Gradient Based Optimization I

Asst. Prof. Dr.-Ing. Sudchai Boonto

Department of Control System and Instrumentation Engineering
King Mongkut's University of Technology Thonburi
Thailand

August 27, 2025

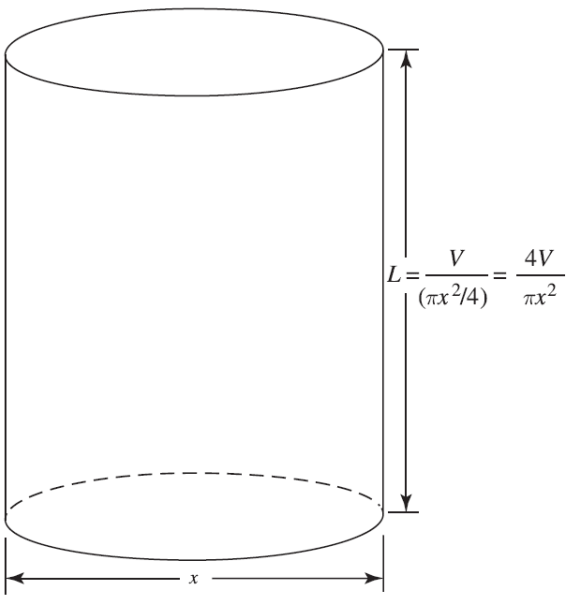
Objective

At the end of this chapter you should be able to:

- Mathematically define the optimality conditions for an unconstrained problem.
- Describe, implement, and use line-search-based methods.
- Gradient Descent based method

Unconstrained optimization problems

Determine the objective function for building a minimum cost cylinderical refrigeration tank of volume 50 m^3, if the circular ends cost \$ 10 per m^2 , the cylindrical wall costs \$6 per m^2, and it costs \$80 per m^2 to refrigerate over the useful life.



$$
\begin{aligned}
f(x, L) &= (10)(2)\left(\frac{\pi x^2}{4}\right) \\
&\quad + (6)(\pi x L) + 80\left((2)\frac{\pi x^2}{4} + \pi x L\right) \\
&= 45\pi x^2 + 86\pi x L \\
L &= \frac{(50)(4)}{\pi x^2} = \frac{200}{\pi x^2} \\
f(x) &= 45\pi x^2 + \frac{17200}{x}
\end{aligned}
$$

One problem is minimize $f(x)$ for all real x. How!

Unconstrained optimization problems

We consider unconstrained optimization problems with continuous design variables,

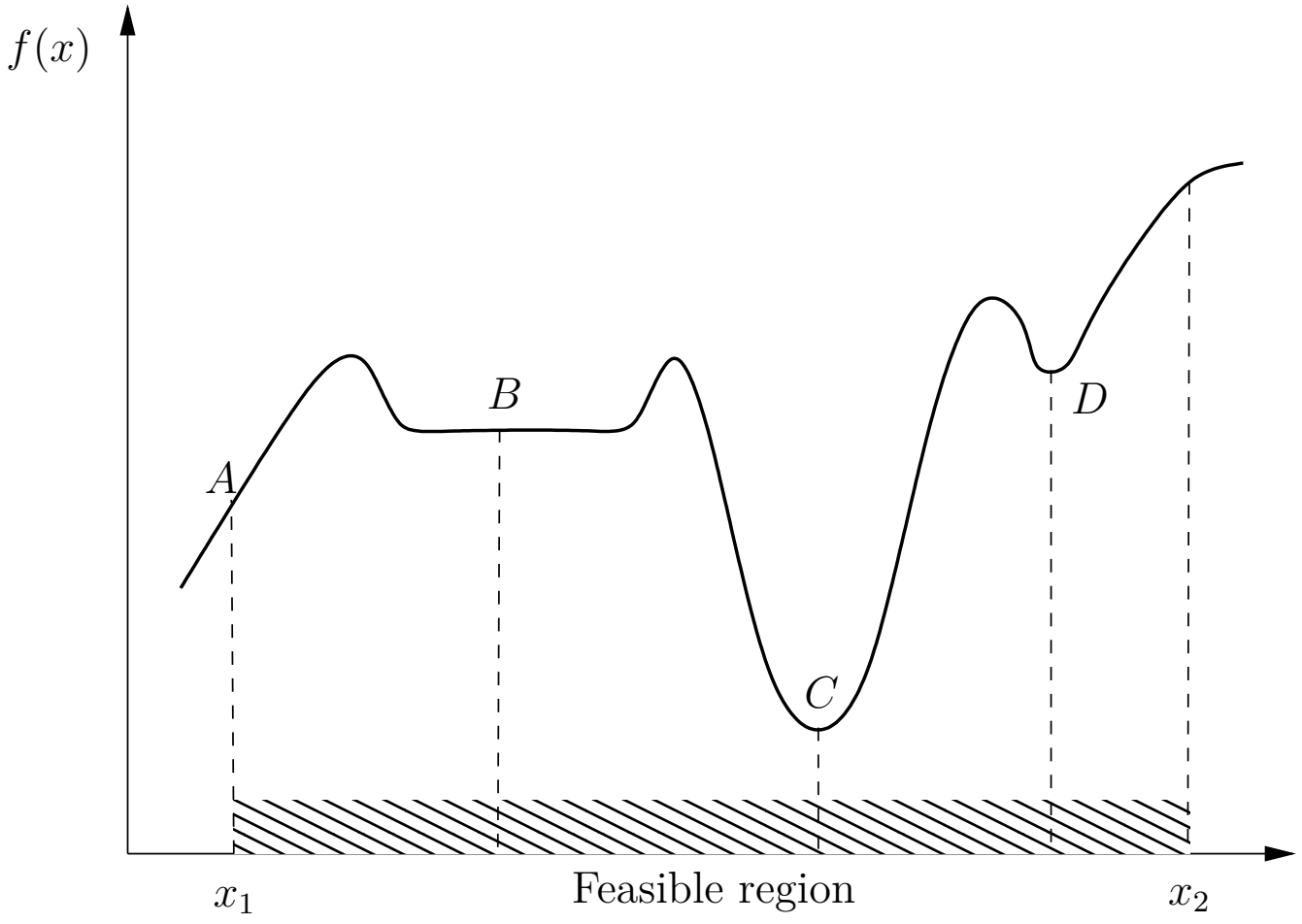
$$\underset{\mathbf{x}}{\text{minimize}}\, f(\mathbf{x}),$$

where $\mathbf{x} = [x_1, \ldots, x_n]$ is composed of the design variables that the optimization algorithm can change.

Minimum Points:

- the point $\mathbf{x}^*$ is a **weak local minimum** if there exists a $\delta > 0$ such that $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x}$ such that $|\mathbf{x} - \mathbf{x}^*| < \delta$, that is $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x}$ in a δ-neighborhood of $\mathbf{x}^*$.
- the point $\mathbf{x}^*$ is a **strong local minimum** if there exists a $\delta > 0$ such that $f(\mathbf{x}^*) < f(\mathbf{x})$ for all $\mathbf{x}$ such that $|\mathbf{x} - \mathbf{x}^*| < \delta$.
- $\mathbf{x}^*$ is a **global minimum** if $f(\mathbf{x}^*) < f(\mathbf{x})$ for all $\mathbf{x}$

Unconstrained optimization problems



Optimality Conditions

A point $\mathbf{x}^*$ is a local minimum if $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x}$ in the neighborhood of $\mathbf{x}^*$. A second-order Taylor series expansion about $\mathbf{x}^*$ for small steps of size $\mathbf{p}$ yields

$$f(\mathbf{x}^* + \mathbf{p}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T \mathbf{p} + \frac{1}{2}\mathbf{p}^T \mathbf{H}(\mathbf{x}^*)\mathbf{p} + \dots$$
$$f(\mathbf{x}^* - \mathbf{p}) = f(\mathbf{x}^*) - \nabla f(\mathbf{x}^*)^T \mathbf{p} + \frac{1}{2}\mathbf{p}^T \mathbf{H}(\mathbf{x}^*)\mathbf{p} + \dots$$

For x^* to be an optimal point, we must have $f(\mathbf{x}^* + \mathbf{p}) \geq f(\mathbf{x}^*)$ for all $\mathbf{p}$. This implies that

$$\nabla f(\mathbf{x}^*)^T \mathbf{p} + \frac{1}{2}\mathbf{p}^T \mathbf{H}(\mathbf{x}^*)\mathbf{p} \geq 0 \text{ and } -\nabla f(\mathbf{x}^*)^T \mathbf{p} + \frac{1}{2}\mathbf{p}^T \mathbf{H}(\mathbf{x}^*)\mathbf{p} \geq 0 \qquad \forall \mathbf{p}.$$

The magnitude of $\mathbf{p}$ is small, the second term can be neglected. Therefore, we require that

$$\nabla f(\mathbf{x}^*)^T \mathbf{p} \geq 0 \text{ and } -\nabla f(\mathbf{x}^*)^T \mathbf{p} \geq 0 \implies \nabla f(\mathbf{x}^*) = 0$$

Optimality Conditions

The condition $\nabla f(\mathbf{x}^*) = 0$ is called the first-order necessary optimality condition.

