Optimization (ND Methods)

What is the optimal solution? (ND)

$$f(\boldsymbol{x}^*) = \min_{\boldsymbol{x}} f(\boldsymbol{x})$$

$f(x)$

$\underset{=}{f}(\underset{\sim}{x})$

(First-order) Necessary condition

1D: $f'(x) = 0$

ND : $\underset{\sim}{\nabla} f(\underset{\sim}{x}^*) = \underset{\sim}{0} \longrightarrow$ gives stationary solution x^*

(Second-order) Sufficient condition

1D: $f''(x) > 0$

ND : $\underset{=}{H}(\underset{-}{x}^*)$ is positive definite $\longrightarrow$ x^* is minimizer

Taking derivatives…

$f: \mathbb{R}^n \longrightarrow \mathbb{R}$

$f(\underset{\sim}{x}) = f(x_1, x_2, \ldots, x_n)$

$\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \cdots \quad \frac{\partial f}{\partial x_n} \Longrightarrow \underbrace{\underset{\sim}{\nabla f}(\underset{\sim}{x})}_{\text{gradient of } f} = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} (n \times 1)$

$\frac{d}{\partial x_i} \underset{\sim}{\nabla f}$

$$\underset{\sim}{H}(\underset{\sim}{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2 \partial x_3} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}_{n \times n}$$

$\Downarrow$ symm!

$(\underset{\sim}{\nabla f})_i = \frac{\partial f}{\partial x_i}$

$(\underset{\sim}{H})_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$

From linear algebra:

$y^T \underline{\underline{H}} y \rightarrow$ scalar

vector

$y \cdot \underline{\underline{H}} y$

A symmetric $n \times n$ matrix $\boldsymbol{H}$ is **positive definite** if $\boldsymbol{y}^T \boldsymbol{H}\, \boldsymbol{y} > \boldsymbol{0}$ for any $\boldsymbol{y} \neq \boldsymbol{0}$

A symmetric $n \times n$ matrix $\boldsymbol{H}$ is **positive semi-definite** if $\boldsymbol{y}^T \boldsymbol{H}\, \boldsymbol{y} \geq \boldsymbol{0}$ for any $\boldsymbol{y} \neq \boldsymbol{0}$

A symmetric $n \times n$ matrix $\boldsymbol{H}$ is **negative definite** if $\boldsymbol{y}^T \boldsymbol{H}\, \boldsymbol{y} < \boldsymbol{0}$ for any $\boldsymbol{y} \neq \boldsymbol{0}$

A symmetric $n \times n$ matrix $\boldsymbol{H}$ is **negative semi-definite** if $\boldsymbol{y}^T \boldsymbol{H}\, \boldsymbol{y} \leq \boldsymbol{0}$ for any $\boldsymbol{y} \neq \boldsymbol{0}$

A symmetric $n \times n$ matrix $\boldsymbol{H}$ that is not negative semi-definite and not positive semi-definite is called **indefinite**

$y \cdot H y$:

- $> 0 \rightarrow$ PD
- $\geq 0 \rightarrow$ PSD
- $< 0 \rightarrow$ ND
- $\leq 0 \rightarrow$ NSD

$$\left.\begin{array}{l} y \rightarrow > 0 \\ y \rightarrow < 0 \end{array}\right\} \mathrm{I}$$

la.eig(H)

$$f(\boldsymbol{x}^*) = \min_{\boldsymbol{x}} f(\boldsymbol{x})$$

First order necessary condition: $\boldsymbol{\nabla} f(\boldsymbol{x}) = \mathbf{0}$

Second order sufficient condition: $\boldsymbol{H}(\boldsymbol{x})$ **is positive definite**

How can we find out if the Hessian is positive definite?

