_4}{\partial V_1} = \frac{\partial(5V_2V_1\cos\theta_2 - V_2^2 + V_2V_1\sin\theta_2)}{\partial V_1} = 5V_2\cos\theta_2 + V_2\sin\theta_2$$

$$\frac{\partial h_1}{\partial V_2} = \frac{\partial(V_1V_2\cos\theta_2 - V_1^2 + 5V_1V_2\sin\theta_2)}{\partial V_2} = V_1\cos\theta_2 + 5V_1\sin\theta_2$$

$$\frac{\partial h_2}{\partial V_2} = \frac{\partial(5V_1V_2\cos\theta_2 - 5V_1^2 - V_1V_2\sin\theta_2)}{\partial V_2} = 5V_1\cos\theta_2 - V_1\sin\theta_2$$

$$\frac{\partial h_3}{\partial V_2} = \frac{\partial(V_2V_1\cos\theta_2 - V_2^2 - 5V_2V_1\sin\theta_2)}{\partial V_2} = V_1\cos\theta_2 - 2V_2 - 5V_1\sin\theta_2$$

$$\frac{\partial h_4}{\partial V_2} = \frac{\partial(5V_2V_1\cos\theta_2 - V_2^2 + V_2V_1\sin\theta_2)}{\partial V_2} = 5V_1\cos\theta_2 - 10V_2 + V_1\sin\theta_2$$

$$\frac{\partial h_1}{\partial \theta_2} = \frac{\partial(V_1V_2\cos\theta_2 - V_1^2 + 5V_1V_2\sin\theta_2)}{\partial \theta_2} = -V_1V_2\sin\theta_2 + 5V_1V_2\cos\theta_2$$

$$\frac{\partial h_2}{\partial \theta_2} = \frac{\partial(5V_1V_2\cos\theta_2 - 5V_1^2 - V_1V_2\sin\theta_2)}{\partial \theta_2} = -5V_1V_2\sin\theta_2 - V_1V_2\cos\theta_2$$

$$\frac{\partial h_3}{\partial \theta_2} = \frac{\partial(V_2V_1\cos\theta_2 - V_2^2 - 5V_2V_1\sin\theta_2)}{\partial \theta_2} = -V_2V_1\sin\theta_2 - 5V_2V_1\cos\theta_2$$

$$\frac{\partial h_4}{\partial \theta_2} = \frac{\partial(5V_2V_1\cos\theta_2 - V_2^2 + V_2V_1\sin\theta_2)}{\partial \theta_2} = -5V_2V_1\sin\theta_2 + V_2V_1\cos\theta_2$$

Finally, by applying these results to each partial derivative:

$$\frac{\partial f}{\partial V_1} = -2\begin{bmatrix} (P_{12} - h_1)\cdot(V_2\cos\theta_2 - 2V_1 + 5V_2\sin\theta_2) \\ +(Q_{12} - h_2)\cdot(5V_2\cos\theta_2 - 10V_1 - V_2\sin\theta_2) \\ +(P_{21} - h_3)\cdot(V_2\cos\theta_2 - 5V_2\sin\theta_2) \\ +(Q_{21} - h_4)\cdot(5V_2\cos\theta_2 + V_2\sin\theta_2) \end{bmatrix}$$

$$\frac{\partial f}{\partial V_2} = -2\begin{bmatrix} (P_{12} - h_1)\cdot(V_1\cos\theta_2 + 5V_1\sin\theta_2) \\ +(Q_{12} - h_2)\cdot(5V_1\cos\theta_2 - V_1\sin\theta_2) \\ +(P_{21} - h_3)\cdot(V_1\cos\theta_2 - 2V_2 - 5V_1\sin\theta_2) \\ +(Q_{21} - h_4)\cdot(5V_1\cos\theta_2 - 10V_2 + V_1\sin\theta_2) \end{bmatrix}$$

$$\frac{\partial f}{\partial \theta_2} = -2\begin{bmatrix} (P_{12} - h_1)\cdot(-V_1V_2\sin\theta_2 + 5V_1V_2\cos\theta_2) \\ +(Q_{12} - h_2)\cdot(-5V_1V_2\sin\theta_2 - V_1V_2\cos\theta_2) \\ +(P_{21} - h_3)\cdot(-V_2V_1\sin\theta_2 - 5V_2V_1\cos\theta_2) \\ +(Q_{21} - h_4)\cdot(-5V_2V_1\sin\theta_2 + V_2V_1\cos\theta_2) \end{bmatrix}$$

