# Homework 3. Global convergence theorem

Hyuntaek Oh

ohhyun@oregonstate.edu

Due: April 30, 2025

ECE599/ AI539 Nonlinear Optimization (Spring 2025)
Homework 3. Global convergence theorem (Due: 11:59pm on April 30, Wednesday.)

*Instruction:* Students should provide enough detail of the logical procedure of deriving answers. Answers without sufficient justification will receive partial or no credit.

**Reading:** Section 7.3, 7.4, 7.5, and Section 7.7 of the textbook (Luenberger and Ye).

1. Define the point-to-set mapping on $E^n$ by

$$\mathbf{A}(\mathbf{x}) = \{\mathbf{y} : \mathbf{y}^T\mathbf{x} \le b\},$$

where $b$ is a fixed constant. Is **A** closed?

*The definition of the closedness of algorithm $A$ is that a point-to-set mapping $A$ from $\mathcal{X}$ and $\mathcal{Y}$ is said to be closed at $\mathbf{x} \in \mathcal{X}$ if the following two conditions:*

- $x_k \to x$, and $x_k \in \mathcal{X}$ for all $k$,
- $y_k \to y$, and $y_k \in A(x_k)$ for all $k$

*These always imply $y \in A(x)$.*

*Based on the conditions above, when taking the limit on $x_k$ and $y_k$, they are close to* **x** *and* **y** *in algorithm $A$, meaning that:*

$$\lim_{k\to\infty} y_k^T x_k = y^T x$$

*However, there is a need to prove the relationship between $y_k^T x_k$ and $y^T x$ such that their relationship always holds. The gap between them should be zero if $x_k$ or $y_k$ converge to $x$ and $y$ respectively while going to the limit. This means:*

$$x_k - x \to 0, \quad y_k - y \to 0$$
$$y_k^T x_k - y^T x = 0$$

*By adding a term $y_k x$, the left side $y_k^T x_k - y^T x$ can be changed to:*

$$y_k^T x_k - y^T x \Rightarrow y_k^T x_k + (-y_k^T x + y_k^T x) - y^T x$$
$$y_k^T x_k + (-y_k^T x + y_k^T x) - y^T x = y_k^T (x_k - x) + (y_k - y)^T x$$

*Since the terms $(x_k - x)$ and $(y_k - y)$ go to zero based on $x_k - x \to 0, \quad y_k - y \to 0$, $\lim_{k\to\infty} y_k^T x_k = y^T x$ holds, and then the elements in the algorithm $A$ are:*

$$y \in A(x)$$

*Thus, A is closed.*

2. (Unconstrained Quadratic Optimization) Consider $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{a}^T\mathbf{x}$ and an unconstrained optimization:

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \tag{1}$$

Suppose that $\mathbf{Q} \in \mathbb{R}^{n\times n}$ is positive definite with its eigenvalues being $\lambda_n \geq \lambda_{n-1} \geq ... \geq \lambda_2 \geq \lambda_1 > 0$. We will consider applying a gradient descent algorithm with a fixed step size $\alpha$ ($\alpha > 0$) to solve this minimization problem. In other words, starting from some initial point $\mathbf{x}_0$, the sequence $\{\mathbf{x}_k\}$ is generated according to

$$\mathbf{x}_{k+1} = \mathbf{A}(\mathbf{x}_k), \text{ where } \mathbf{A}(\mathbf{x}_k) = \mathbf{x}_k - \alpha\mathbf{g}_k \quad (\mathbf{g}_k \triangleq \nabla f(\mathbf{x}_k)^T). \tag{2}$$

(a) Find a condition on $\alpha$ that ensures that for *any* $\mathbf{x}_k$ with $\nabla f(\mathbf{x}_k)^T \neq \mathbf{0}$, $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$.

