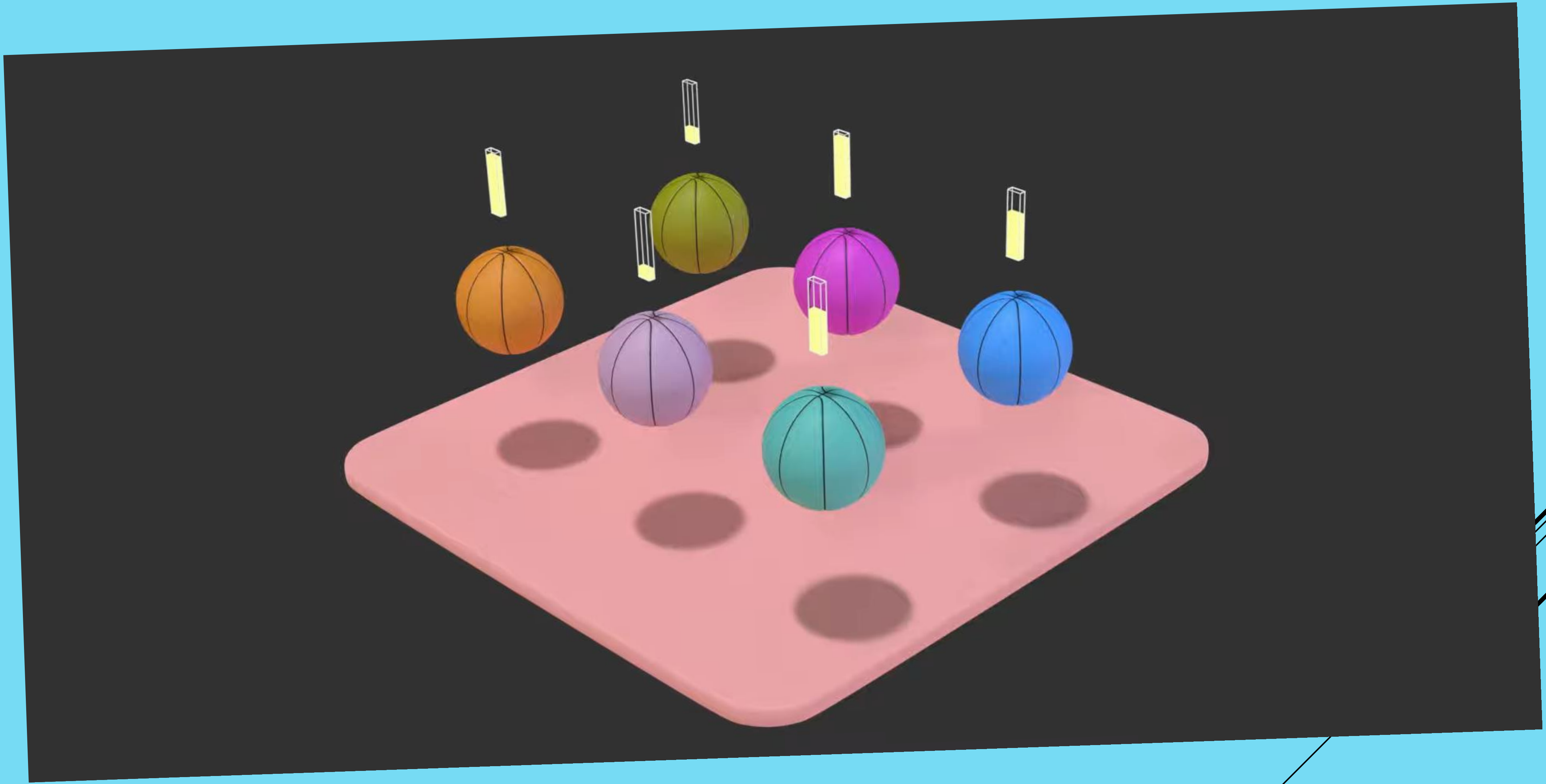
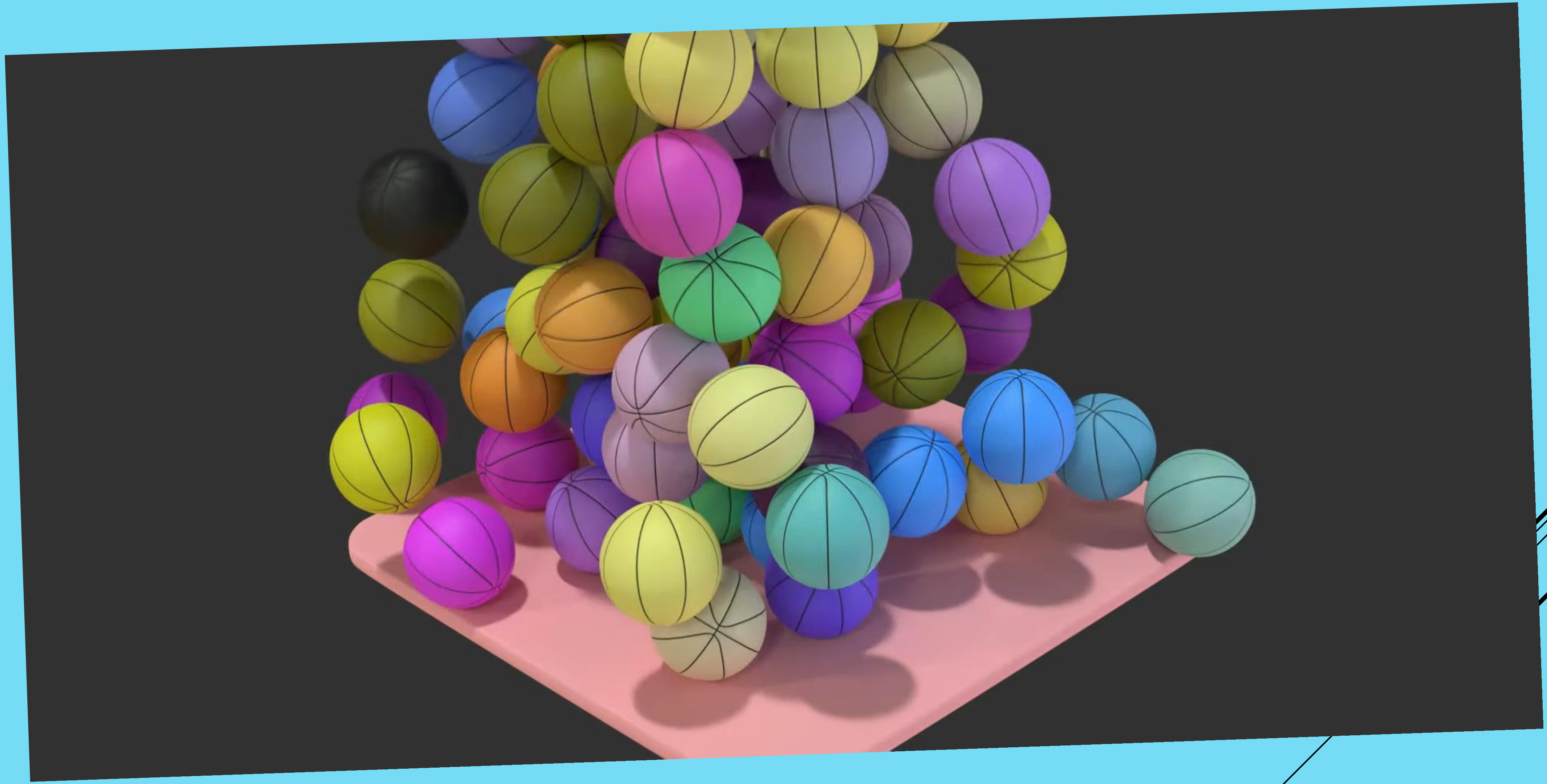
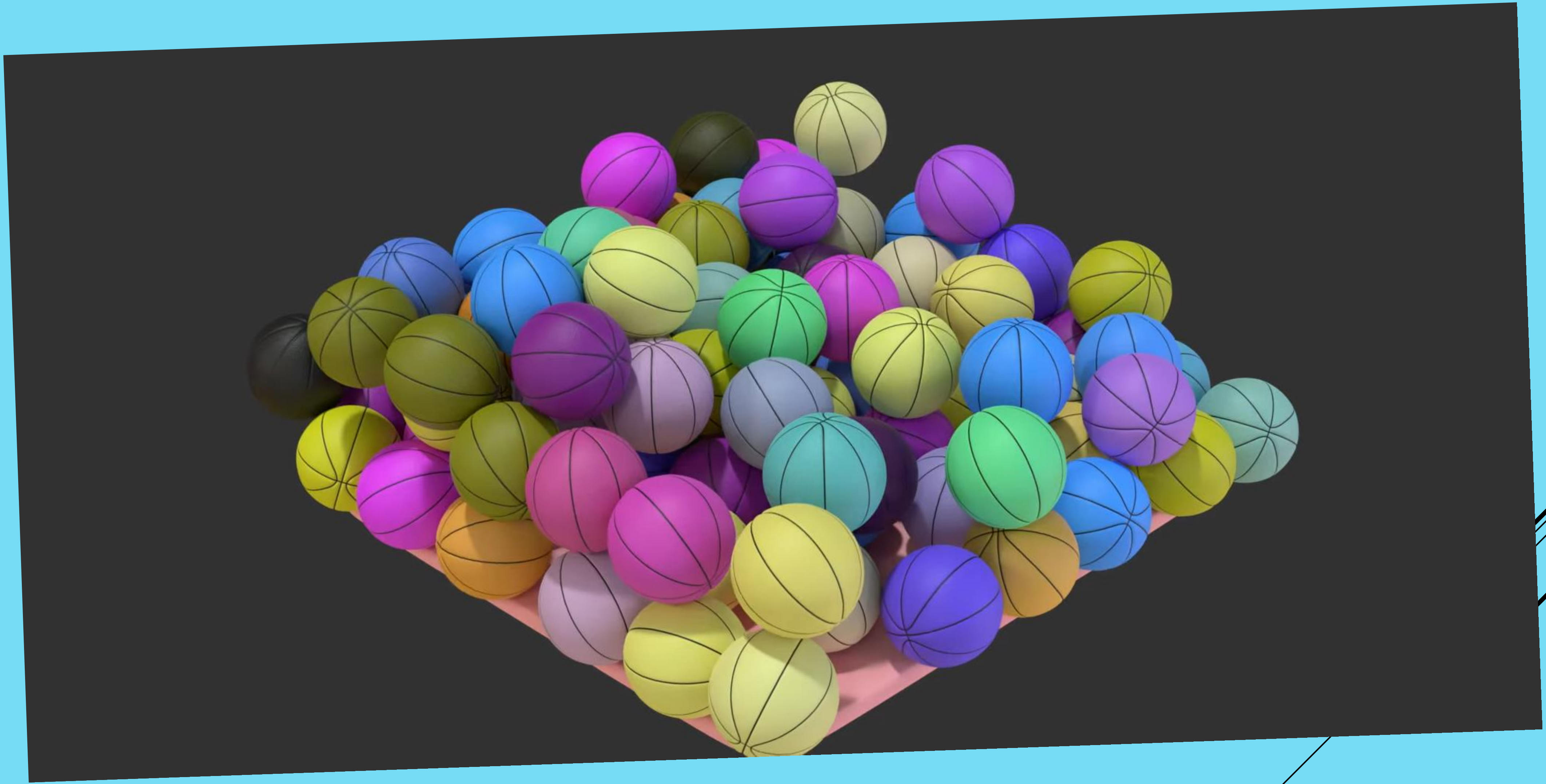
OPTIMIZATION FOR DATA SCIENCE