Thus:

$$\nabla f(V_1, V_2, \theta_2)^T = \begin{bmatrix} -2 \begin{bmatrix} (P_{12} - h_1) \cdot (V_2 \cos\theta_2 - 2V_1 + 5V_2 \sin\theta_2) \\ +(Q_{12} - h_2) \cdot (5V_2 \cos\theta_2 - 10V_1 - V_2 \sin\theta_2) \\ +(P_{21} - h_3) \cdot (V_2 \cos\theta_2 - 5V_2 \sin\theta_2) \\ +(Q_{21} - h_4) \cdot (5V_2 \cos\theta_2 + V_2 \sin\theta_2) \end{bmatrix} \\ -2 \begin{bmatrix} (P_{12} - h_1) \cdot (V_1 \cos\theta_2 + 5V_1 \sin\theta_2) \\ +(Q_{12} - h_2) \cdot (5V_1 \cos\theta_2 - V_1 \sin\theta_2) \\ +(P_{21} - h_3) \cdot (V_1 \cos\theta_2 - 2V_2 - 5V_1 \sin\theta_2) \\ +(Q_{21} - h_4) \cdot (5V_1 \cos\theta_2 - 10V_2 + V_1 \sin\theta_2) \end{bmatrix} \\ -2 \begin{bmatrix} (P_{12} - h_1) \cdot (-V_1 V_2 \sin\theta_2 + 5V_1 V_2 \cos\theta_2) \\ +(Q_{12} - h_2) \cdot (-5V_1 V_2 \sin\theta_2 - V_1 V_2 \cos\theta_2) \\ +(P_{21} - h_3) \cdot (-V_2 V_1 \sin\theta_2 - 5V_2 V_1 \cos\theta_2) \\ +(Q_{21} - h_4) \cdot (-5V_2 V_1 \sin\theta_2 + V_2 V_1 \cos\theta_2) \end{bmatrix} \end{bmatrix}$$

(b) In power systems, the system state (V_1, V_2, θ_2) is known to be close to (1, 1, 0) which is called "flat start". So, (1, 1, 0) is a good candidate as the initial point of iterative descent algorithms.
Suppose that the true state is

$$\mathbf{x}_{true} = (V_1, V_2, \theta_2) = (1.05, 0.98, 0.1).$$

Generate the measurements $(P_{12}, Q_{12}, P_{21}, Q_{21})$ for this state using the equations (14).
Based on the measurements you generated, implement a Gradient Descent algorithm to solve (15) and eventually obtain an estimate of $\mathbf{x}_{true}$. Use the flat start point as the initial point, i.e., $\mathbf{x}_0 = (1, 1, 0)$. For the line search method, use a backtracking method (explain your parameters). Let $\mathbf{x}_k$ denote the solution point obtained in the k-th iteration of the Gradient Descent. Plot $||\nabla f(\mathbf{x}_k)||$ versus k. Plot $||\mathbf{x}_k - \mathbf{x}_{true}||^2$ (i.e., estimation error) versus k. When plotting these curves, make the y-axis log-scale so that you can observe the order of convergence in the plot. Interpret the plots. Discuss global convergence and local convergence behaviors. What is the order of local convergence?

- *Parameter Initialization*

  As can be seen in Figure 4, the following parameters are used in the gradient descent algorithm with Backtracking line search method:

  - $x_0 = [1.0; 1.0; 0]$*: Initial point. This point is chosen such that Hessian is invertible, direction is descent, and Backtracking can proceed normally.*
  - *tol* $(= 1e^{-4} = 0.0001)$*: Convergence tolerance for gradient norm. The algorithm stops when the norm of the gradient is less than tolerance.*

- ϵ $(= 1e^{-6} = 0.000001)$: *Armijo sufficient decrease condition constant.*
- *max_iter* $(= 1000)$: *The maximum number of iterations.*
- α_{init} $(= 1.0)$: *initial step size in backtracking line search. It is essential for quadratic convergence near the optimum.*
- η $(= 2.0)$: *a step size reduction factor, and if the Armijo's condition is not satisfied, the current α is reduced by using $\alpha \leftarrow \alpha/\eta$.*

```
x_0 = [1.0; 1.0; 0];
x_k = x_0;
tol = 1e-4;
epsilon = 1e-6;
max_iter = 1000;
alpha_init = 1.0;
eta = 2.0;
```

Figure 4: Parameter Initialization

- *Plot $||\nabla f(x_k)||$ versus k and Interpret the plot.*

  As shown in Figure 5, the gradient norm plot, $||\nabla f(x_k)||$ versus iteration k, appears on a log scale. In the initial phase, the gradient norm decreases sharply, implying that the method quickly reaches a region near optimality. However, it then settles into a persistent oscillatory pattern rather than converging to zero monotonically. This is a well-known characteristic of coordinate descent: since only one component of x is updated at each iteration, the full gradient is not simultaneously minimized, resulting in non-smooth, zigzagging behavior in gradient norm. The overall low gradient means that the iterates are staying close to stationary points, and the oscillation amplitude is bounded, suggesting convergence to a neighborhood of the optimum.

- *Plot $||x_k - x_{true}||^2$ versus k and Interpret the plot.*

  Figure 6 shows the plot of the estimation error, defined as $||x_k - x_{true}||^2$, versus the iteration k on a log scale. The error presents a rapid decrease in the early iterations, indicating a strong initial descent phase. Afterward, it continues to make smaller but consistent progress. The approximately linear slope in the log-scale plot during the later phase suggests linear convergence of the iterates to the true solution x_{true}. This is consistent with theoreticcal expectations for coordinate descent methods applied to strongly convex quadratic problems, where the error contracts over time.