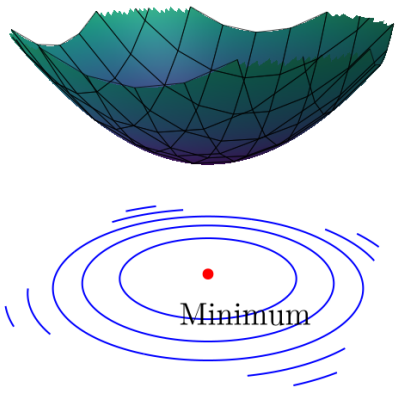
Because the gradient term has to be zero, we must satisfy the remaining term in the previous inequality, that is

$$\mathbf{p}^T \mathbf{H}(\mathbf{x}^*)\mathbf{p} \geq 0 \quad \forall \mathbf{p} \quad \text{or } \mathbf{H}(\mathbf{x}^*) \succeq 0$$

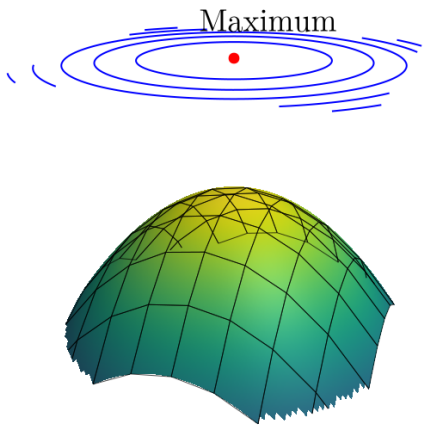
- These two conditions $\nabla f(\mathbf{x}^*) = 0$ and $\mathbf{H}(\mathbf{x}^*) \succeq 0$ are necessary conditions for a local minimum but not sufficient.
- In some direction $\mathbf{p}^T \mathbf{H}(\mathbf{x}^*)\mathbf{p}$ can be zero. We need to check the third-order term. If it is a minimum, it is a weak minimum.
- To have the sufficient optimality condition, $\mathbf{H}(\mathbf{x}^*)$ must be positive definite.

Quadratic function with different types of Hessian

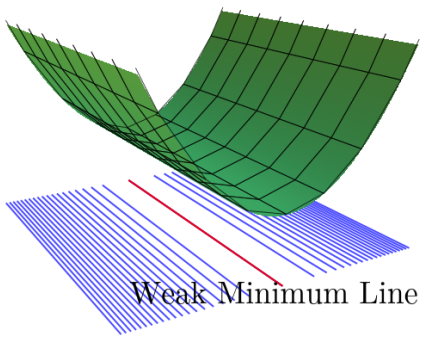
Positive Definite



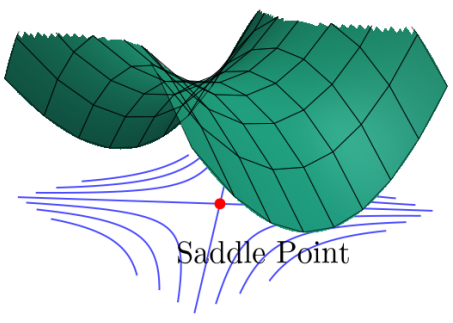
Negative Definite



Positive Semidefinite



Indefinite



Optimality Conditions: Finding minima analytically

Consider the following function of two variables:

$$f(x_1, x_2) = 0.5x_1^4 + 2x_1^3 + 1.5x_1^2 + x_2^2 - 2x_1x_2$$

Let the gradient equal to zero,

$$\nabla f(x_1, x_2) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1^3 + 6x_1^2 + 3x_1 - 2x_2 \\ 2x_2 - 2x_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

From the second row, we have $x_1 = x_2$. Substituting this in to the first equation yields

$$x_1 \left(2x_1^2 + 6x_1 + 1\right) = 0 \quad \Longrightarrow \quad x_1 = 0, -2.8223, -0.1771$$

The solution of this equation has three points: $x_A = (0, 0)$, $x_B = (-2.8223, -2.8223)$, and $x_C = (-0.1771, -0.1771)$. `(see ch2/optimal_condition.jl)`

Optimality Conditions: Finding minima analytically

To clarify these three points, we need to find the Hessian matrix.

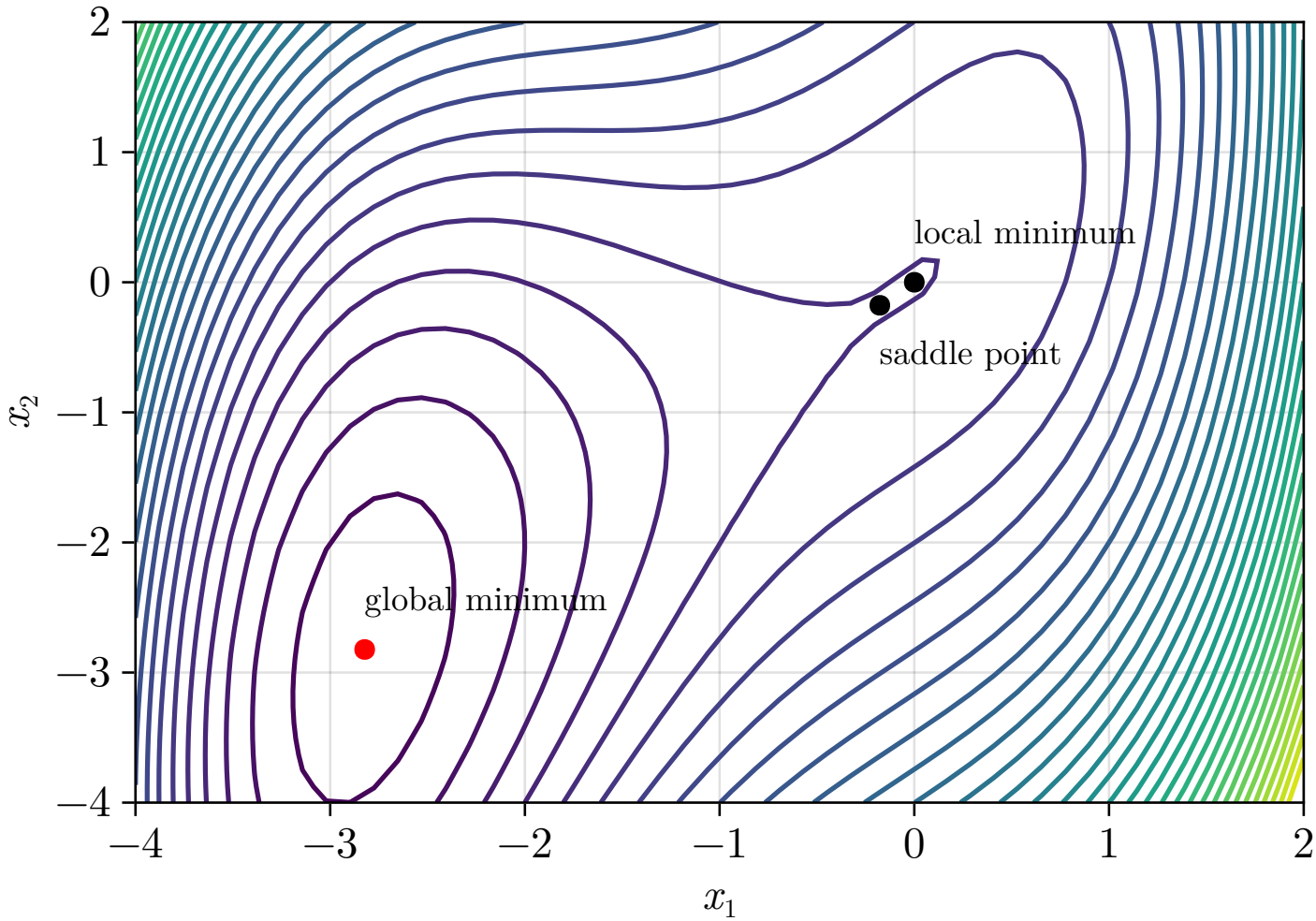
$$\mathbf{H}(x_1, x_2) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 6x_1^2 + 12x_1 + 3 & -2 \\ -2 & 2 \end{bmatrix}$$

For each point, we have

$$\mathbf{H}(\mathbf{x}_A) = \begin{bmatrix} 3 & -2 \\ -2 & 2 \end{bmatrix}, \qquad \mathbf{H}(\mathbf{x}_B) = \begin{bmatrix} 16.9373 & -2 \\ -2 & 2 \end{bmatrix}, \qquad \mathbf{H}(\mathbf{x}_C) = \begin{bmatrix} 1.0627 & -2 \\ -2 & 2 \end{bmatrix}$$

The eigenvalues are $\lambda_A = (0.438, 4.561)$, $\lambda_B = (1.737, 17.200)$, and $\lambda_C = (-0.523, 3.586)$, respectively. The first two eigenvalues show the evidence of the local minimum points, while the last one addresses the saddle point.