Dr. Syed Inayatullah

WHAT IS OPTIMIZATION







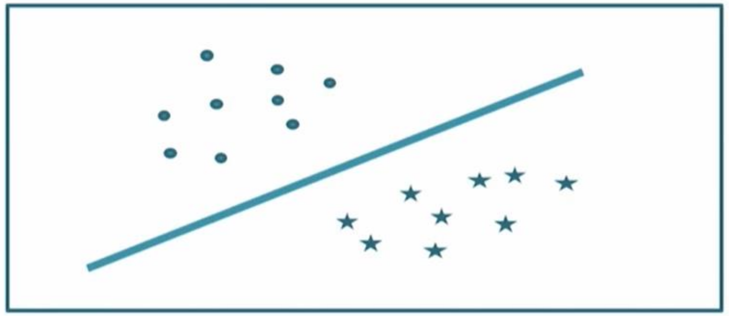
USE OF OPTIMIZATION IN DATA SCIENCE

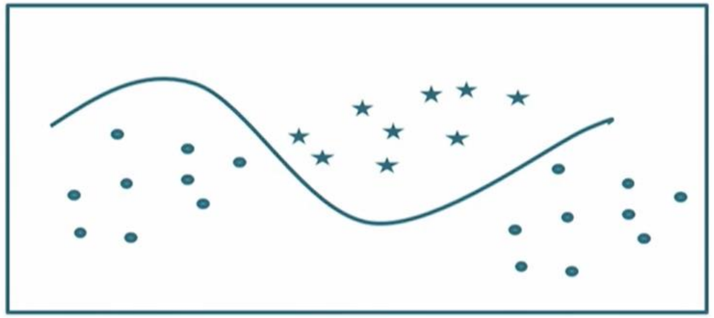
Techniques

- Regression analysis
- K-nearest-neighbor
- K-means clustering
- Logistics regression
- Principal Component Analysis
- Predictive Modeling
  - Lasso, Elastic net