*To find the condition, we use $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$:*

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) < 0$$

$f(x_{k+1})$ *would be:*

$$f(x_{k+1}) = \frac{1}{2}x_{k+1}^T Q x_{k+1} + a^T x_{k+1}$$

*Based on the given condition,* $x_{k+1}$ *can be changed to:*

$$x_{k+1} = A(x_k) = x_k - \alpha g_k$$

*It applies to $f(x_{k+1})$:*

$$\frac{1}{2}(x_k - \alpha g_k)^T Q(x_k - \alpha g_k) + a^T(x_k - \alpha g_k)$$

$$\frac{1}{2}x_k^T Q x_k + a^T x_k - \alpha g_k^T Q x_k - \alpha a^T g_k + \frac{1}{2}\alpha g_k^T Q g_k$$

*By using those equations, we can get:*

$$f(x_{k+1}) - f(x_k) = (-\alpha g_k^T Q x_k - \alpha a^T g_k) + \frac{1}{2}\alpha^2 g_k^T Q g_k$$

$$= -\alpha(g_k^T Q x_k + a^T g_k) + \frac{1}{2}\alpha^2 g_k^T Q g_k$$

$g_k$ *is* $g_k \triangleq \nabla f(x_k)^T = Qx_k + a$, *and this can be* $Qx_k = g_k - a$. *This means:*

$$(g_k^T Q x_k + a^T g_k) \Rightarrow g_k^T(g_k - a) + a^T g_k = g_k^T g_k = ||g_k||^2$$

Oregon State University College of Engineering

$$-\alpha(g_k^T Q x_k + a^T g_k) + \frac{1}{2}\alpha^2 g_k^T Q g_k \Rightarrow -\alpha||g_k||^2 + \frac{1}{2}\alpha^2 g_k^T Q g_k$$

$$-\alpha||g_k||^2 + \frac{1}{2}\alpha^2 g_k^T Q g_k < 0$$

*$g_k$ is equal to $\nabla f(x_k)^T$), and the inequality above can be simplified with respect to $\alpha$:*

$$\frac{\alpha}{2}\nabla f(x_k)^T Q \nabla f(x_k) < ||\nabla f(x_k)||^2$$

*So, the condition on $\alpha$ is:*

$$\alpha < \frac{2||\nabla f(x_k)||^2}{\nabla f(x_k)^T Q \nabla f(x_k)}$$

*For finding a safer and fixed $\alpha$, the condition that $g_k^T Q g_k$ is symmetric, positive definite $Q$ is used:*

$$\lambda_1 ||g_k||^2 \leq g_k^T Q g_k \leq \lambda_n ||g_k||^2$$

*This boundary condition applies to $f(x_{k+1}) - f(x_k)$:*

$$f(x_{k+1}) - f(x_k) = -\alpha(||g_k||^2 - \frac{1}{2}\alpha g_k^T Q g_k)$$

$$f(x_{k+1}) - f(x_k) \leq -\alpha(||g_k||^2 - \frac{1}{2}\alpha g_k^T Q g_k) = \alpha||g_k||^2(1 - \frac{1}{2}\alpha\lambda_n)$$

*To make the mathematical expression negative:*

$$1 - \frac{1}{2}\alpha\lambda_n > 0 \Rightarrow \alpha < \frac{2}{\lambda_n}$$

*By using positive definiteness of $Q$, $(\lambda_1 > 0, \quad \lambda_n > \lambda_1)$, we can conclude:*

$$f(x_{k+1}) < f(x_k) \text{ whenever } \nabla f(x_k) \neq 0, \text{ if } \alpha < \frac{2}{\lambda_n}$$

*Thus, the condition on $\alpha$ is:*

$$\alpha < \frac{2}{\lambda_n}$$

(b) Suppose that we set $\alpha$ such that it satisfies the condition in (a). Define a solution set $\Gamma$ to be $\Gamma = \{\mathbf{x} \in \mathbb{R}^n : \nabla f(\mathbf{x})^T = \mathbf{0}\}$. Consider a function $Z(\mathbf{x}) \triangleq f(\mathbf{x})$. Show that $Z$ is a descent function for the solution set $\Gamma$ and the algorithm $\mathbf{A}$.

*According to (a), The gradient $\nabla f(x)$ is:*

$$\nabla f(x) = Qx + a$$

*Since Q is positive definite, $\nabla f(x) = 0$ means:*

$$\nabla f(x) = 0 \Rightarrow Qx + a = 0 \Rightarrow x^* = -Q^{-1}a$$

*The solution $\Gamma$ is:*

$$\Gamma = \{x^* = -Q^{-1}a\}$$

*Considering a function $Z(x) \triangleq f(x)$, if $x \notin \Gamma$, the function $Z(x_{k+1})$ should be lower than $Z(x_k)$:*

$$x \notin \Gamma \rightarrow Z(x_{k+1}) < Z(x_k)$$

*In (a), the $\alpha$ is small enough, $f(x_{k+1}) < f(x_k)$ holds. If $x_k \notin \Gamma$, $\nabla f(x_k) \neq 0$, meaning that the gradient is not zero, and $||\nabla f(x_k)||^2 > 0$. So, by choosing very small $\alpha$, $f(x_{k+1}) < f(x_k)$ holds.*
*Thus, $Z(x) = f(x)$*