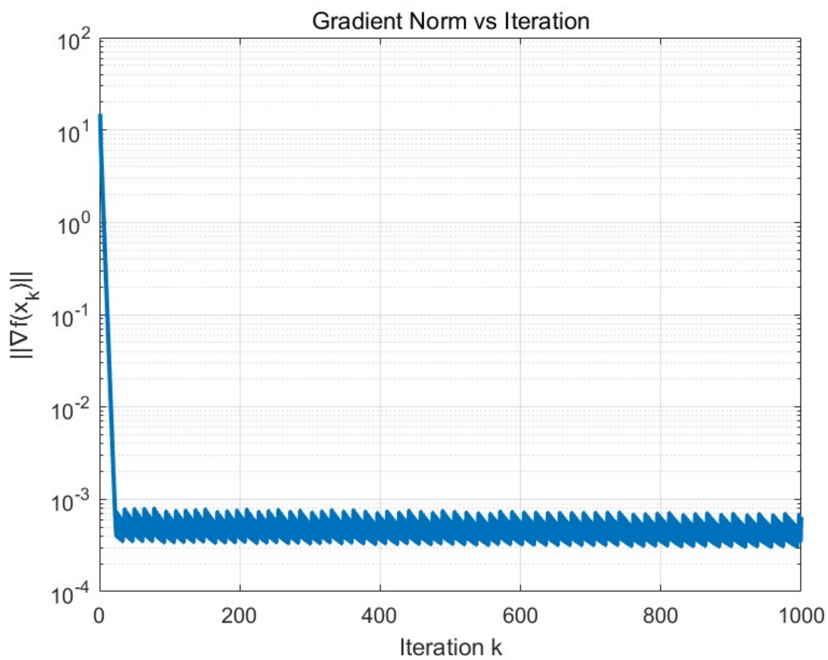

Figure 5: $||\nabla f(x_k)||$ versus Iteration k

- *Discuss global convergence and local convergence behaviors. What is the order of local convergence?*

The estimation error plot shows a clear, steady decline from the very first iteration, with no indication of divergence or oscillation in error magnitude. This confirms global convergence. Since the flat point $x_0 = (1, 1, 0)$ is initialized, and successfully converged to the neighborhood of $x_{true} = (1.0369, 0.9660, 0.1027)$, indicating that the gradient descent algorithm with backtracking line search method achieves global convergence in this problem.

In the later iterations, the estimation error decreases more slowly and approaches a flat tail, implying that the algorithm has entered the local convergence. Meanwhile, the gradient norm plot shows a sharp drop early on, followed by persistent oscillations around a small value, rather than a smooth decay to zero. This behavior typically appears when some directions converge faster than others due to curvature differences. These oscillations suggest that the gradient is not reduced uniformly in all directions per iteration, but rather component-wise in a non-uniform way.

From the estimation error plot, the tail of the curve is linear on the log-scale, meaning that he order of local convergence is linear (first-order, $p = 1$). This matches theoretical expectations for standard gradient descent with backtracking on sufficiently

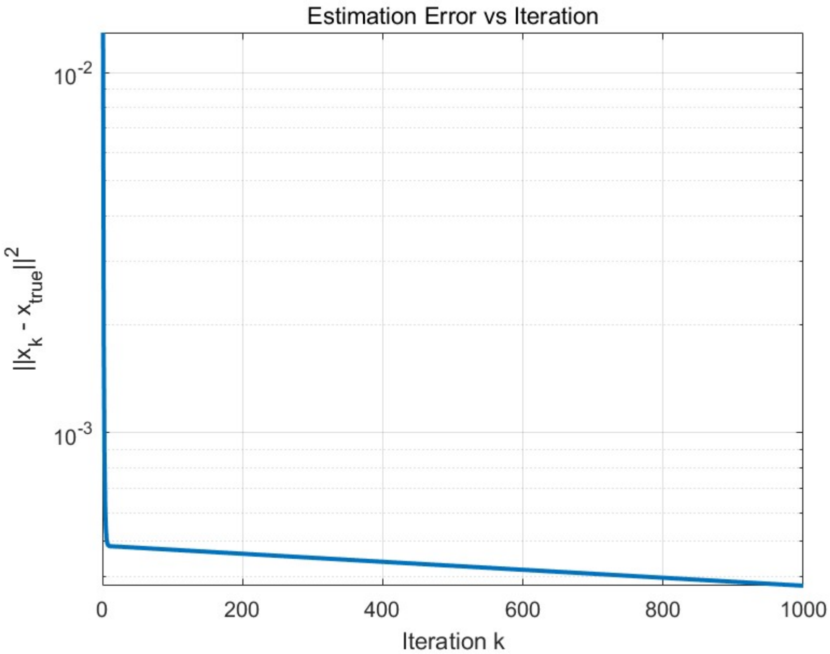

Figure 6: $||x_k - x_{true}||^2$ versus Iteration k

smooth non-linear problems, where the local convergence rate is linear unless second-order methods like Newton's methods are used.