$\boxed{\underline{\underline{H}}\, y = \lambda y} \longrightarrow (\lambda, y) \longrightarrow$ are eigenpairs of H

$y^T H y = \lambda y^T y = \lambda \|y\|_2^2 \Longrightarrow \boxed{\lambda = \dfrac{y^T H y}{\|y\|_2^2}}$ → always positive

* $\lambda_i > 0 \quad \forall i \Rightarrow y^T H y > 0 \quad \forall y \Longrightarrow$ H is pos. def $\Rightarrow x^*$ is minimizer

* $\lambda_i < 0 \quad \forall i \Rightarrow y^T H y < 0 \quad \forall y \Longrightarrow$ H is neg def $\Rightarrow x^*$ is maximizer

* $\left.\begin{matrix}\lambda_i > 0 \\ \lambda_i < 0\end{matrix}\right\} \longrightarrow$ H is indefinite $\longrightarrow x^*$ is saddle point

Types of optimization problems

$$f(\boldsymbol{x}^*) = \min_{\boldsymbol{x}} f(\boldsymbol{x})$$

f: nonlinear, continuous and smooth

Gradient-free methods

Evaluate $f(\boldsymbol{x})$

Gradient (first-derivative) methods

Evaluate $f(\boldsymbol{x}), \boldsymbol{\nabla} f(\boldsymbol{x})$

Second-derivative methods

Evaluate $f(\boldsymbol{x}), \boldsymbol{\nabla} f(\boldsymbol{x}), \boldsymbol{\nabla}^2 f(\boldsymbol{x}) = \underline{\underline{H}}(\underline{x})$

Example (ND)

Consider the function $f(x_1, x_2) = 2x_1^3 + 4x_2^2 + 2x_2 - 24x_1$

Find the stationary point and check the sufficient condition

$$\nabla f = \begin{bmatrix} 6x_1^2 - 24 \\ 8x_2 + 2 \end{bmatrix}; \quad H = \begin{bmatrix} 12x_1 & 0 \\ 0 & 8 \end{bmatrix}$$

1) $\nabla f = 0 \Longrightarrow \begin{bmatrix} 6x_1^2 - 24 \\ 8x_2 + 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Longrightarrow$ $6x_1^2 = 24 \rightarrow x_1^2 = 4 \rightarrow x_1 = \pm 2$

$8x_2 = -2 \Longrightarrow x_2 = -0.25$

stationary points: $x^* = \begin{bmatrix} +2 \\ -0.25 \end{bmatrix}$ $\quad x^* = \begin{bmatrix} -2 \\ -0.25 \end{bmatrix}$

2) $H\begin{pmatrix} -2 \\ -0.25 \end{pmatrix} = \begin{bmatrix} -24 & 0 \\ 0 & 8 \end{bmatrix} \Rightarrow$ indefinite $\Downarrow$ saddle point

$H\begin{pmatrix} 2 \\ -0.25 \end{pmatrix} = \begin{bmatrix} 24 & 0 \\ 0 & 8 \end{bmatrix}$ pos. def. $\Downarrow$ Minimizer!