(c) Is **A** a closed mapping? Provide a justification for your answer.
*The definition of a closed mapping is based on the following conditions:*

$$x_k \rightarrow x, \text{ and } x_k \in \mathcal{X} \text{ for all } k,$$

$$y_k \rightarrow y, \text{ and } y_k \in A(x_k) \text{ for all } k$$

$$\text{then, } y \in A(x)$$

*The algorithm $A$ is written as:*

$$A(x_k) = x_k - \alpha \nabla f(x_k)^T$$
$$= x_k - \alpha(Qx_k + a)$$

*$y_k \in A(x_k)$ is needed to prove the algorithm $A$ is closed, and it means:*

$$y_k = x_k - \alpha(Qx_k + a)$$

*Based on the assumptions $x_k \rightarrow x$ and $y_k \rightarrow y$, when taking the limit on both side of the equation above, it would be:*

$$\lim_{k\rightarrow\infty} y_k = x - \alpha(Qx + a)$$
$$y = x - \alpha(Qx + a)$$

*This means that $y$ is:*

$$y = A(x)$$

*Therefore, the algorithm $A$ is closed.*

(d) Suppose that $\{||\mathbf{x}_k||\}_{k=0}^{\infty}$ is a bounded sequence, i.e., there exists $M > 0$ such that $||\mathbf{x}_k|| < M$ for all $k$. Verify that all the conditions of the Global Convergence Theorem hold for $\mathbf{A}$, $\Gamma$, and $Z$ defined above. Then, apply the Global Convergence Theorem to conclude that any limit point of $\{\mathbf{x}_k\}$ is in $\Gamma$.

*i. Verification of all the conditions of the Global Convergence Theorem*

*For algorithm $A$, a bounded sequence $\{||x_k||\}$ implies that there exist subsequences, and that $x_k$ converges to $x$ when taking the limit on it. In a similar way of (c), $y_k$ is:*

$$y_k = A(x_k) = x_k - \alpha(Qx_k + a)$$

*By taking the limit on both side:*

$$\lim_{k\to\infty} y_k = \lim_{k\to\infty}(x_k - \alpha(Qx_k + a))$$

*Based on the assumptions for closedness, the equation above would be:*

$$y = \lim y_k = x - \alpha(Qx + a) = A(x)$$

*Moreover, assume that the sequence $S := \{||x_k||\}_{k=0}^{\infty} \subset \mathbb{R}$ and its subsequence $S_1 = \{||x_k|| : k \in \mathbb{N}\} \subseteq S_1$. By assumption, $||x_k|| < M$ for all $k$, so $S_1$ is bounded:*

$$S_1 \subset [0, M) \Rightarrow S_1 \text{ is bounded in } \mathbb{R}$$

*The set $S_1$ is closed and bounded, meaning that compact in $\mathbb{R}$. Thus, the algorithm $A$ is a compact set (bounded, closedness)*
*Next, according to the lecture slides, the solution set $\Gamma$ is:*

$$\Gamma = \{x : \nabla f(x) = 0\}$$

*We know $\nabla f(x) = Qx + a$, so $\nabla f(x) = 0$ is:*

$$Qx + a = 0 \Rightarrow x = -Q^{-1}a$$

*Therefore, solution set $\Gamma$ is:*

$$\Gamma = \{-Q^{-1}a\}$$

*Lastly, let's check that the descent function $Z(x) = f(x)$ for $\Gamma$ and $A$, meaning that $f(x_{k+1}) < f(x_k)$ when $x_k \notin \Gamma$. So, $f(x_{k+1}) - f(x_k) < 0$ would be:*

$$f(x_{k+1}) - f(x_k) < 0$$

$$\Rightarrow f(x_{k+1}) - f(x_k) = -\alpha||\nabla f(x_k)||^2 + \frac{1}{2}\alpha^2 \nabla f(x_k)^T Q \nabla f(x_k)$$

$$-\alpha||\nabla f(x_k)||^2 + \frac{1}{2}\alpha^2 \nabla f(x_k)^T Q \nabla f(x_k) < 0$$