Optimality Conditions
Example: Finding minima analytically



Optimality Conditions
Example: Finding minimum analytically

We want to

$$\underset{x}{\text{minimize}} \qquad 45\pi x^2 - \frac{17200}{x}$$

We set

$$\nabla f = 90\pi x - \frac{17200}{x^2} = 0 \qquad \Longrightarrow \qquad x^3 = \frac{17200}{90\pi} = 60.833$$

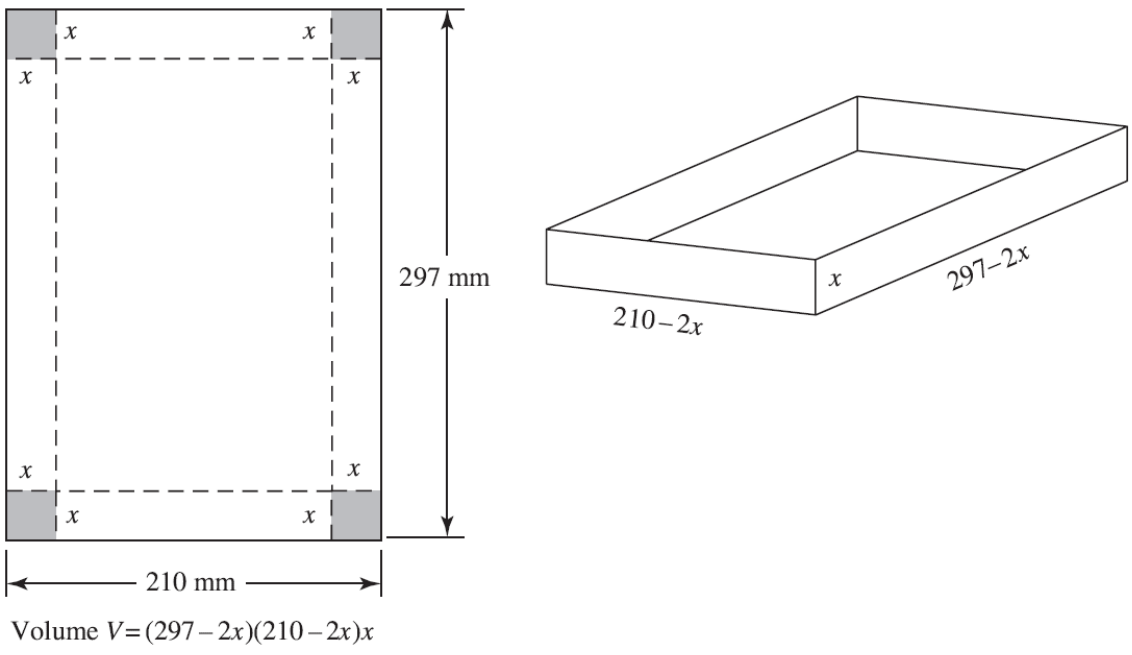
We have $x = 3.93$ m and $L = 200/(\pi x^2) = 4.12$ m.
The cost is

$$f(x^*) = 45(x^*)^2 + \frac{17200}{x^*} = 6560$$

Since $\mathbf{H}(x^*) = 90\pi + (3(17200))/(x^*)^3 = 1132.85$, it is strictly positive. Thus the solution is a strict or strong minimum.

Optimality Conditions
Example: Maximize Problem

Determine the dimensions of an open box of maximum volume that can be constructed form an A4 sheet 210 mm $\times$ 297 mm by cutting four squares of side x from the corners and folding and gluing the edges as shown in Fig.



Volume $V = (297 - 2x)(210 - 2x)x$

The problem is to

$$\underset{x}{\text{maximize}} \qquad V(x) = (297 - 2x)(210 - 2x)x = 62370x - 1014x^2 + 4x^3$$

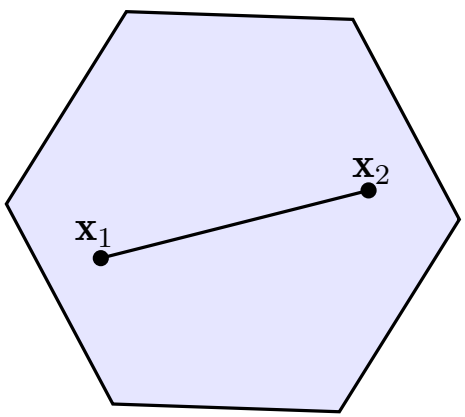
We set $f(x) = -V(x) = -62370x + 1014x^2 - 4x^3$. Setting $\nabla f(x) = 0$, we get

$$\nabla f(x) = -62370 + 2028x - 12x^2 = 0 \quad \Longrightarrow \quad x = 40.423 \text{ and } 128.577 \text{ mm.}$$

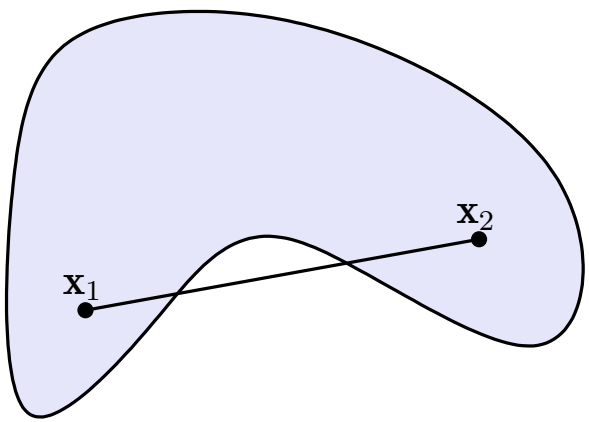
The possible solution of x is only the first one, where $x^* = 40.423$ mm. The $\mathbf{H}(x^*) = 2028 - 24x^* = 1057.848 > 0$. implies that x^* is a strict minimum of $f(x)$ or maximum of $V(x)$.
The maximum value of the box is 1128.5 cm^3.

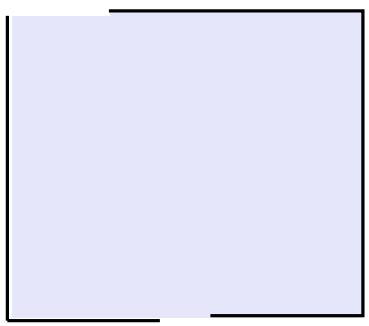
Convex Sets

- A set $\mathbb{S}$ is called a convex set if for any two points in the set, every point on the line joining the two points is in the set.
- Alternatively, the $\mathbb{S}$ is convex if for every pair of points $\mathbf{x}_1$ and $\mathbf{x}_2$ in $\mathbb{S}$, and every α such that $0 < \alpha < 1$, the point $\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2$ is in $\mathbb{S}$.
- example of convex sets:
  - the set of all real numbers $\mathbb{R}$ is a convex set.
  - any closed interval of $\mathbb{R}$ is also a convex set.
  - $\mathbf{A} = \{x \in \mathbb{R} : 0 \leq x \leq 1\}$, $\mathbf{B} = \{x \in \mathbb{R} : 2 \leq x \leq 3\}$ and $\mathbb{S} = \mathbf{A} \cup \mathbf{B}$. $\mathbb{S}$ is not a convex set.



convex set





nonconvex set

Convex Functions

- A function $f(\mathbf{x})$ defined over a convex set $\mathbb{R}_c$ is said to be convex if for every pair of points $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}_c$ and every real number $0 \leq \alpha \leq 1$, the inequality

$$f(\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2) \leq \alpha f(\mathbf{x}_1) + (1-\alpha)f(\mathbf{x}_2)$$

  hold. If $x_1 \neq x_2$ and

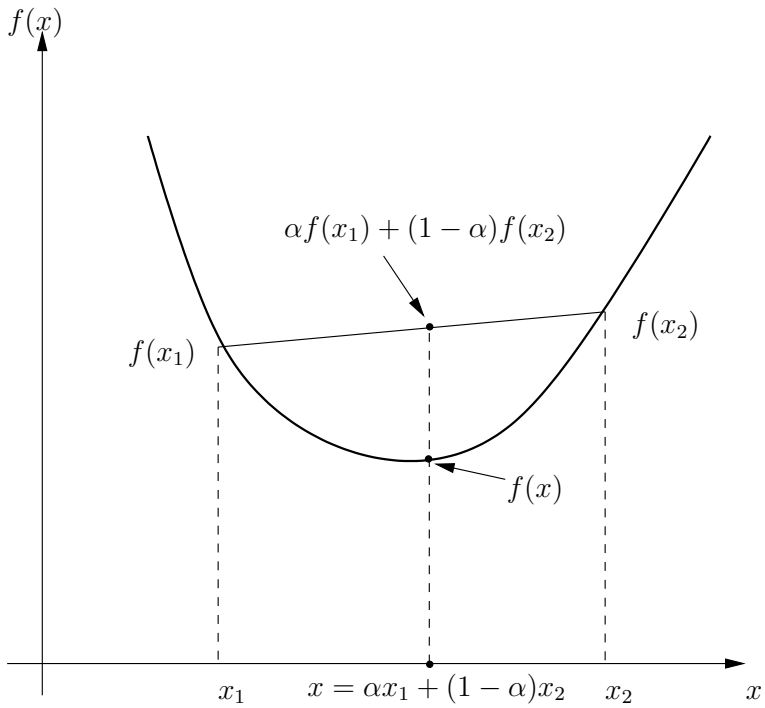
$$f(\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2) < \alpha f(\mathbf{x}_1) + (1-\alpha)f(\mathbf{x}_2)$$

  then $f(\mathbf{x})$ is said to be strictly convex.