Optimization in ND: Steepest Descent Method

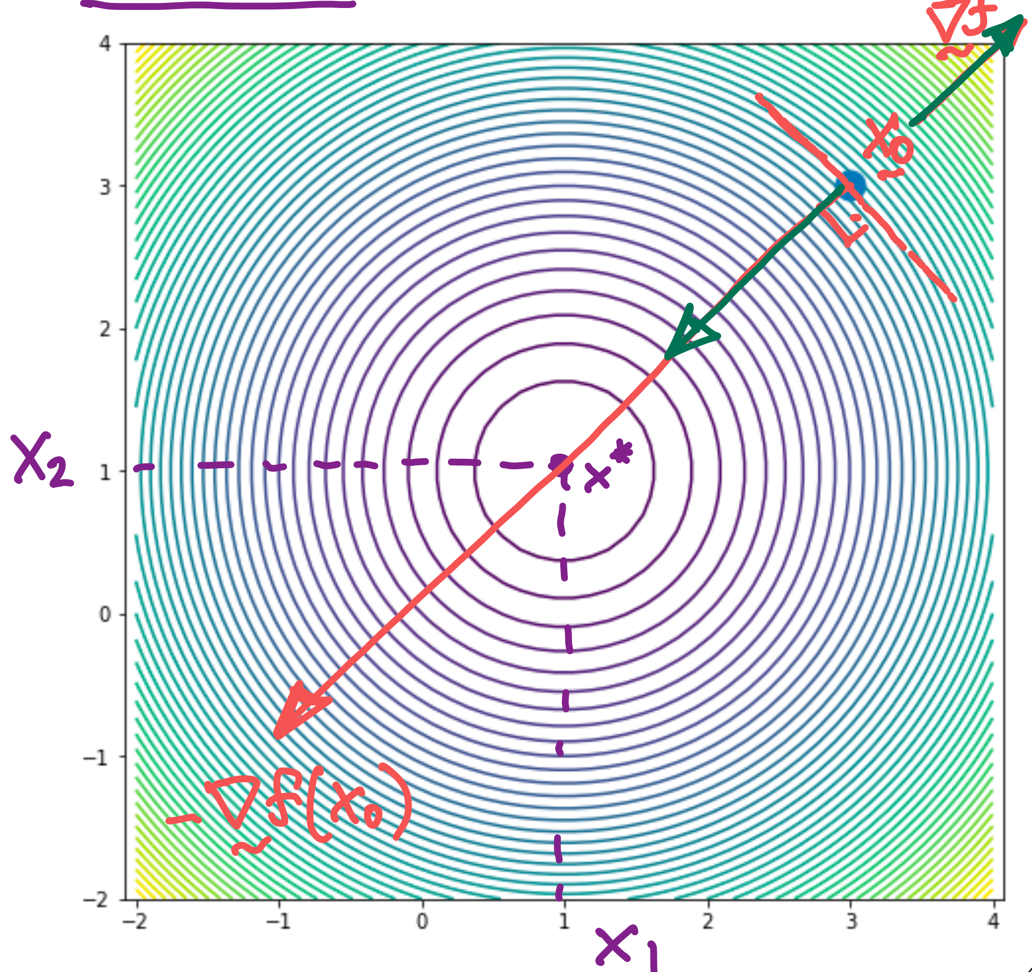
$\min_{x} f(x)$

$\boxed{-\nabla f}$

Given a function $f(\boldsymbol{x}): \mathcal{R}^n \rightarrow \mathcal{R}$ at a point $\boldsymbol{x}$, the function will decrease its value in the direction of steepest descent: $-\boldsymbol{\nabla} f(\boldsymbol{x})$

What is the steepest descent direction?

$$f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2$$



Steepest Descent Method

$\underset{\sim}{x}_2 = \underset{\sim}{x}_1 - \nabla f(\underset{\sim}{x}_1)$

Start with initial guess:

$$\boldsymbol{x}_0 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

Check the update:

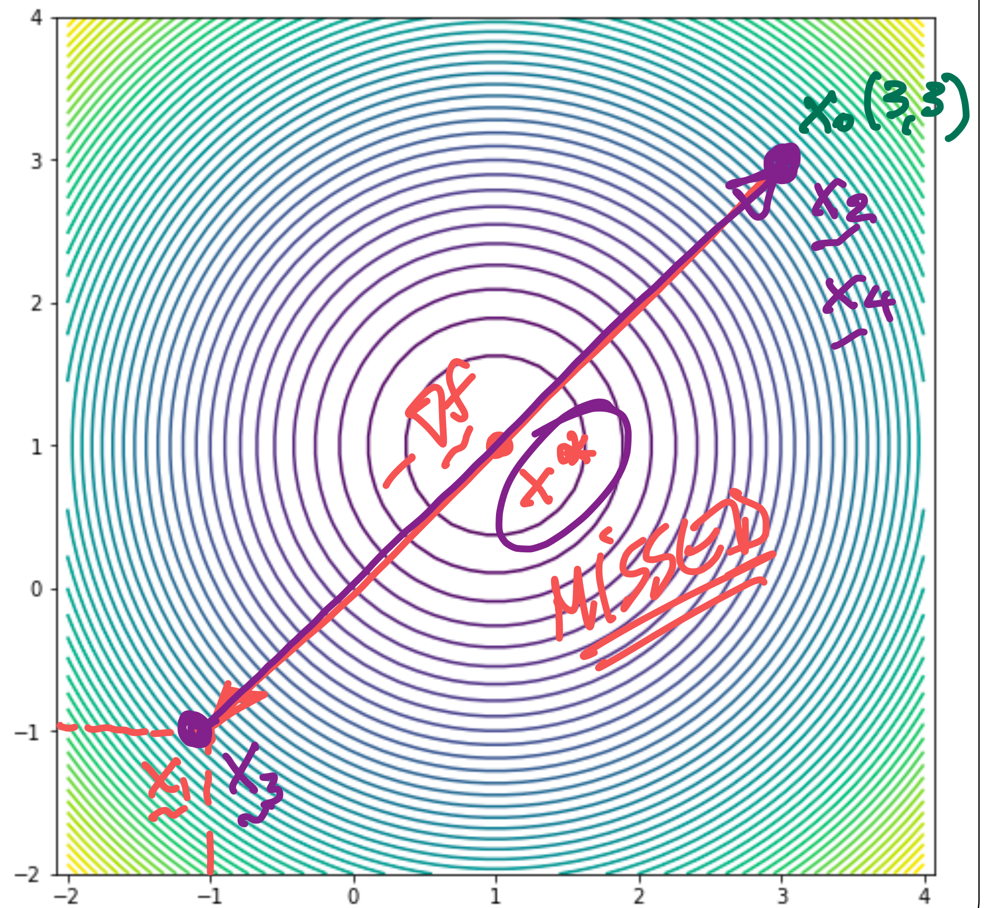
$$\boxed{\underset{\sim}{x}_1 = \underset{\sim}{x}_0 - \underset{\sim}{\nabla f}(\underset{\sim}{x}_0)}$$