*If $x_k \notin \Gamma$, $\nabla f(x_k) \neq 0$, and thus $||\nabla f(x_k)||^2 > 0$. In this case, the first term $-\alpha||\nabla f(x_k)||^2$ is definitely negative, and the second term can be smaller since $\alpha^2$. Thus, $Z$ holds.*

*ii. Apply the Global Convergence Theorem to conclude that any limit point of $x_k$ is in $\Gamma$*

*Based on the verification, we know that:*

- *Sequence $\{||x_k||\}$ is a bounded sequence ($x_k \to x$)*
- *algorithm $A$ is closed mapping*
- *solution set $\Gamma = \{x : \nabla f(x) = 0\} = \{-Q^{-1}a\}$*
- *Descent function $Z(x) = f(x)$ holds*

*Therefore, due to these properties, any limit point $x^*$ of $\{x_k\}$ satisfies the first order condition and a solution.*

(e) Use the convexity of the problem to prove that any limit point of $\{\mathbf{x}_k\}$ is a global minimum point.

*The function $f$ is convex function when $\nabla^2 f(x) \geq 0$). Hessian matrix $F(x) = \nabla^2 f(x)$ is always $Q$, which is positive definite. As a result, $\nabla f(x^*) = 0$ by Global Convergence Theorem means $x^* \in \Gamma$, and $f(x)$ is convex function since Hessian matrix is positive definite, guaranteeing that the point is a global minimizer. Thus, $x^*$ is the unique global minimum point.*

3. (MATLAB experiment) Consider the unconstrained quadratic optimization in Problem 2 with

$$\mathbf{Q} = \begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 0.25 \\ 0 & 0.25 & 1 \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}. \tag{3}$$

(a) Find a proper step size $\alpha$ that satisfies the condition in Problem 2-(a).

*In Problem 2-(a), proper $\alpha$ would be:*

$$\alpha < \frac{2||\nabla f(x_k)||^2}{\nabla f(x_k)^T Q \nabla f(x_k)}$$

*The gradient of the function $f$ with given arbitrary input $x_k = [0, 0, 0]$ is:*

$$\nabla f(x_k) = Qx_k + a$$

$$= \begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 0.25 \\ 0 & 0.25 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}$$

*Based on the gradient above, the numerator of the fraction in inequality would be:*

$$2||\nabla f(x_k)||^2 = 2 * 6 = 12$$

*the denominator of the fraction would be:*

$$\nabla f(x_k)^T Q \nabla f(x_k) = \begin{bmatrix} 1 & -1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 0.25 \\ 0 & 0.25 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} = 4.5$$

*Thus, a proper step size $\alpha$ is:*

$$\alpha < \frac{12}{4.5} \approx 2.6667$$

(b) Find a global minimal point $\mathbf{x}^*$ of this problem using necessary conditions and sufficient conditions we discussed in lectures. Before using any condition, you should first verify whether the condition is applicable to this problem.

*For clarifying the First-Order Necessary Condition is applicable, we need to check these points:*

- *The function $f(x) = \frac{1}{2}x^T Q x + a^T x$ is differentiable everywhere*
- *The feasible set is $\Omega = \mathbb{R}^3$, which is convex and unconstrained.*
- *There exists any feasible direction at $x^*$*

*The function $f$ is Quadratic problem, and its gradient $\nabla f(x) = Qx + a$ is differentiable, meaning that $f \in C^1$. Moreover, the problem is unconstrained optimization, defining the feasible set as convex set. There are no constraints, meaning that any $d$ can be a feasible direction, $\nabla f(x^*)^T d \geq 0$ for all directions $d$. Therefore, necessary condition is applicable.*

*Likely, the following conditions are indispensable parts of meeting the Second-Order Sufficient Conditions:*

- *The function $f$ is twice continuously differentiable*
- *$\nabla f(x^*) = 0$ for some $x^*$*
- *The Hessian matrix $\nabla^2 f(x^*)$ is positive definite*

*The function $f$ is quadratic problem, so it can be twice continuously differentiable. In Problem 2, we get $x^* = -Q^{-1}a$ as a solution. This satisfies the first-order condition:*

$$\nabla f(x^*) = Qx^* + a = Q(-Q^{-1}) + a = -a + a = 0$$

*The Hessian matrix of the function $f$ is $Q$, which is positive definite since its eigenvalues are non-negative. As a result, the second-order sufficient conditions are applicable.*
*In conclusion, $x^* = -Q^{-1}a$ is a global minimal point of the function $f(x)$.*

(c) Implement the gradient descent algorithm with the $\alpha$ value you found in part (a). The gradient descent algorithm can be performed as follows:

**Step 1. Initialize** Set $\mathbf{x}_0$ to an arbitrary vector in $\mathbb{R}^3$ (e.g., $\mathbf{0}$). Set $k = 0$.

**Step 2. Descent** Obtain $\mathbf{x}_{k+1}$ as $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \mathbf{g}_k$ where $\mathbf{g}_k \triangleq \nabla f(\mathbf{x}_k)^T$.