- If $\psi(\mathbf{x})$ is defined over a convex set $\mathbb{R}_c$ and $f(\mathbf{x}) = -\psi(\mathbf{x})$ is convex, then $\phi(\mathbf{x})$ is said to be concave. If $f(\mathbf{x})$ is strictly convex, $\psi(\mathbf{x})$ is strictly concave.

is located in $\mathbb{R}_c$

Properties Convex Functions



Properties of Convex Functions

- If f has continuous first derivatives then f is convex over a convex set $\mathbf{S}$ if and only if for every x and y in $\mathbf{S}$, $f(y) \geq f(x) + f'(x)(y - x)$ This means that the graph of the function lies above the tangent line drawn at point show in above figure.

Properties Convex Functions

- If f has continuous second derivatives then f is convex over a convex set $\mathbf{S}$ if and only if for every x in $\mathbf{S}$,

$$f''(x) \geq 0$$

- If $f(x^*)$ is a local minimum for a convex function f on a convex set $\mathbf{S}$, then it is also a global minimum.
- If f has continuous first derivatives on a convex set $\mathbf{S}$ and for a point x^* in $\mathbf{S}$, $f'(x^*)(y - x^*) \geq 0$ for every y in $\mathbf{S}$, then x^* is a global minimum point of f over $\mathbf{S}$.

Example: Convex Function

Prove that $f = |x|, x \in \mathbb{R}^1$, is a convex function. Using the triangular inequality $|x + y| \leq |x| + |y|$, we have, for any two real numbers x_1 and x_2 and $0 < \alpha < 1$,

$$\begin{aligned} f(\alpha x_1 + (1-\alpha)x_2) &= |\alpha x_1 + (1-\alpha)x_2| \leq \alpha|x_1| + (1-\alpha)|x_2| \\ &\leq \alpha f(x_1) + (1-\alpha)f(x_2) \end{aligned}$$

Descent Direction

Consider an unconstraint problem

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \qquad f(\mathbf{x})$$

Definition: Descent Direction

A vector $\mathbf{d} \in \mathbb{R}^n$ is called a *descent direction* for f at $\mathbf{x}$ if moving a small amount in that direction decreases the function value.

$$\exists \alpha > 0 \text{ such that } f(\mathbf{x} + \alpha\mathbf{d}) < f(\mathbf{x})$$

To check the descent direction, we could use the First-order Condition. From a first-order Taylor approximation around $\mathbf{x}$:

$$f(\mathbf{x} + \alpha\mathbf{d}) \approx f(\mathbf{x}) + \alpha\nabla f(\mathbf{x})^T\mathbf{d}$$

For small $\alpha > 0$, we will have decrease if: $\nabla f(\mathbf{x})^T\mathbf{d} < 0$. This condition defines a descent direction.

Example: Steepest Descent

The most common descent direction is the *negative gradient:*

$$\mathbf{d} = -\nabla f(\mathbf{x})$$

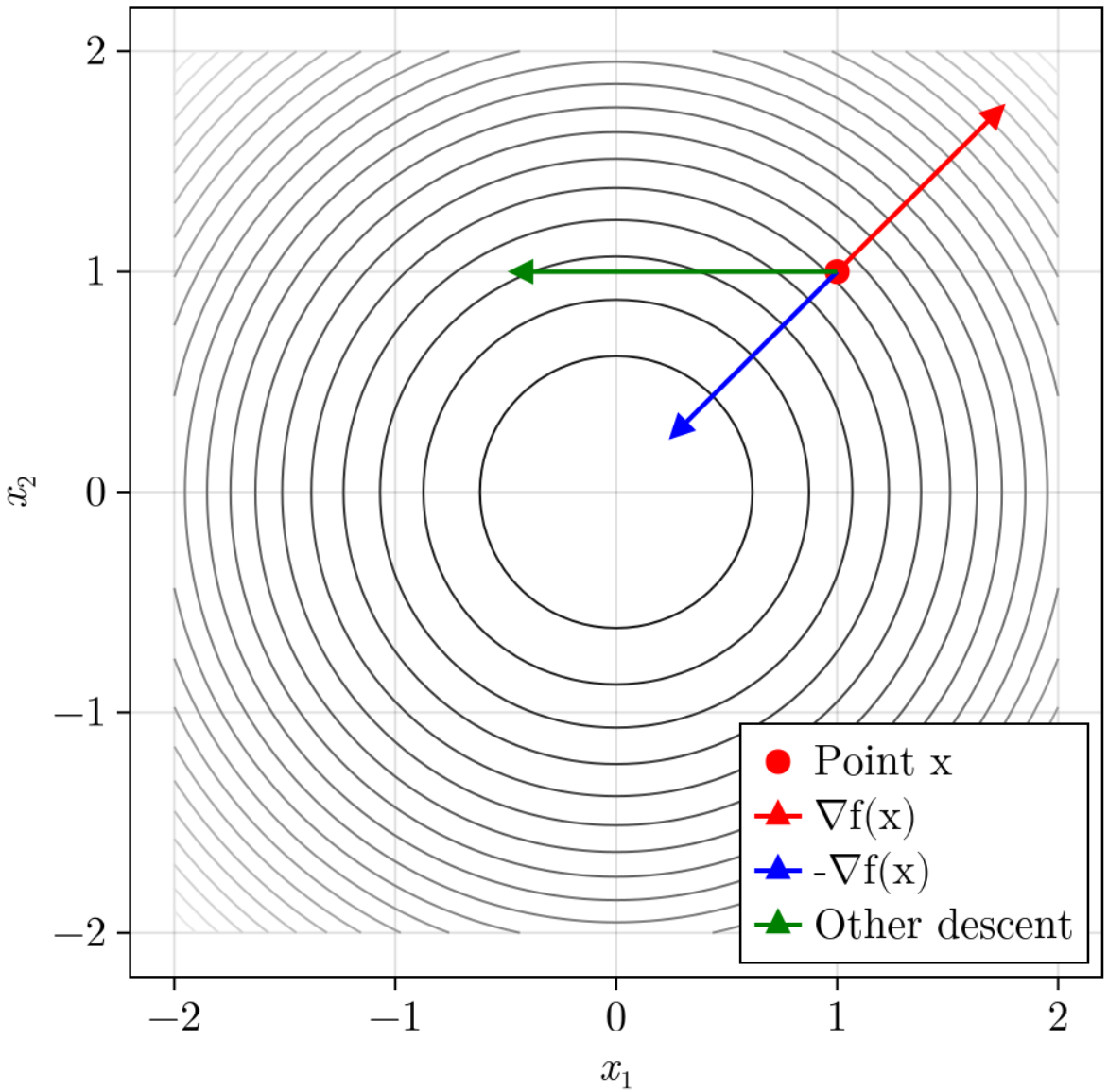
Why? Because:

$$\nabla f(\mathbf{x})^T(-\nabla f(\mathbf{x})) = -\|\nabla f(\mathbf{x})\|^2 < 0, \text{ where } \nabla f(\mathbf{x}) > 0$$

so it always guarantees descent.

- A descent direction is any vector $\mathbf{d}$ such that $\nabla f(\mathbf{x})^T\mathbf{d} < 0$
- It ensures that moving a little in direction $\mathbf{d}$ decreases $f(\mathbf{x})$
- The negative gradient $-\nabla f(\mathbf{x})$ is the steepest descent direction.

Example: Steepest Descent



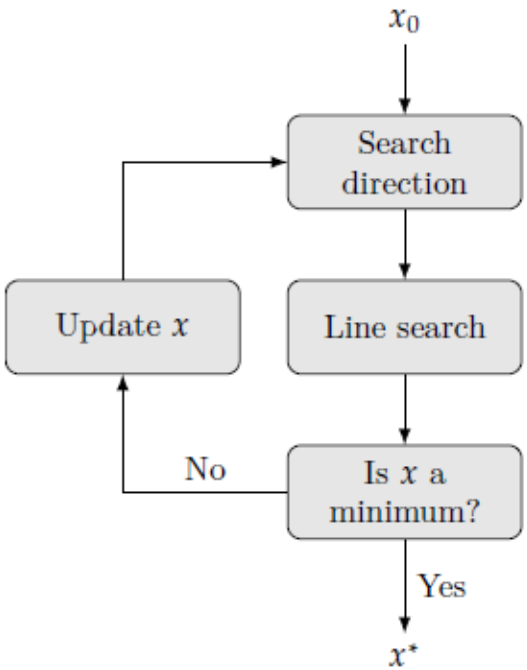
Gradient Based Optimization

A Gradient descent is one of the most popular and versatile technique in all of optimization.