$$\underset{\sim}{\nabla f} = \begin{bmatrix} 2(x_1 - 1) \\ 2(x_2 - 1) \end{bmatrix}$$

$$\underset{\sim}{\nabla f}(\underset{\sim}{x}_0) = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

$$\underset{\sim}{x}_1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix} - \begin{bmatrix} 4 \\ 4 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$$f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2$$



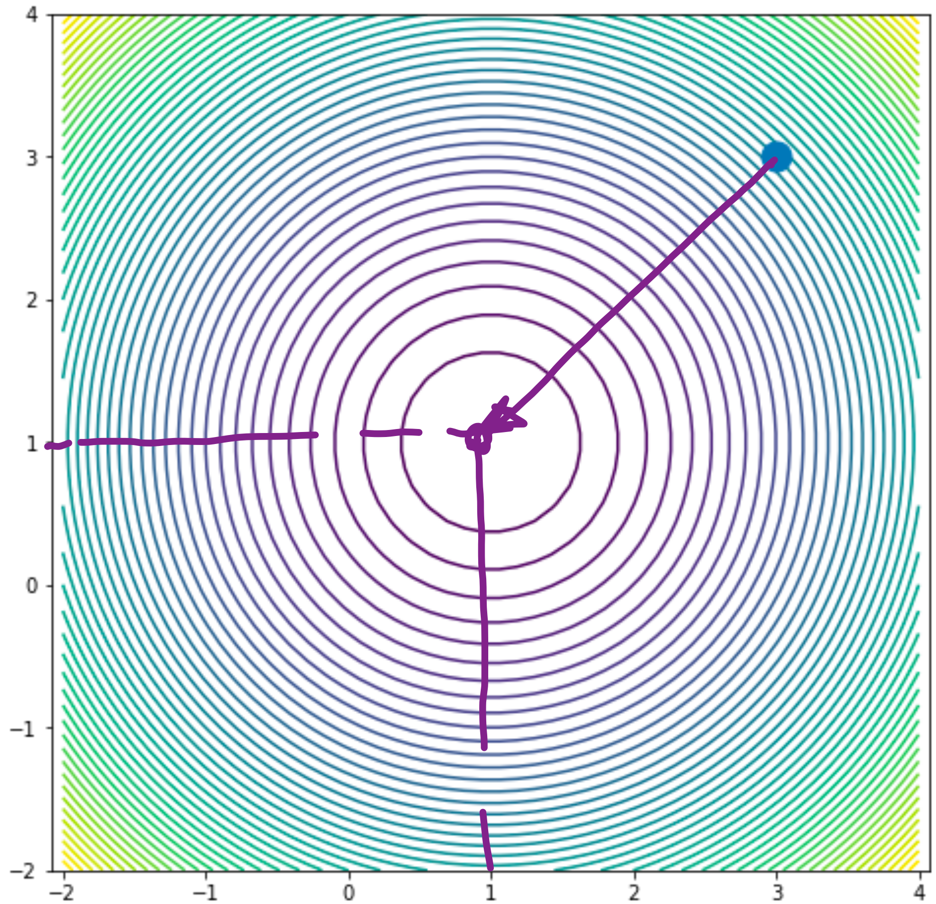
Steepest Descent Method

Update the variable with:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \underline{\alpha_k} \nabla f(\boldsymbol{x}_k)$$