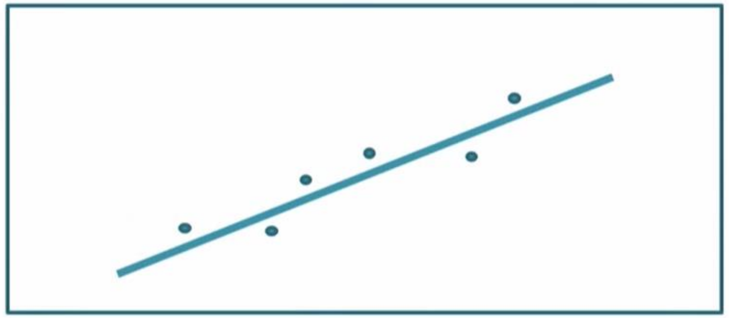
Types of Problems

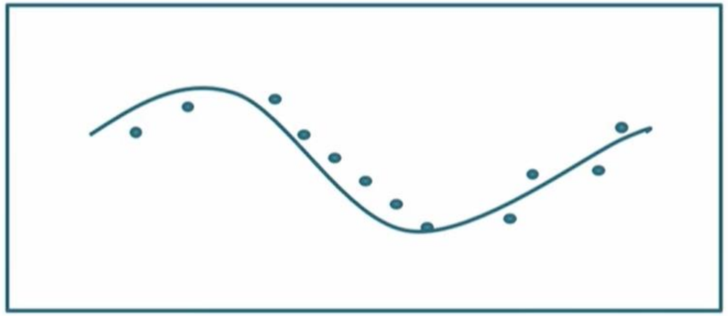
- Classification problems



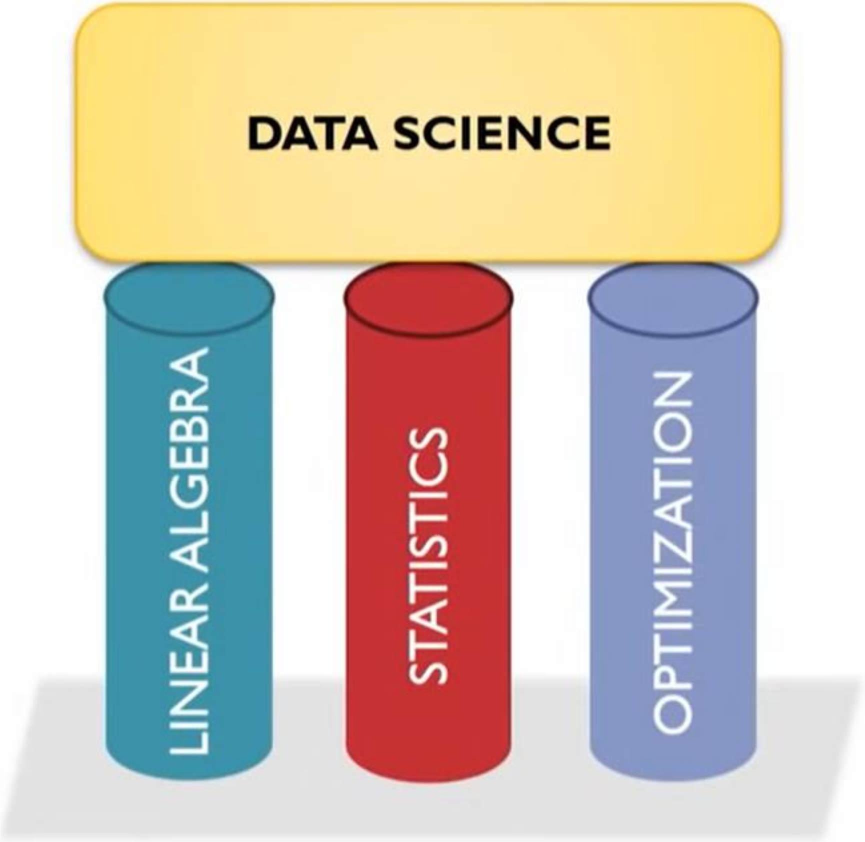


- Function approximation





Three pillars of data science



Fundamentals of optimization

What is optimization ?

"An optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function."*

What is optimization?

- … the use of specific methods to determine the "best" solution to a problem
  - Find the best functional representation for data
  - Find the best hyperplane to classify data

Why optimization for machine learning

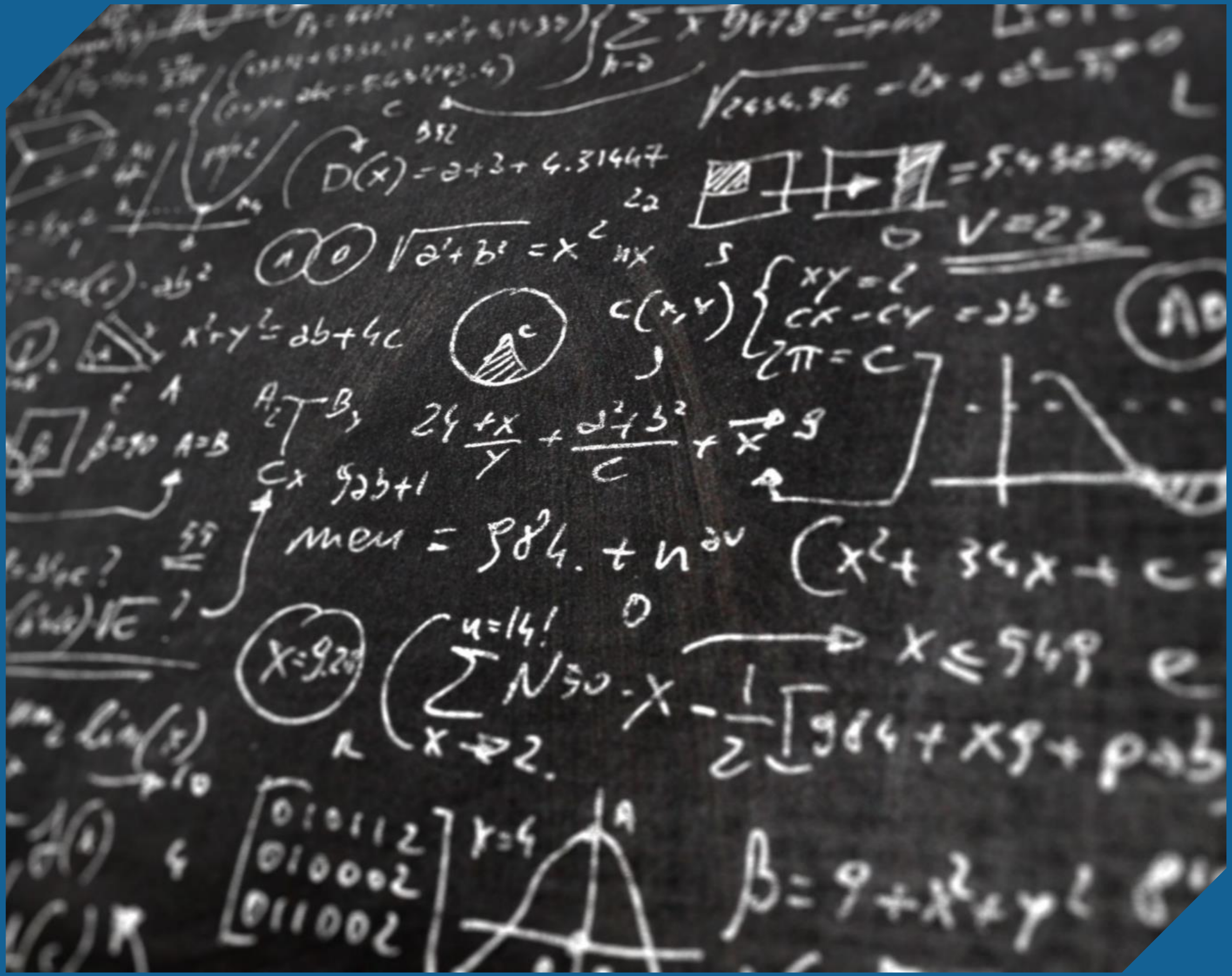
- (Almost) All machine learning (ML) algorithms can be viewed as solutions to optimization problems
  - Even in cases where, the original machine learning technique has a basis derived from other fields
- A basic understanding of optimization approaches help in
  - More deeply understand the working of the ML algorithm
  - Rationalize the workings of the algorithm
  - And (may be !!!), develop new algorithms ourselves

Components of an optimization problem

- Objective function
  - We look at minimization problem
- Decision variables
- Constraints

Types of optimization problems

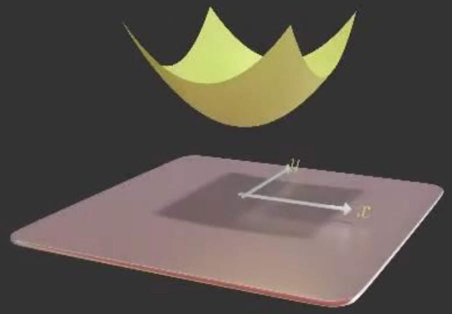
- Depending on the type of objective function, constraints and decision variables
  - Linear programming problem
  - Nonlinear programming problem
    - Convex vs Non-convex
  - Integer programming problem (linear and nonlinear)
  - Mixed integer linear programming problem
  - Mixed integer nonlinear programming problem