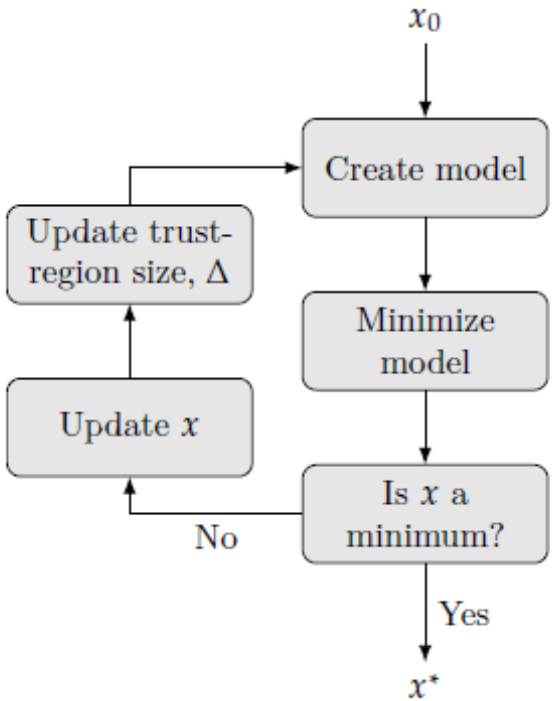
- A local minimum of a smooth, unconstrained objective function $f(\mathbf{x}$ will be a point $\mathbf{x}^*$ with zero gradient, $\nabla f(\mathbf{x}^*) = 0$, and positive semi-definite Hessian, $\nabla^2 f(\mathbf{x}^*) > 0$.
- The idea is find the point $\mathbf{x}$ where $\nabla f(\mathbf{x}) = 0$. We known as a root finding problem.
- However, in general, the function $\nabla f(x)$ is too complex to solve for its roots. We need to use iterative method to obtain a sequence of point $\mathbf{x}_k$ that eventually converge towards the local minimum $\mathbf{x}^*$:

$$f(\mathbf{x}_0) \geq f(\mathbf{x}_1) \geq f(\mathbf{x}_2) \geq \cdots \geq f(\mathbf{x}^*)$$
$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) < 0 \implies \mathbf{x}_{k+1} = F(\mathbf{x}_k)$$

Two Approaches to Finding and Optimum



Line search approach



Trust-region approach

Basic Concept

Consider a problem

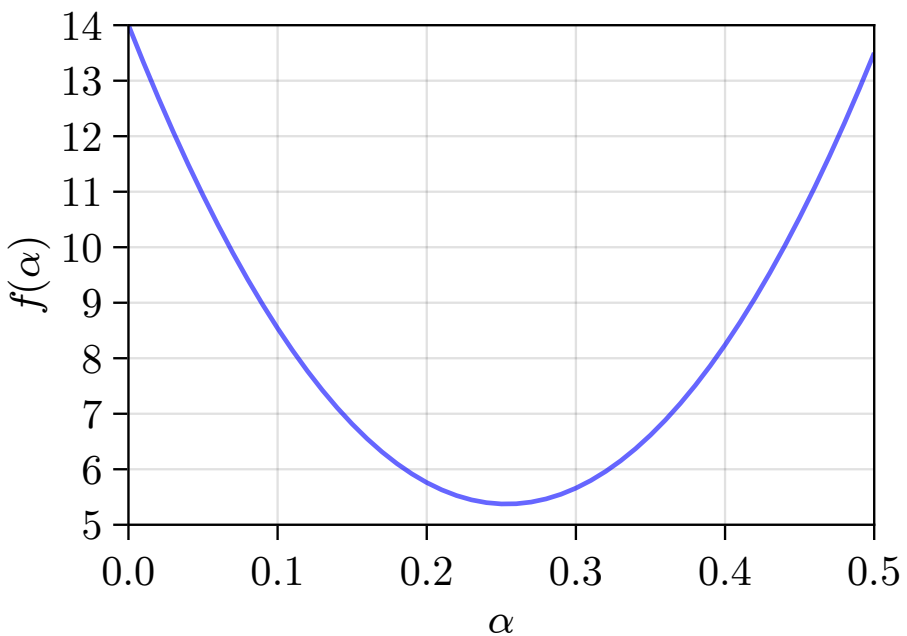
$$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n$$

- Most numerical methods require a starting design or point which we call $\mathbf{x}_0$ (initial point).
- We then determine the *direction of travel* $\mathbf{d}_0$.
- A *step size* α_0 is then determined based on minimizing f as much as possible and the design point is updated as $\mathbf{x}_1 = \mathbf{x}_0 + \alpha_0 \mathbf{d}_0$.
- The process of where to go and how far to go are repeated from $\mathbf{x}_1$ or $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$.

Basic Concept: Example

Given $f(x_1, x_2) = x_1^2 + 5x_2^2$, a point $\mathbf{x}_0 = [3\ 1]^T$, $f_0 = f(\mathbf{x}_0) = 14$.

1. Assume we already have a descent direction $\mathbf{d} = [-3\ \ -5]^T$. We need to find a step size α.
2. Construct $f(\alpha) = f(\mathbf{x}_0 + \alpha\mathbf{d})$ along the direction $\mathbf{d}$ and provide a plot of $f(\alpha)$ versus α, for $\alpha \geq 0$. We have $\mathbf{x}(\alpha) = \mathbf{x}_0 + \alpha\mathbf{d} = [3 - 3\alpha,\ 1 - 5\alpha]^T$ and $f(\alpha) = (3 - 3\alpha)^2 + 5(1 - 5\alpha)^2$.



Basic Concept : Example

2. Find the slope $df(\alpha)/d\alpha)$ at $\alpha = 0$. Verify that this equal $\nabla f(\mathbf{x}_0)^T \mathbf{d}$ (direction derivative). We have

$$\left.\frac{df(\alpha)}{d\alpha}\right|_{\alpha=0} = (-6(3-3\alpha) - 50(1-5\alpha))|_{\alpha=0} = -68$$

$$\nabla f(\mathbf{x}_0)^T \mathbf{d} = \begin{bmatrix} 2(3) & 10(1) \end{bmatrix} \begin{bmatrix} -3 \\ -5 \end{bmatrix} = -68$$

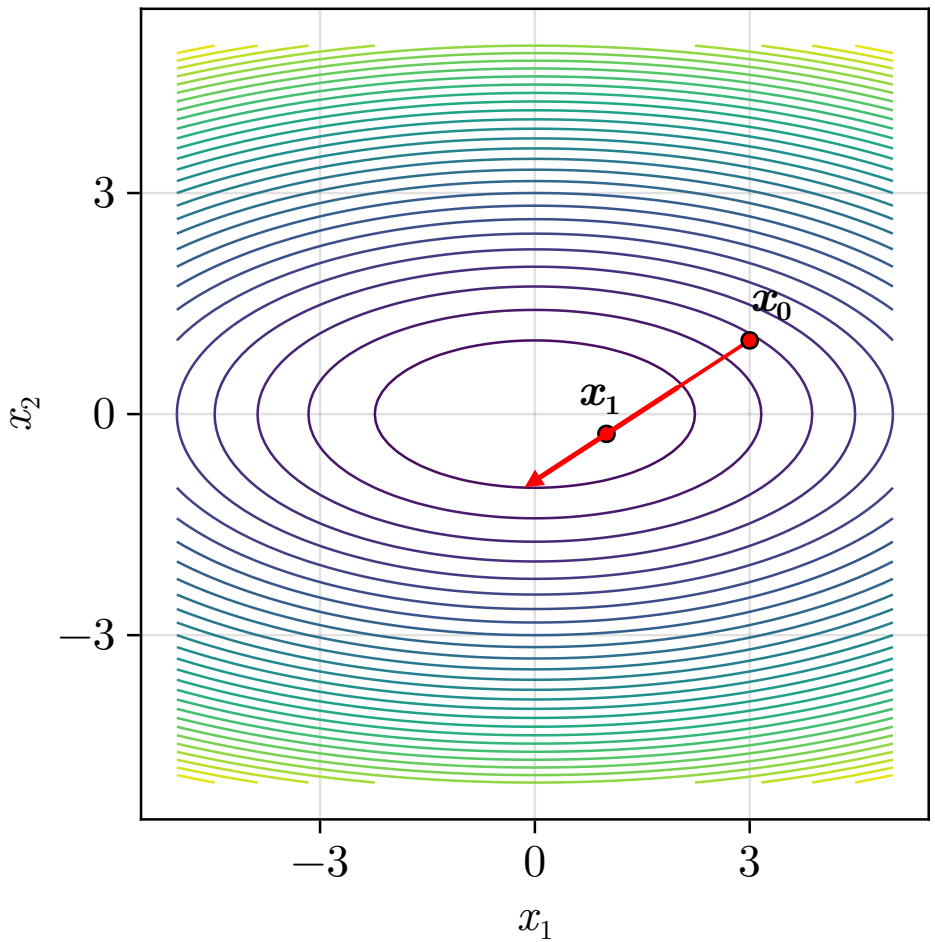
3. Minimize $f(\alpha)$ with respect to α, to obtain step size α_0. Given the corresponding new point $\mathbf{x}_1$ and value of $f_1 = f(\mathbf{x}_1)$. We have $df(\alpha)/d\alpha = 0$ or

$$\frac{df(\alpha)}{d\alpha} = -6(3-3\alpha) - 50(1-5\alpha) = 0 \Longrightarrow 268\alpha = 68 \text{ or } \alpha = 0.2537$$