**Step 3. Stopping Rule** If $||\nabla f(\mathbf{x}_{k+1})^T|| < \epsilon$, return $\mathbf{x}_{k+1}$ and terminate. Otherwise, increase $k$ by 1 and go to Step 2.

You can set $\epsilon$ to a small positive real number, e.g., $10^{-7}$.

Run this algorithm to numerically find a global minimal point. Plot $||\mathbf{x}_k - \mathbf{x}^*||$ versus $k$ and interpret the plot. Please make the $y$-axis of the plot have log scale such that you can distinguish a very small value (e.g., $10^{-5}$) from an even smaller value (e.g., $10^{-6}$). Please provide your code as an attachment.

*By running the program code 2, a global minimal point $x^*$ is [-1.8, 1.6, -2.4] given $\alpha = 1.25$. Figure 1 shows the convergence behavior of the gradient algorithm applied to the unconstrained quadratic optimization problem. The vertical axis represents the logarithmic scale of the error norm $||x_k - x^*||$, where $x^*$ is the computed global minimizer, and the horizontal axis denotes the iteration count $k$.*

*The error decreases nearly linearly in the log-scale plot, indicating exponential convergence of the algorithm. Within 29 iterations, the error norm drops from above $10^0$ to below $10^-7$. This convergence behavior confirms that the step size $\alpha$ was properly chosen to lie within the theoretically valid range, and that the algorithm efficiently and reliably converges to the global minimizer.*

```
Maximum eigenvalue: 1.2500
Safe upper bound for alpha: alpha < 1.6000
Converged at iteration 29
Current alpha: 1.2500
x* = [-1.800000, 1.600000, -2.400000]
Final error ||x_k - x^*|| = 5.56e-08
```

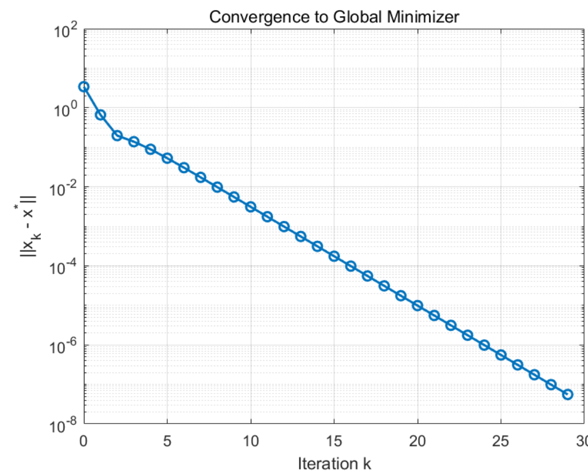


Figure 1: The result of the algorithm to find a global minimal point

```
% -------- Problem setting ------------
Q = [1, 0.5, 0;
     0, 1, 0.25;
     0, 0.25, 1];

a = [1; -1; 2];

grad_f = @(x) Q*x + a;
f = @(x) 0.5 * x' * Q * x + a' * x;

lambda = eig(Q);
lambda_max = max(lambda);
alpha_upper = 2 / lambda_max;

fprintf("Maximum eigenvalue: %.4f\n", lambda_max);
fprintf("Safe upper bound for alpha: alpha < %.4f\n", alpha_upper);
%-------------------------------------

% --------- Gradient Descent -------------
for k = 1: max_iter
    g = grad_f(x);
    x_diff_norms(end+1) = norm(x - x_star);

    if norm(g) < epsilon
        break;
    end

    x = x - alpha * g;
end
%-----------------------------------------
```

```
% --------- Initialization ------------
x = [0; 0; 0]; %initial point x_0

%alpha = 1.0;
alpha = lambda_max;
epsilon = 1e-7;
max_iter = 1000;

x_star = -Q \ a;
x_diff_norms = [];
%-----------------------------------
```

Figure 2: Program code