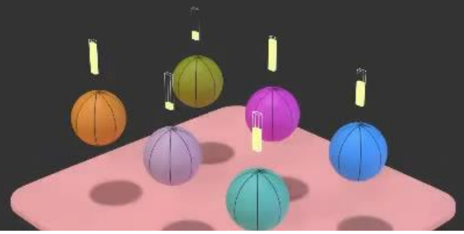
DEFINITION OF CONVEXITY

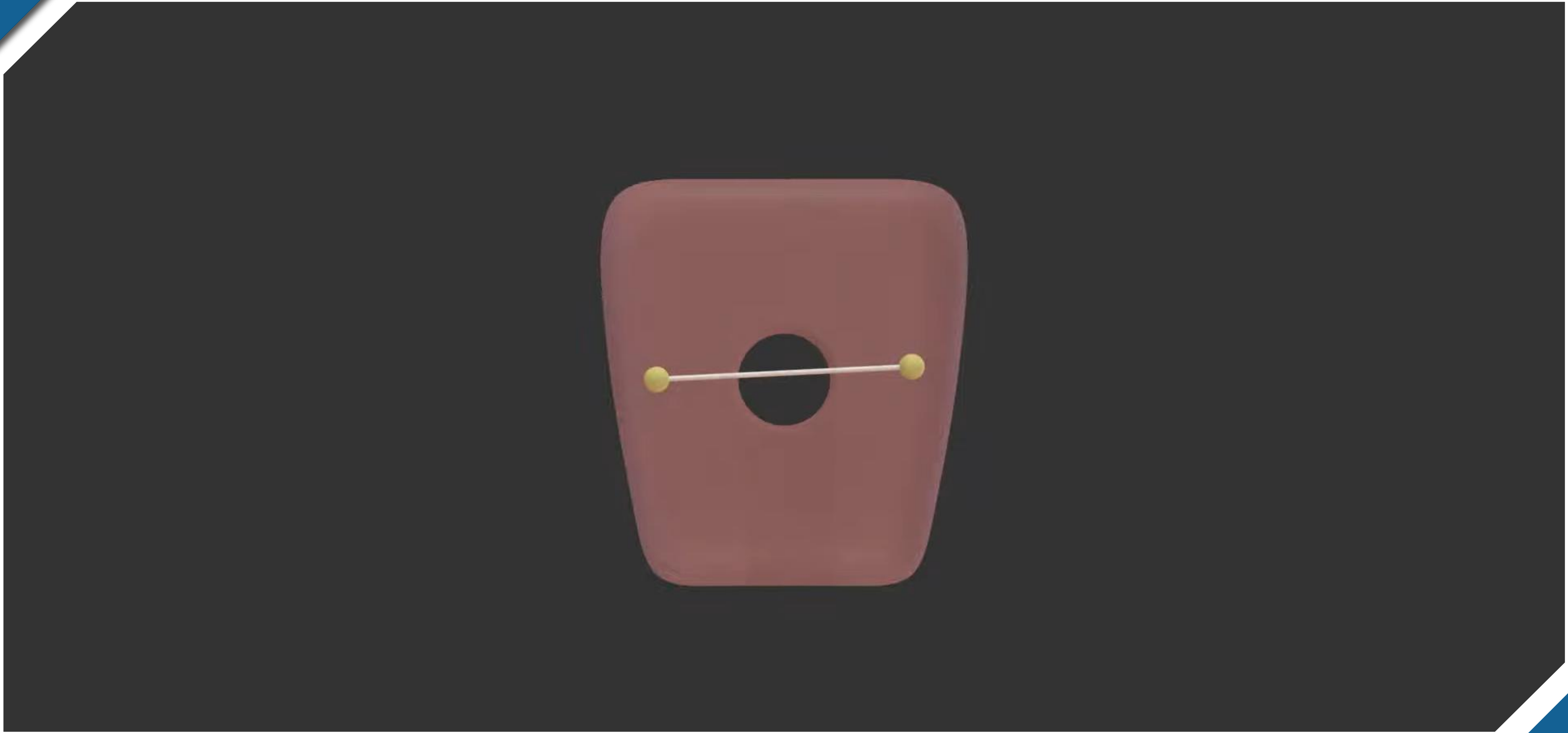
Sets

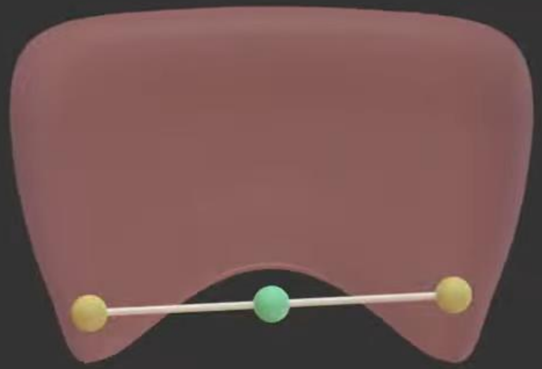