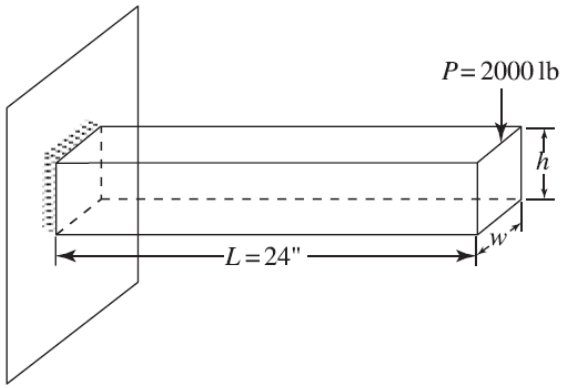
$$\mathbf{x}_1 = \begin{bmatrix} 3 \\ 1 \end{bmatrix} + \alpha_0 \begin{bmatrix} -3 \\ -5 \end{bmatrix} = \begin{bmatrix} 2.2388 \\ -0.2687 \end{bmatrix}, \; f(\mathbf{x}_1) = 5.3732, \text{ less than } f_0 = 14.$$

Basic Concept : Example

2. Provide a plot showing contours of the function, steepest descent direction x_0 and x_1.



Basic Concept : Example



We want to design the width and height of the rectangular cross-section to increase the bending stress defined by

$$\sigma_0 = \frac{6M}{wh^2}, \text{ where } M \text{ is a moment.}$$

With the initial design $\mathbf{x} = (w, h) = (1, 3)$, we have

$$\sigma_0 = \frac{6(2000 \times 24)}{1(3^2)} = 32,000\text{psi}$$

Using $\mathbf{d} = [-1/\sqrt{5} \ -2/\sqrt{5}]^T$ and $\alpha = 0.2$ we have

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha\mathbf{d} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} + 0.2 \begin{bmatrix} -1/\sqrt{5} \\ -2/\sqrt{5} \end{bmatrix} = \begin{bmatrix} 0.9106 \\ 2.8211 \end{bmatrix}, \qquad \sigma_1 = 71,342 \text{ psi}$$

Gradient Descent Method

Gradient descent seeks to find a local minima of an objective function $f(\mathbf{x})$ by taking iterative steps in the negative gradient direction.

- The gradient of a function $\nabla f(\mathbf{x})$ determines the direction of locally steepest ascent, and so the negative gradient $-\nabla f(\mathbf{x})$ is the direction of locally steepest descent. The gradient algorithm is:

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

  where α is the step size (learning rate)

- The method is based on a first-order Taylor series approximation of $f(\mathbf{x})$

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^T (\mathbf{x} - \mathbf{x}_0)$$

- There are two questions: stability and convergence of the algorithm.
- How to choose a good step size α. If α is too large, the iteration will be unstable or barely stable (overshoot the minimum, causing oscillations, or even divergence). If α is too small, the iteration will converge very slowly, making it ineffective for problems.

Gradient Descent Method: Fixed Step Size

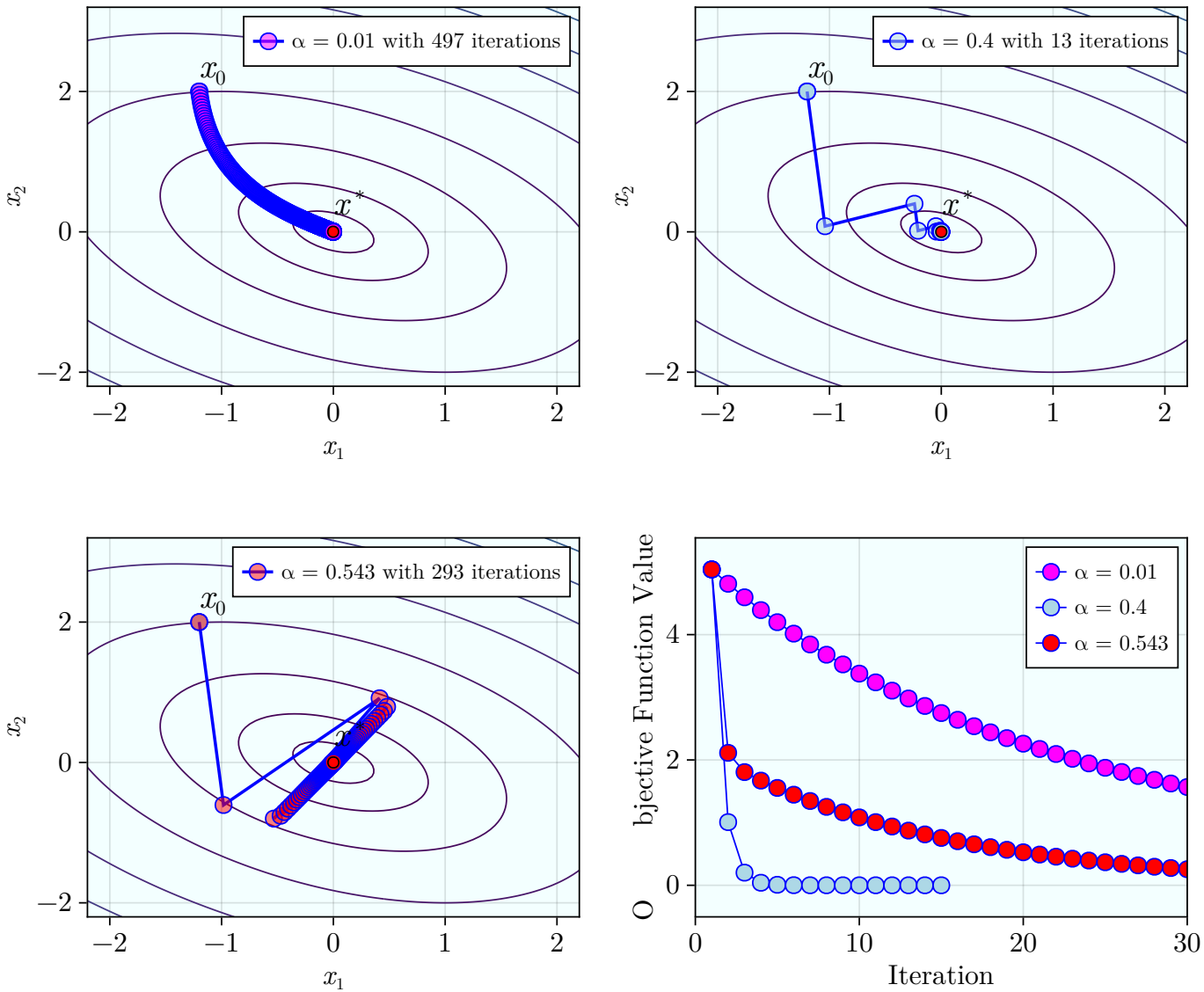
The simplest approach to determine the step-size α is fixed step size.

- A fixed step-size α is good for a convex objective function $f(\mathbf{x})$ with a well-conditioned Hessian $\nabla^2 f(\mathbf{x})$.
- We can show a simple example with quadratic objective to see the effect of the condition number of $\nabla^2 f(\mathbf{x})$ on a fixed-step gradient descent algorithm.
- Fixed-step schemes are simple, but may require tuning for adequate convergence.
- Consider an unconstrained optimization problem:

$$\underset{\mathbf{x}\in\mathbb{R}^2}{\text{minimize}} \quad \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x}, \mathbf{Q} = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$$

  We will solve the problem, using the gradient descent method with $\alpha = 0.01, 0.06, 0.543$.