How far along the gradient should we go? What is the "best size" for α_k?

$$f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2$$

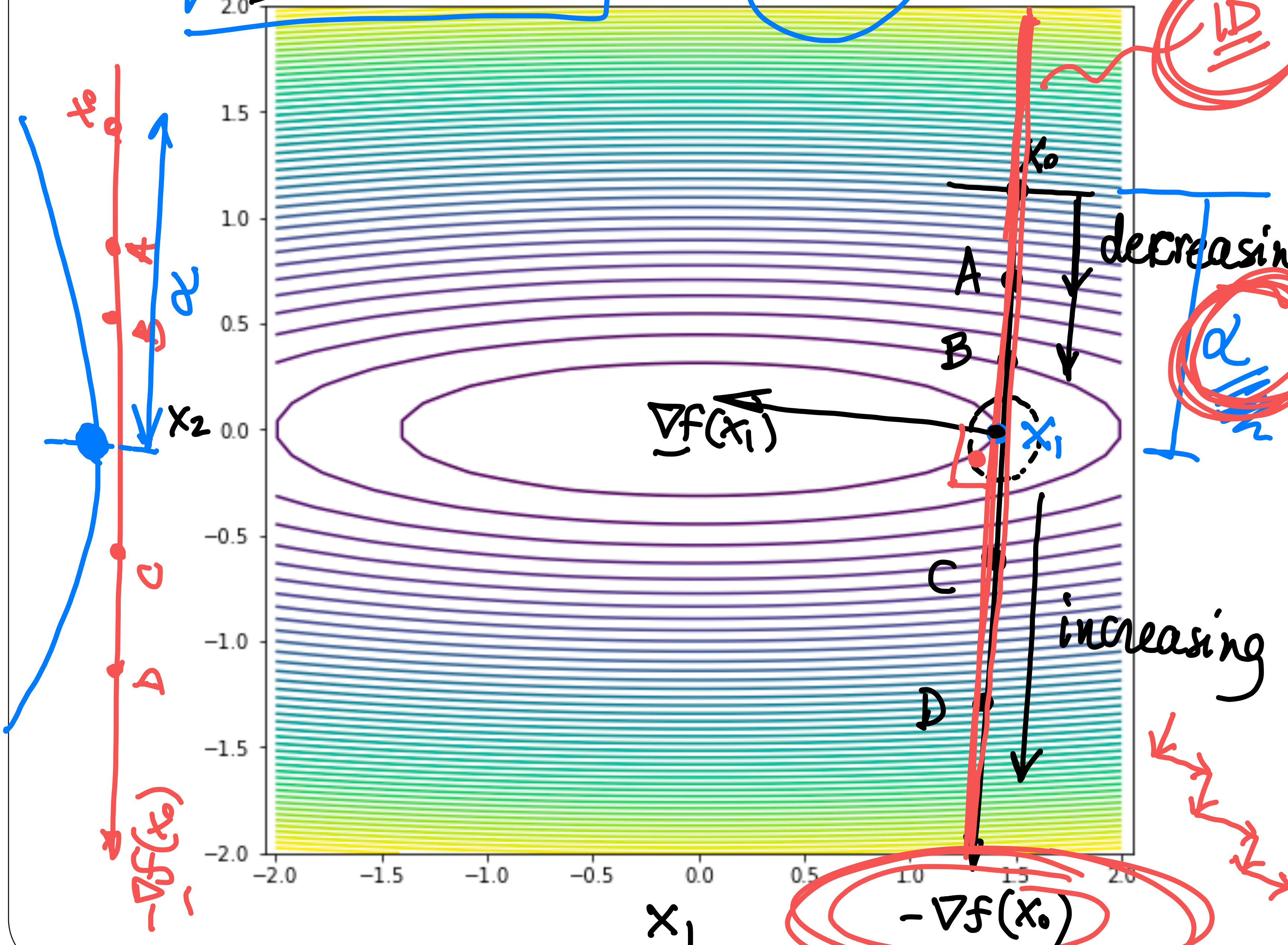


$$\underset{\sim}{x}_1 = \underset{\sim}{x}_0 - \underline{0.5} \nabla f(x_0)$$

$$\boxed{\alpha = 0.5}$$

How can we get α?