Functions

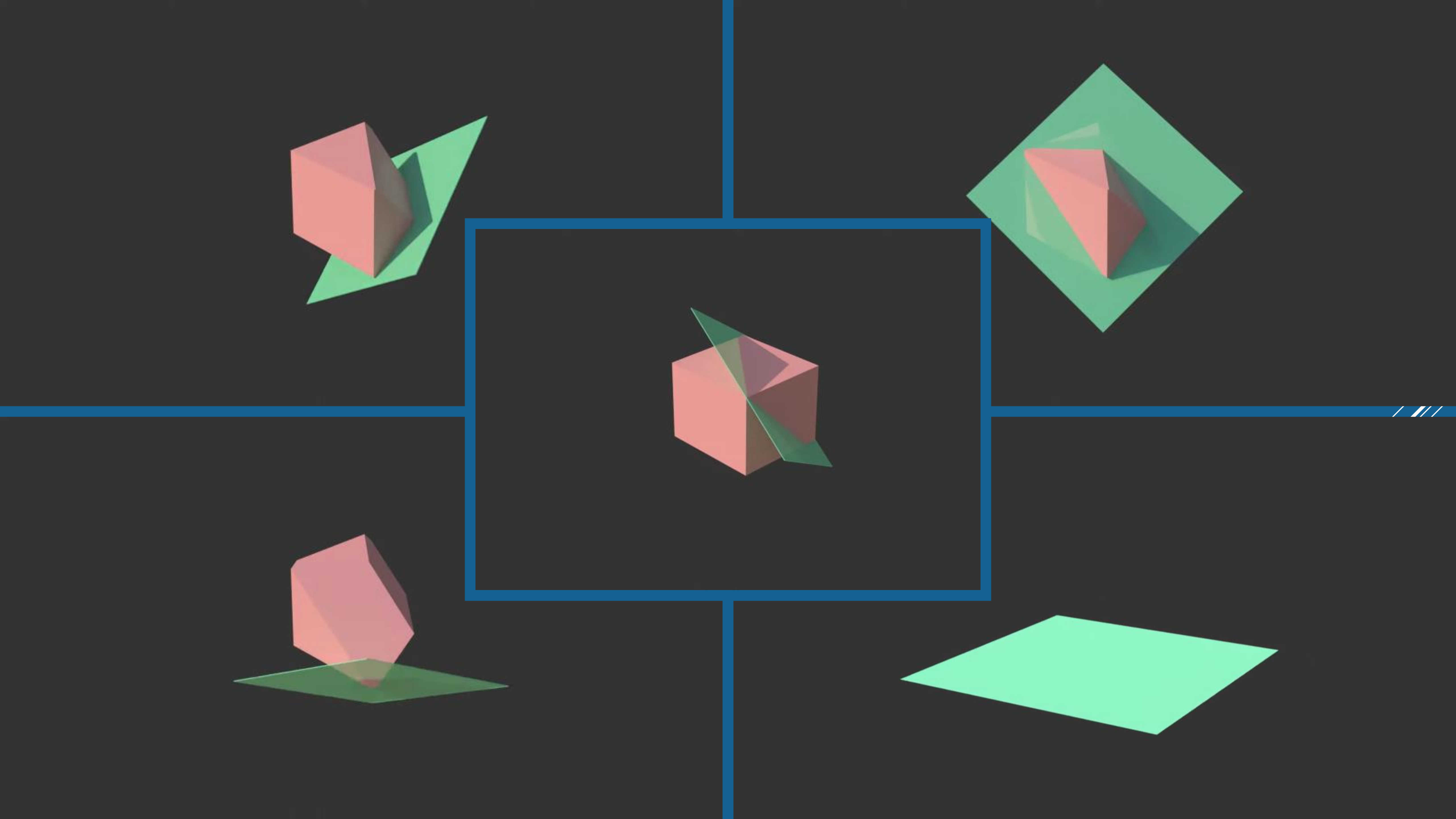


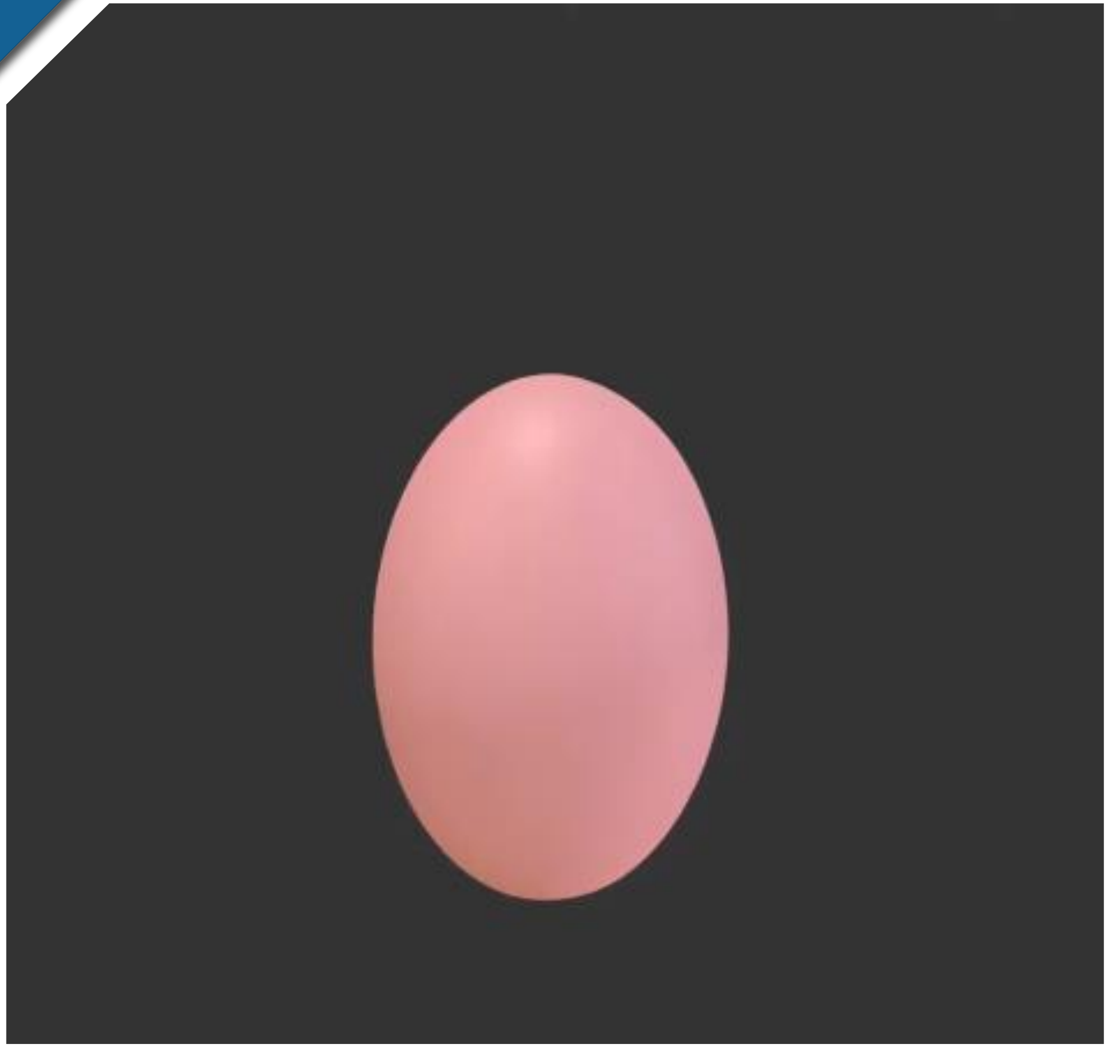
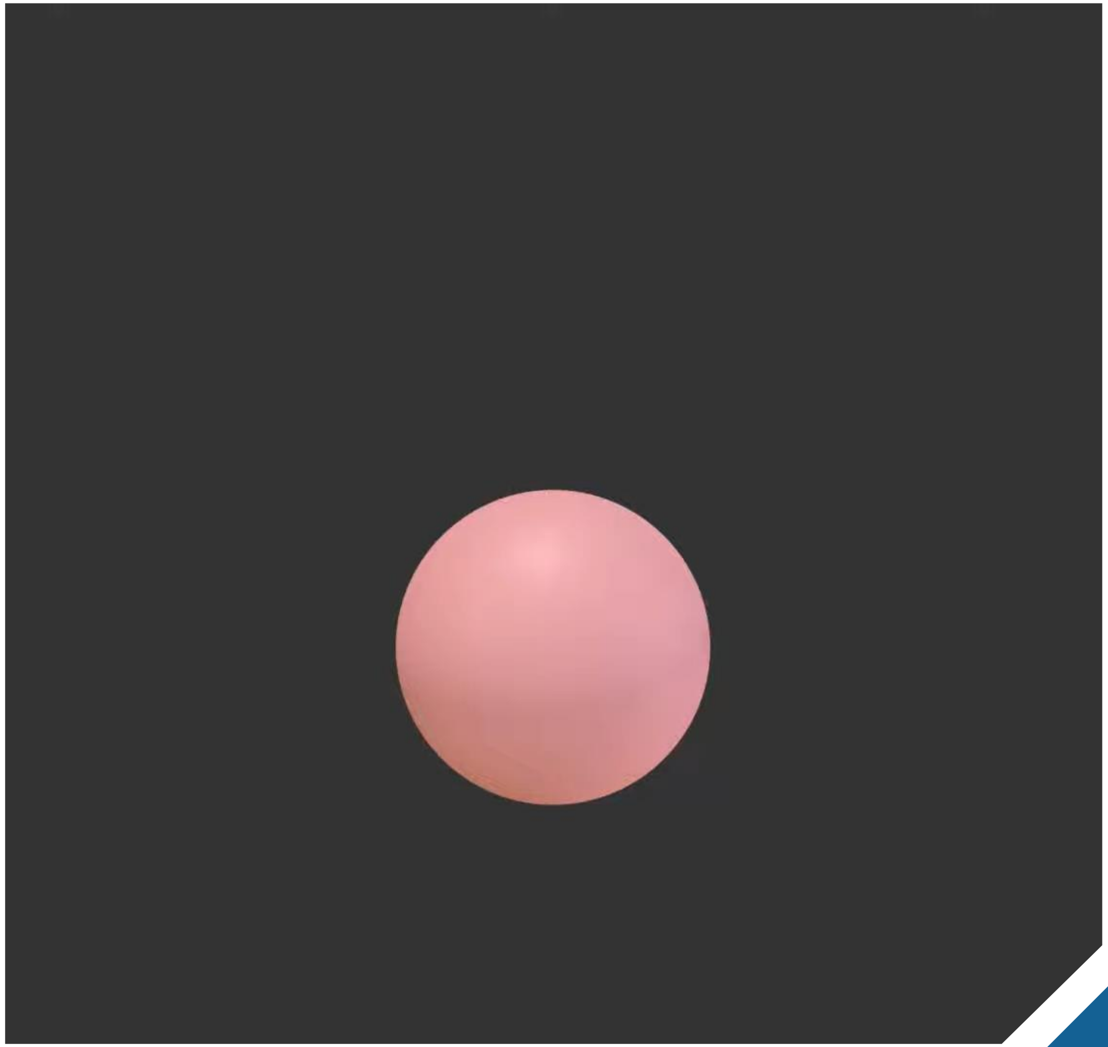
Optimization