Gradient Descent Method: Fixed Step Size



Gradient Descent Method: Analysis

- After we have the direction vector $\mathbf{d}_k$ at the point $\mathbf{x}_k$. If we move along $\mathbf{d}_k$ the design variables and the objective function depend only on α as

$$\mathbf{x}(\alpha) = \mathbf{x}_k + \alpha \mathbf{d}_k, \qquad f(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{d}_k)$$

$$\alpha_k = \underset{\alpha}{\text{arg minimize}} \; f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) \quad \Longrightarrow \quad \frac{\partial f(\mathbf{x}_k + \alpha \mathbf{d}_k)}{\partial \alpha_k} = 0$$

The optimization above implies that the directional derivative equals zero:

$$\nabla f(\mathbf{x}_k + \alpha \mathbf{d}_k)^T \mathbf{d}_k = 0, \quad \mathbf{d}_{k+1} = -\nabla f(\mathbf{x}_k + \alpha \mathbf{d}_k)$$

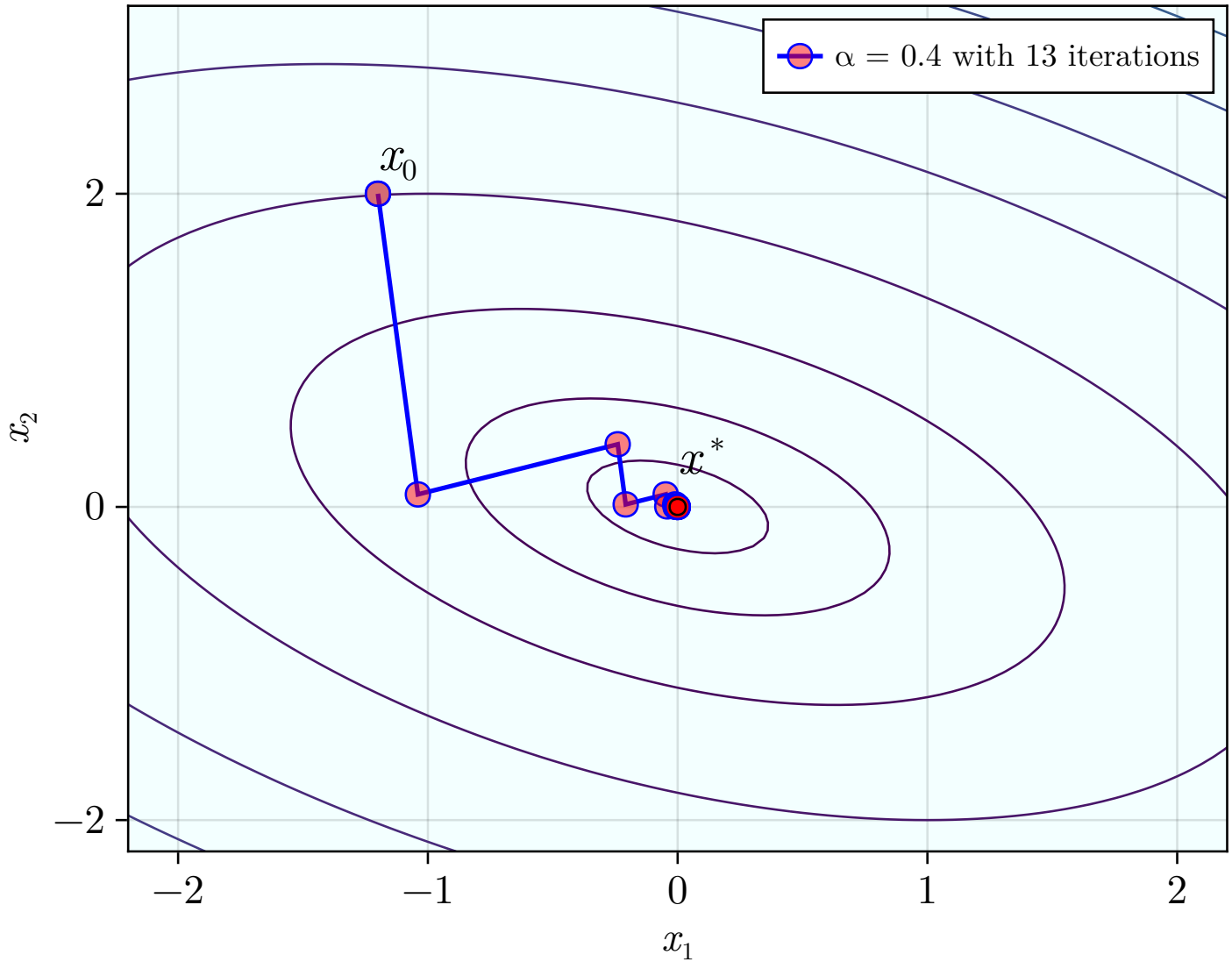
$$\mathbf{d}_{k+1}^T \mathbf{d}_k = 0, \quad \mathbf{d}_{k+1} \text{ and } \mathbf{d}_k \text{ are orthogonal. (source of zig-zags)}$$

- In the steepest descent method, the direction vector is $-\nabla f(\mathbf{x}_k)$ resulting in the slope at the current point $\alpha = 0$ being

$$\left. \frac{df(\alpha)}{d\alpha} \right|_{\alpha=0} = \nabla f(\mathbf{x}_k)^T (-\nabla f(\mathbf{x}_k)) = -\|\nabla f(\mathbf{x}_k)\|^2 < 0$$

Implying a move in a downhill direction.

Gradient Descent Method: Zig-Zags



The Steepest Descent Method: Stoping Criteria

- Starting from an initial point, we determine a direction vector and a step size, and obtain a new point as $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$.
- The question is to know when to stop the iterative process. We have two stop criteria to discuss here.
- **First** Befor performing the step-size, the necessary condition for optimality is checked:

$$\|\nabla f(\mathbf{x}_k)\| \leq \varepsilon_G,$$

  where ϵ_G is a tolerance on the gradient and is supplied by the user. This indicates you are near a stationary point.
- **Second:** We check the successive reductions in f as a criterion for stopping.

$$|f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)| \leq \varepsilon_A + \varepsilon_R |f(\mathbf{x}_k)|$$

  where $\varepsilon_A =$ absolute tolerance on the change in function value and $\varepsilon_R =$ relative tolerance. Only if the condition is satisfied for two consecutive iterations is the descent process stopped.

Gradient Descent Method: Fixed Step Size

Steepest Descent

Require: $\mathbf{x}_0, \varepsilon_G, \varepsilon_A, \varepsilon_R$

$k = 0$, N = max_number

while $k < N$ **do**

Compute $\nabla f(\mathbf{x}_k)$

if $\|\nabla f(x_k)\| \leq \varepsilon_G$ **then**

Stop

else if then

$\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$

end if

$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$,

if $|f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)| \leq \varepsilon_A + \varepsilon_R |f(\mathbf{x}_k)|$ **then**

Stop

else

$k = k + 1$, $\mathbf{x}_k = \mathbf{x}_{k+1}$

end if

end while

Gradient Descent Method: Convergence

Definition: Lipschitz continuity

A function f is Lipschitz continuous with Lipschitz constant L if

$$|f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq L\|\mathbf{x}_1 - \mathbf{x}_2\| \quad \forall \mathbf{x}_1, \mathbf{x}_2$$

- If $f(\mathbf{x})$ is convex and Lipschitz continuous with Lipschitz constant L , then gradient descent with a step size $\alpha = 1/L$ is

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{1}{L}\nabla f(\mathbf{x}_k)$$

- achieves the sublinear rate

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{L}{2k}\|\mathbf{x}_0 - \mathbf{x}^*\|^2, \quad k \geq 1$$

  Thus the convergence rate is $\mathcal{O}(1/k)$, i.e., sublinear in k. (the rate is slower than $\mathcal{O}(k)$)

Gradient Descent Method: Convergence

- If $f(\mathbf{x})$ is strongly convex, the convergence rate is improved. A function is called strongly convex if

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \frac{\mu}{2}\|\mathbf{y}-\mathbf{x}\|^2, \qquad \mathbf{x},\mathbf{y} \in \mathbb{R}^2, \mu > 0$$

  In this case, the convergence rate is:

$$f(\mathbf{x}_k) = f(\mathbf{x}^*) \leq \frac{L}{2}\left(1-\frac{\mu}{L}\right)^k \|\mathbf{x}_0 - \mathbf{x}^*\|^2$$

Gradient Descent Method: Stability

- Consider a quadratic function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x}$, where $\mathbf{Q} \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix. The gradient is

$$\nabla f(\mathbf{x}) = \mathbf{Q}\mathbf{x} \quad \Longrightarrow \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \alpha\mathbf{Q}\mathbf{x}_k = (\mathbf{I} - \alpha\mathbf{Q})\mathbf{x}_k$$

- Then the update rule is simply a discrete-time linear system is

$$\mathbf{x}_{k+1} = (\mathbf{I} - \alpha\mathbf{Q})\mathbf{x}_k = \mathbf{T}^{-1}(\mathbf{I} - \alpha\mathbf{Q})\mathbf{T} = \mathbf{I} - \alpha\mathbf{D}$$

- If $\mathbf{I} - \alpha\mathbf{D}$ is diagonal, then the eigenvalues of it is $1 - \alpha\lambda_i$. From the linear discrete-time system theory, the system is stable if $|1 - \alpha\lambda_i| < 1 \quad \forall i$. This may be rewritten as

$$-1 < 1 - \alpha\lambda_i < 1 \quad \Longrightarrow \quad -2 < -\alpha\lambda_i < 0 \quad \Longrightarrow 0 < \alpha < \frac{2}{\lambda_{\text{max}}}$$

Gradient Descent Method: Optimal Step Size

Now we want to determine an optimal step-size α for fastest convergence of the gradient descent algorithm.