Find $\boxed{x_1 = x_0 - \alpha \nabla f(x_0)}$ s.t $f(x_1)$ is minimized



Steepest Descent Method

Algorithm:

1D optimization problem

Initial guess: $\boldsymbol{x}_0$

several fc eval.

Evaluate: $\boldsymbol{s}_k = -\nabla f(\boldsymbol{x}_k)$

Perform a line search to obtain α_k (for example, Golden Section Search)

$$\boxed{\alpha_k} = \underset{\alpha}{\operatorname{argmin}}\, f(\underbrace{\boldsymbol{x}_k + \alpha\, \boldsymbol{s}_k}_{x_{k+1} = x_k + \alpha s_k})$$

Update: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\, \boldsymbol{s}_k$

Line Search

$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$

we want to find α_k s.t.

$f(x_{k+1})$

$$\min_{\boxed{\alpha}} f(\underbrace{x_k - \boxed{\alpha_k} \nabla f(x_k)}_{x_{k+1}})$$

1st order condition $\frac{df}{d\alpha} = 0 \rightarrow$ gives α

$$\frac{df}{d\alpha} = \frac{\partial f}{\partial x_{k+1}} \frac{\partial x_{k+1}}{\partial \alpha} = \nabla f(x_{k+1}) \cdot \nabla f(x_k) = 0$$

$$\nabla f(x_{k+1}) \cdot \nabla f(x_k) = 0$$

$\boxed{\nabla f(x_{k+1}) \text{ is orthogonal to } \nabla f(x_k)}$

zig-zag pattern convergence.

Example

$$\min_{x_1, x_2} f(x_1, x_2)$$

Consider minimizing the function

$$f(x_1, x_2) = 10(x_1)^3 - (x_2)^2 + x_1 - 1$$

Given the initial guess

$$x_1 = 2,\ x_2 = 2$$

$$\underset{\sim}{x_0} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

what is the direction of the first step of gradient descent?

$$\nabla f = \begin{bmatrix} 30x_1^2 + 1 \\ -2x_2 \end{bmatrix} \qquad \nabla f(\underset{\sim}{x_0}) = \begin{bmatrix} 121 \\ -4 \end{bmatrix}$$

$$\boxed{\text{steepest descent direction} \Longrightarrow \begin{bmatrix} -121 \\ +4 \end{bmatrix}}$$

Newton's Method

Using Taylor Expansion, we build the approximation:

$$\underbrace{f(\underset{\sim}{x}+\underset{\sim}{s})}_{\text{nonlinear}} = \underbrace{f(\underset{\sim}{x}) + \nabla f(\underset{\sim}{x})^T \underset{\sim}{s} + \frac{1}{2}\underset{\sim}{s}^T \underset{=}{H} \underset{\sim}{s}} = \hat{f}(\underset{\sim}{s})$$

quadratic approx of f

1st order condition: $\nabla_s \hat{f} = 0$

$$\boxed{\nabla f(\underset{\sim}{x}) + \underset{=}{H}\,\underset{\sim}{s} = 0}$$

→ $\underset{=}{H}$ is symmetric

$\underset{=}{H} = \underset{=}{H}^T$

$$\boxed{\underset{=}{H}(\underset{\sim}{x})\,\underset{\sim}{s} = -\nabla f(\underset{\sim}{x})}$$

→ solve linsys to find Newton step $\underset{\sim}{s}$

Newton's Method

$H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$ → $O(n^2)$

Algorithm:

Initial guess: $\boldsymbol{x}_0$

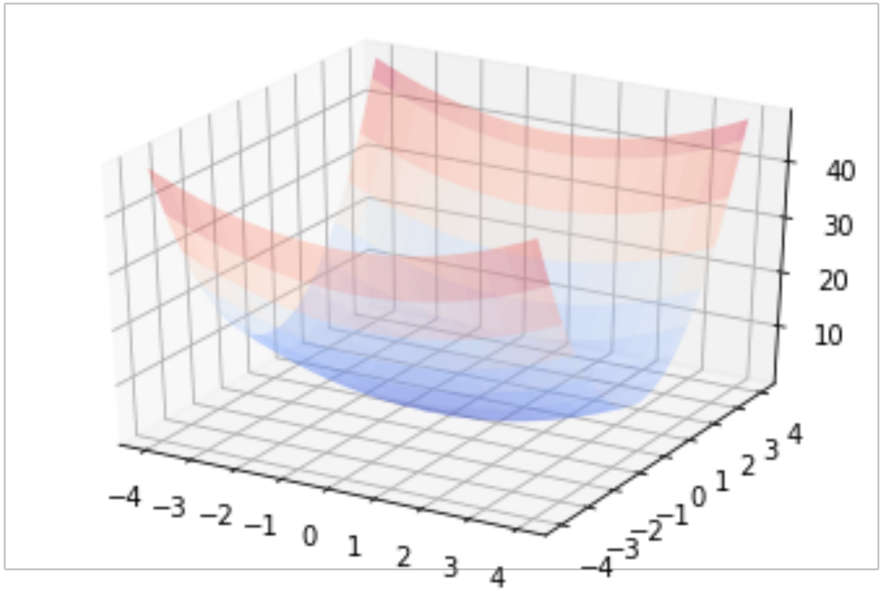
Solve: $\boxed{\boldsymbol{H}_f(\boldsymbol{x}_k)}\, \boldsymbol{s}_k = -\nabla f(\boldsymbol{x}_k)$ → solve $O(n^3)$ $\underset{\sim}{s}$

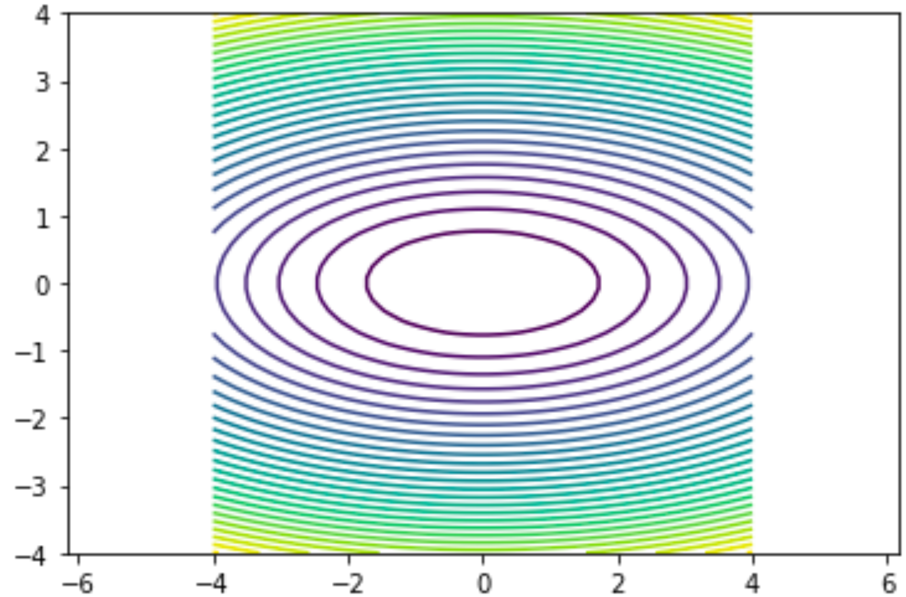
Update: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$

Note that the Hessian is related to the curvature and therefore contains the information about how large the step should be.

Try this out!

$$f(x, y) = 0.5x^2 + 2.5y^2$$





When using the Newton's Method to find the minimizer of this function, estimate the number of iterations it would take for convergence?

A) 1 B) 2-5 C) 5-10 D) More than 10 E) Depends on the initial guess

Newton's Method Summary

Algorithm:

Initial guess: $\boldsymbol{x}_0$

Solve: $\boldsymbol{H_f}(\boldsymbol{x}_k)\, \boldsymbol{s}_k = -\boldsymbol{\nabla} f(\boldsymbol{x}_k)$

Update: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$

About the method…

- Typical quadratic convergence ☺
- Need second derivatives ☹
- Local convergence (start guess close to solution)
- Works poorly when Hessian is nearly indefinite
- Cost per iteration: $O(n^3)$