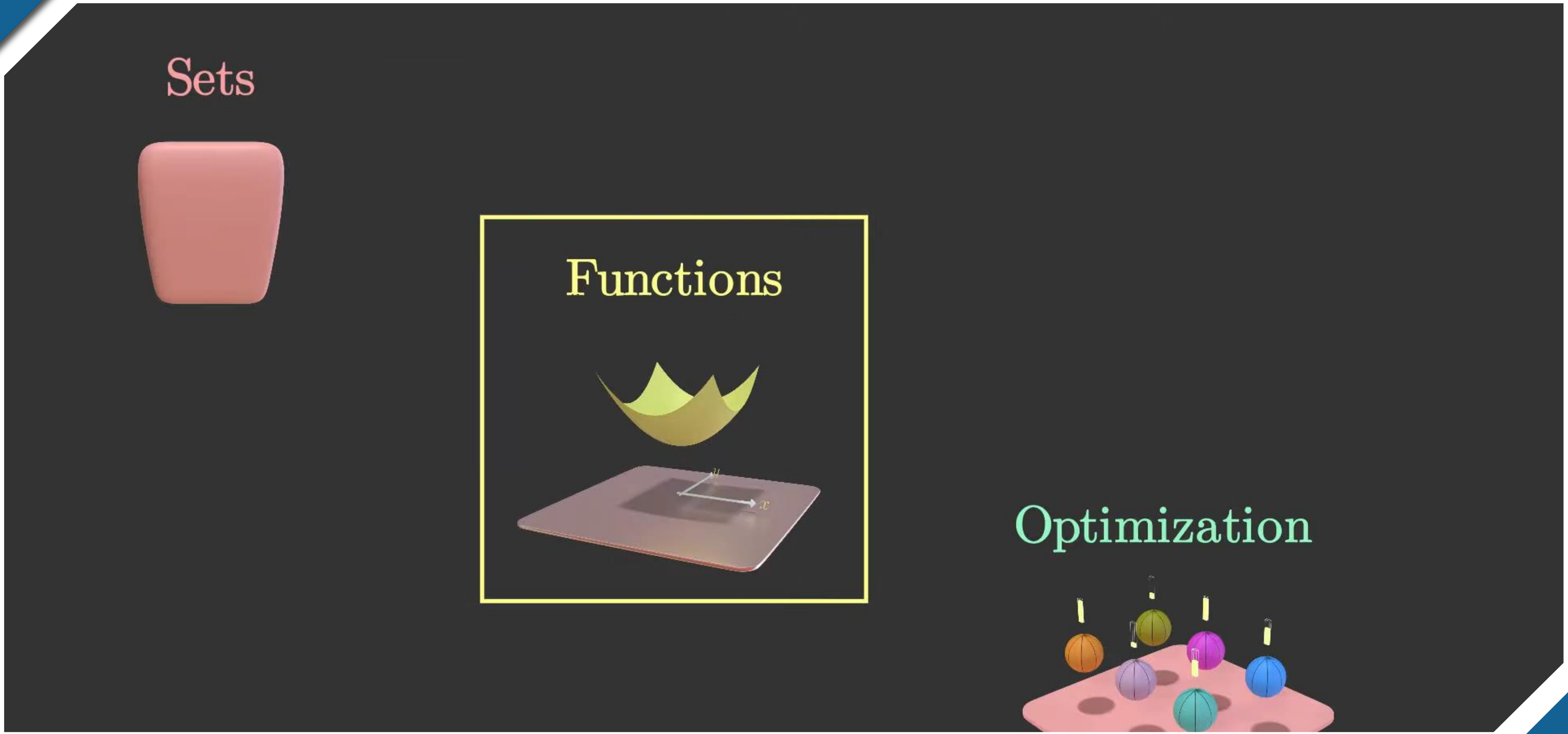


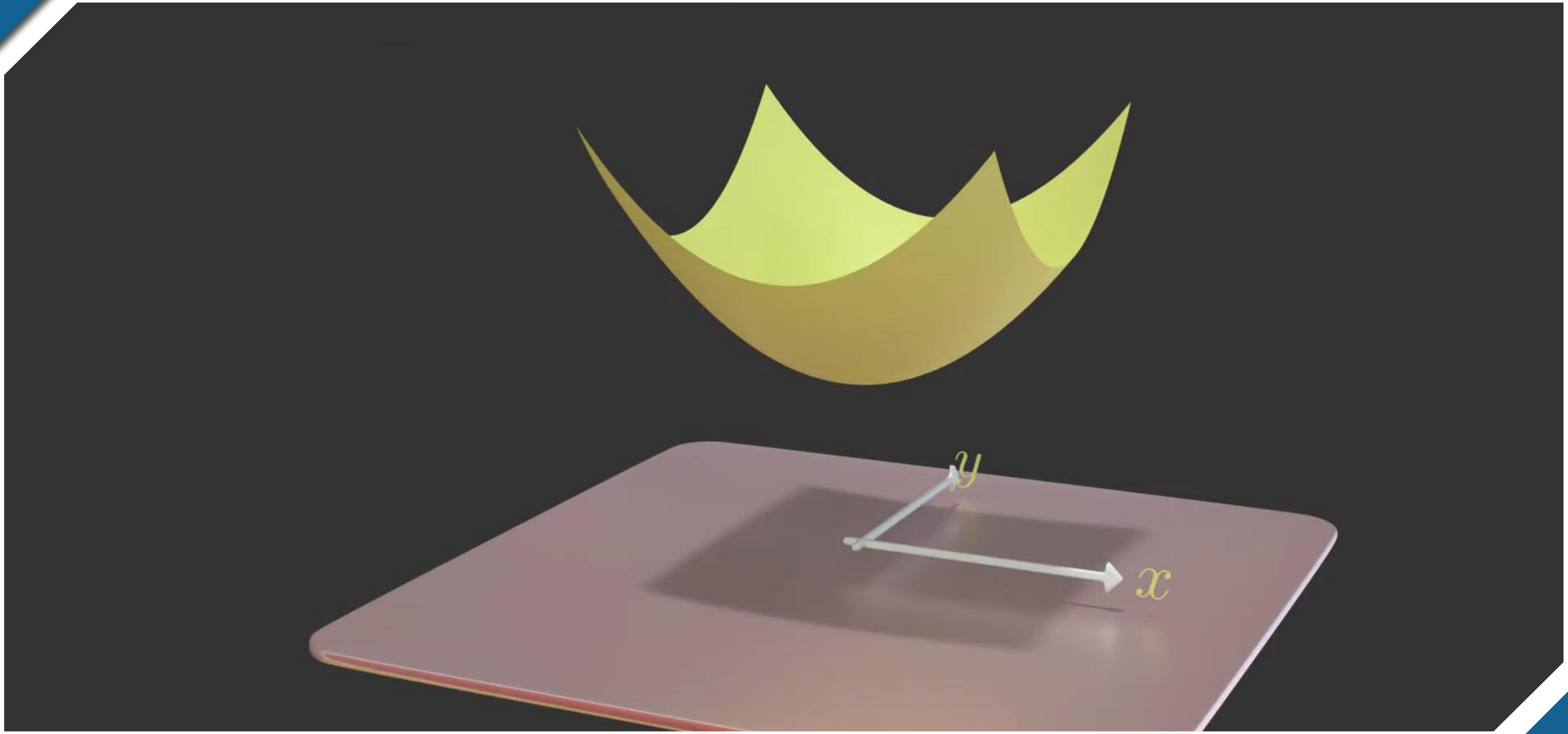


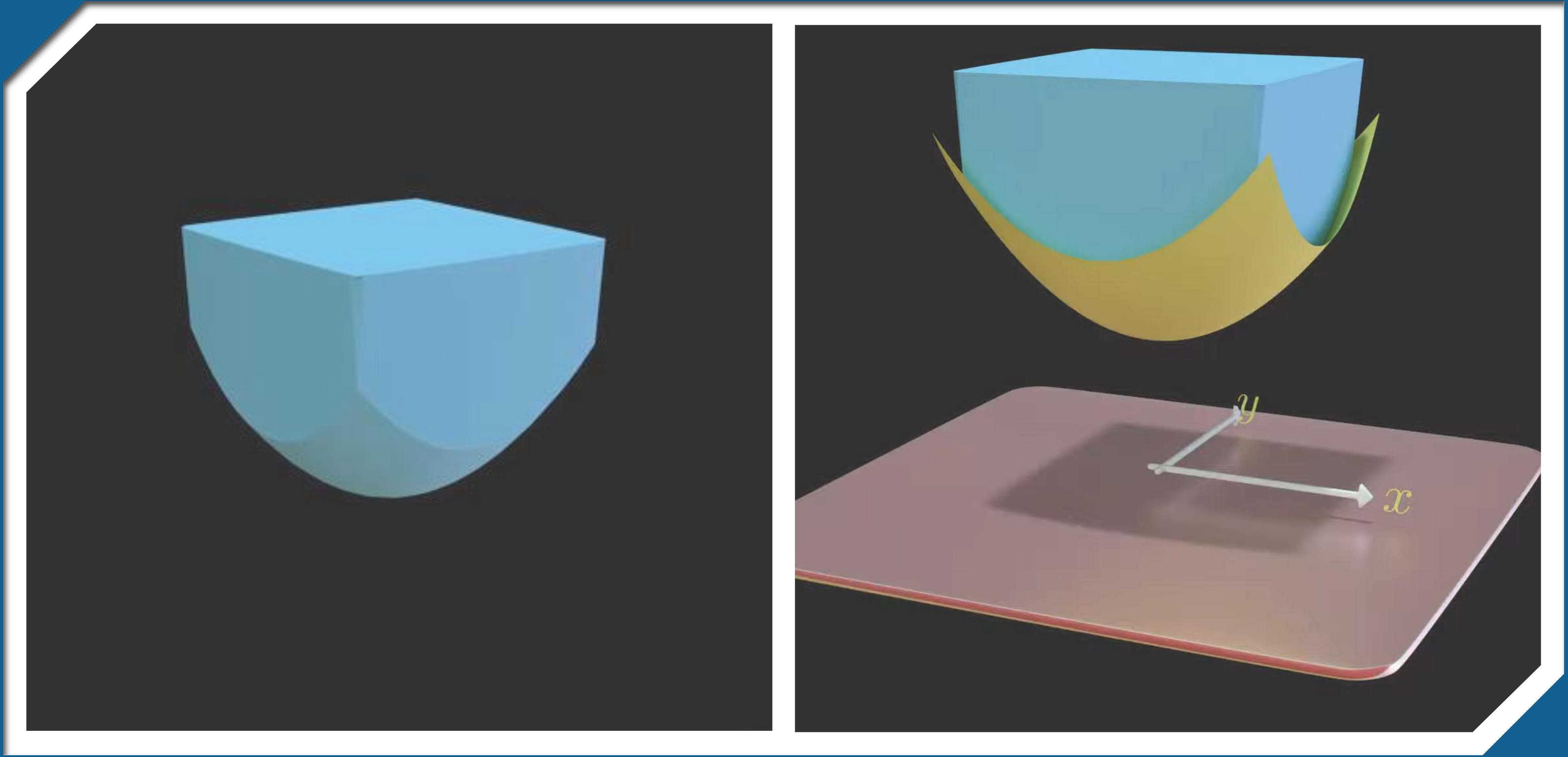