- The optimal step size for gradient descent with a quadratic objective function is determined by the spectral radius of $(\mathbf{I} - \alpha\mathbf{Q})$, which is hte largest absolute eigenvalue:

$$\rho\left(\mathbf{I} - \alpha\mathbf{Q}\right) = \max_i |1 - \alpha\lambda_i|.$$

- To minimize the convergence rate, we choose α such that $\rho\left(\mathbf{I} - \alpha\mathbf{Q}\right)$ is as small as possible. (To get the largest α.)
- We set the optimal value at

$$|1 - \alpha\lambda_{\text{min}}| = |1 - \alpha\lambda_{\text{max}}| \implies 1 - \alpha^*\lambda_{\text{min}} = -(1 - \alpha^*\lambda_{\text{max}})$$

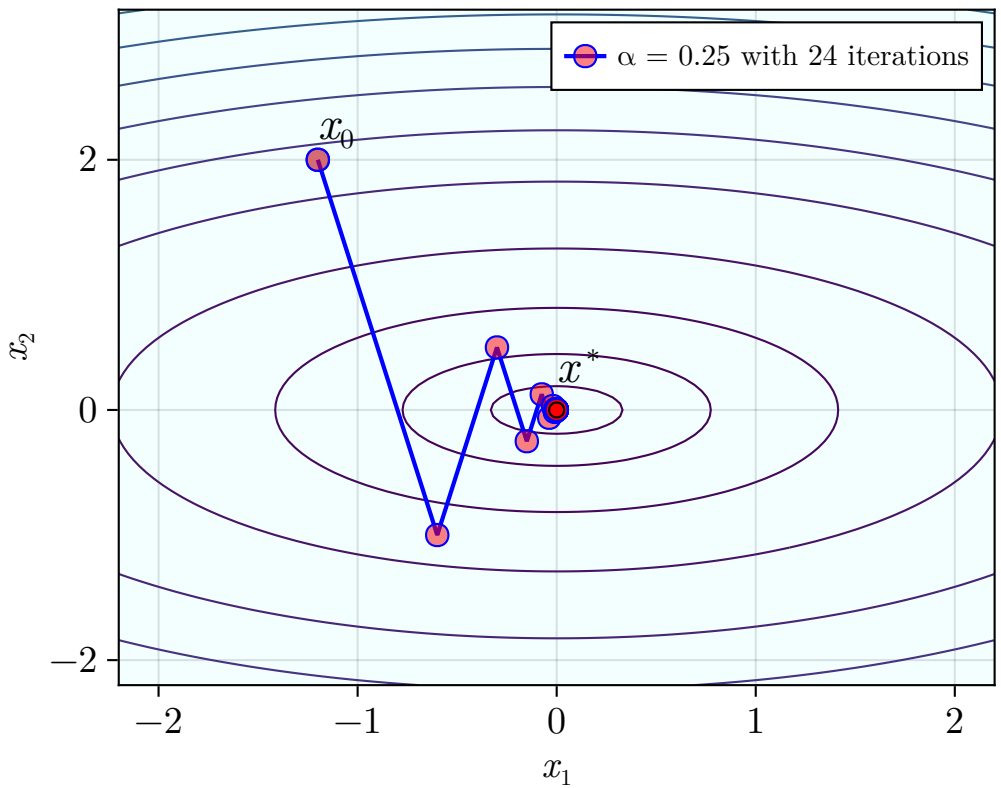
$$\alpha^* = \frac{2}{\lambda_{\text{min}} + \lambda_{\text{max}}}$$

- The best possible performance is limited by the ratio of eigenvalues

$$\kappa = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}}$$

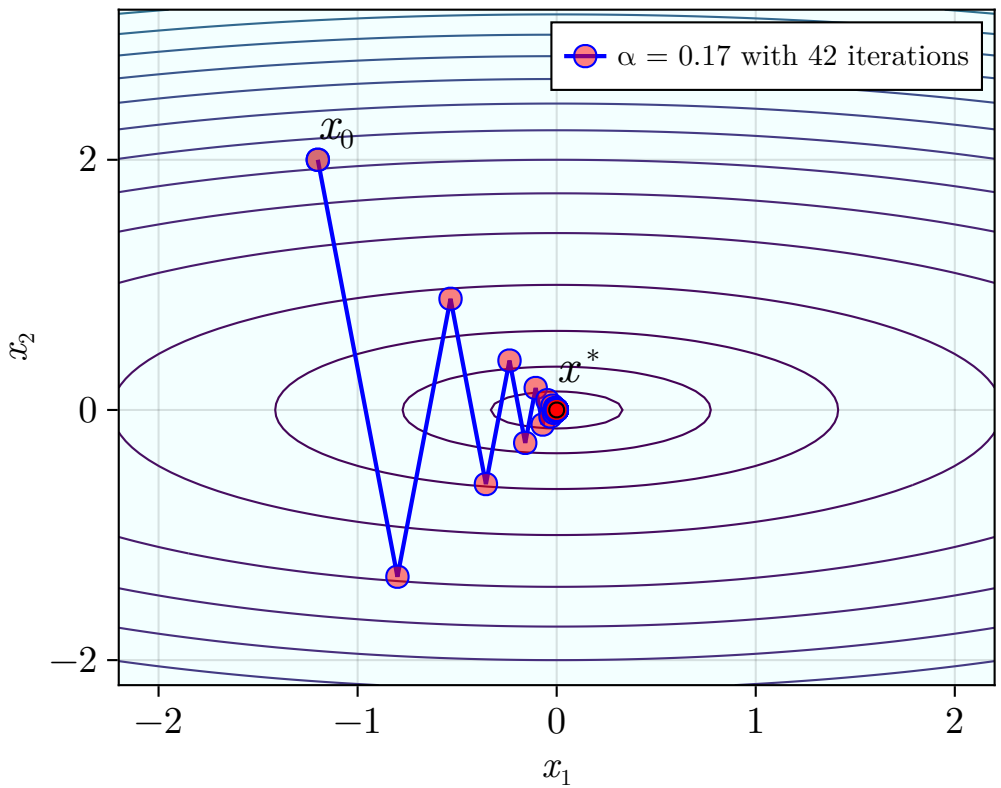
Gradient Descent Method: Optimal Step Size

$f(\mathbf{x}) = \frac{1}{2} x^T \begin{bmatrix} 2 & 0 \\ 0 & 2\gamma \end{bmatrix} x, \gamma = 3$. The Hessian $\nabla^2 f(\mathbf{x})$ is $\begin{bmatrix} 2 & 0 \\ 0 & 6 \end{bmatrix}$, then $\alpha = \frac{2}{8} = 0.25$



Gradient Descent Method: Optimal Step Size

$f(\mathbf{x}) = \frac{1}{2} x^T \begin{bmatrix} 2 & 0 \\ 0 & 2\gamma \end{bmatrix} x, \gamma = 5$. The Hessian $\nabla^2 f(\mathbf{x})$ is $\begin{bmatrix} 2 & 0 \\ 0 & 10 \end{bmatrix}$, then $\alpha = \frac{2}{12} = 0.167$



Gradient Descent Method: Optimal Step Size

Julia Code:

```
function gradient_descent(f,  f, x0; α=0.1, ε_G=1e-6, N=100)
    xgra = zeros(length(x0), N)
    xgra[:, 1] = x0

    for i in 2:N
        xgra[:, i] = xgra[:, i-1] - α *  f(xgra[:, i-1])
        if norm( f(xgra[:, i])) <= ε_G
            return xgra[:, 1:i]
        end
    end
    return xgra
end
#-------------------------------------------------
f(x) = x[1]^2 + 2x[2]^2
 f(x) = [2x[1], 4x[2]]
x0 = [1.0, 1.0]

result = gradient_descent(f,  f, x0)
```

Gradient Descent Method: Optimal Step Size

Matlab Code:

```
f = @(x1, x2) [x1; x2]' *[2 0; 0 2*2]*[x1; x2];   % gamma = 2
gf = matlabFunction(gradient(f(x1,x2),[x1,x2]));

N = 2000; X = zeros(N,2); EG = 1e-10; EA = 1e-10; ER = 1e-10; th0 = 1e-5;

X(1,:) = x0;
for k = 2:N
    % first criteria
    if norm(gf(x0(1), x0(2))) <= EG
        break;
    end
    % transpose to make a column vector
    alpha = 0.3
    % new point
    x0 = x0 - alpha*gf(x0(1), x0(2)))';
    X(k,:) = x0;

    % check second criteria
    nX = abs(fobj(X(k,:)) - fobj(X(k-1,:)));
    if  nX < EA + ER * abs(fobj(X(k-1,:)))
        break;
    end
end
```

Reference

1. Joaquim R. R. A. Martins, Andrew Ning, "*Engineering Design Optimization*," Cambridge University Press, 2021
2. Alexander Mitsos, "*Applied Numerical Optimization*," Lecture Note RWTH AACHEN University
3. Ashok D. Belegundu, Tirupathi R. Chandrupatla, "Optimization Concepts and Applications in Engineering," Cambridge University Press, 2019