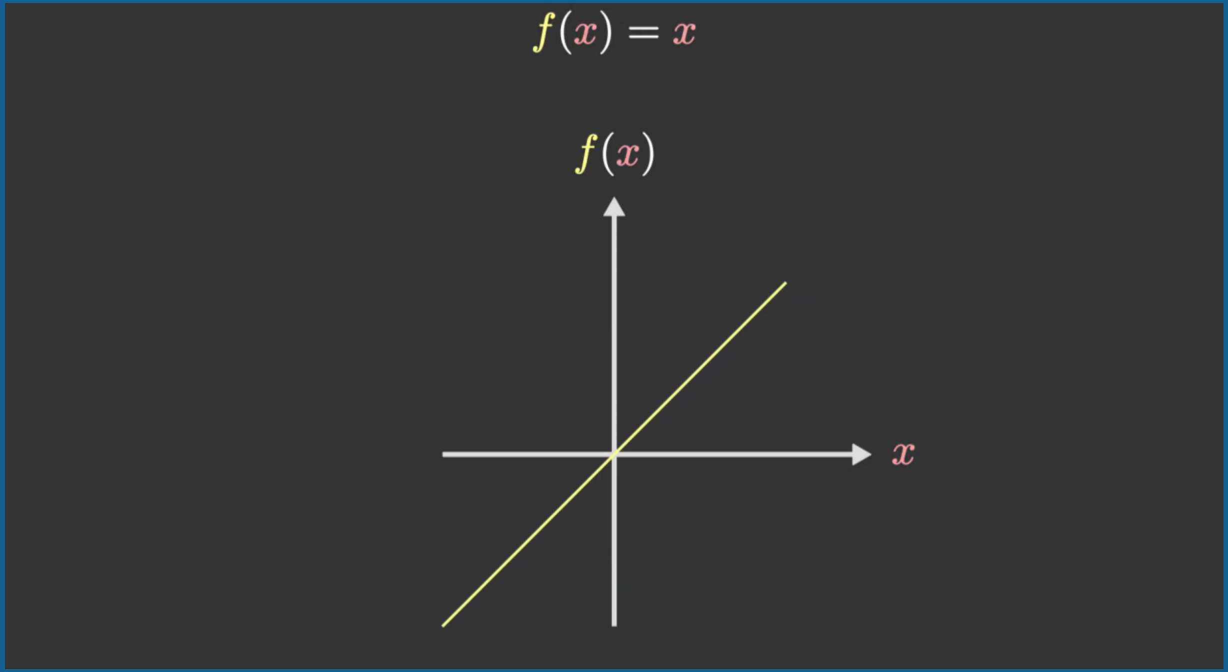



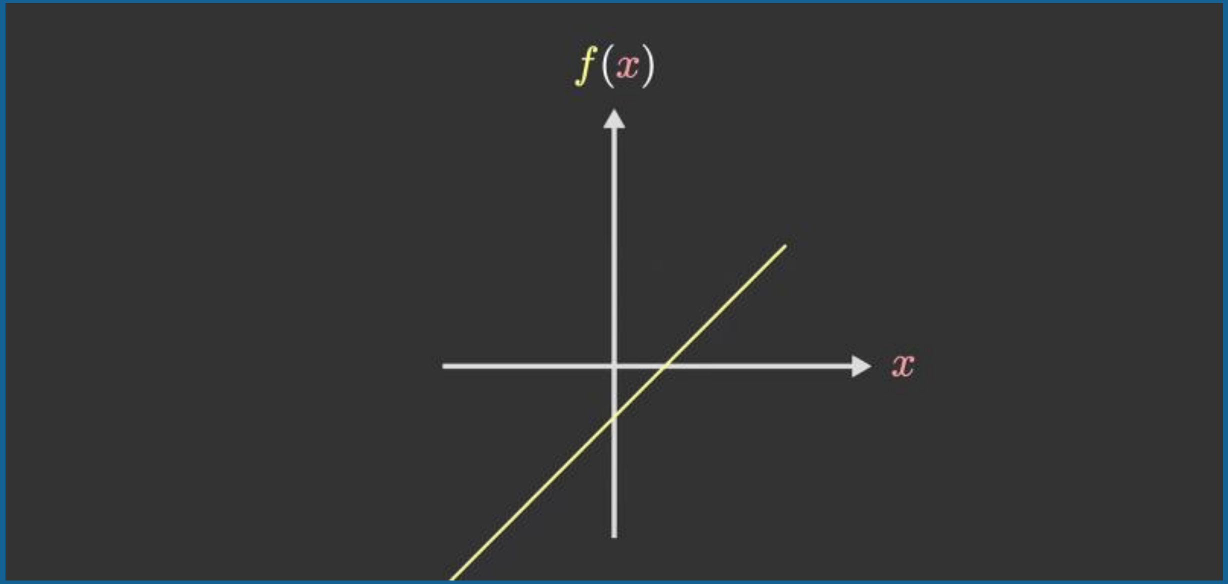

Sets
Functions
Optimization

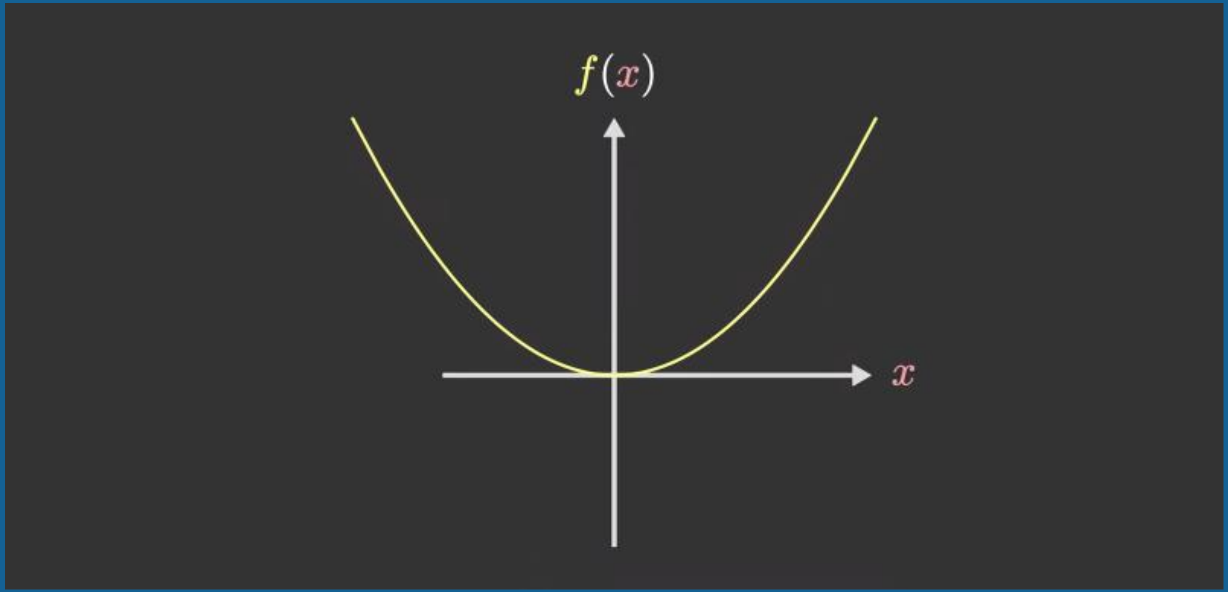

y
x

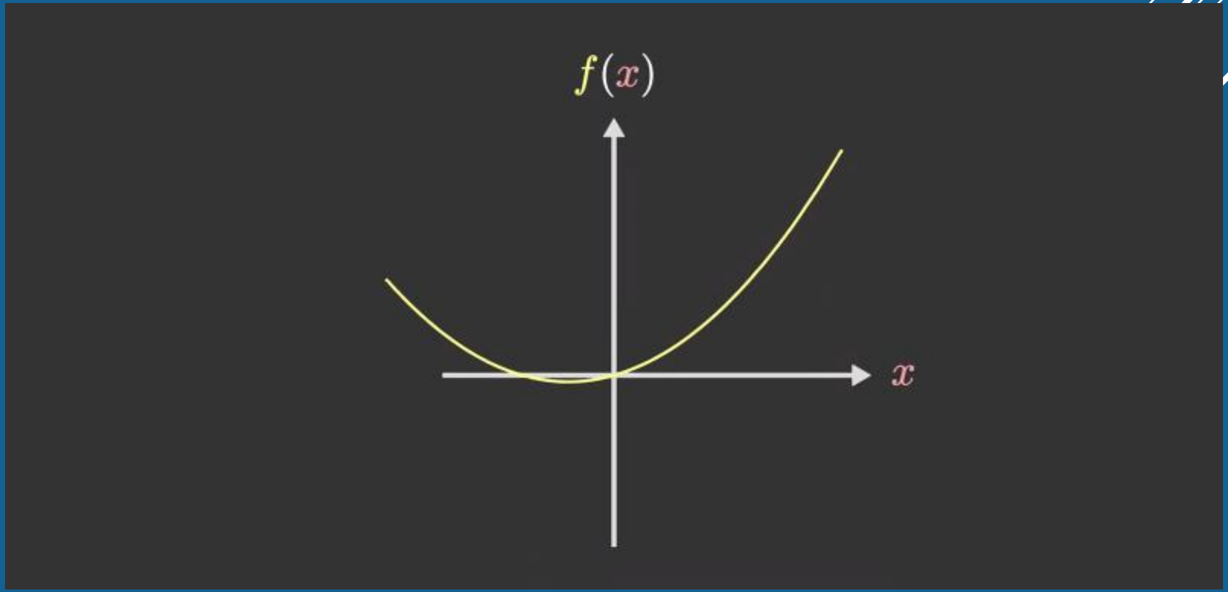


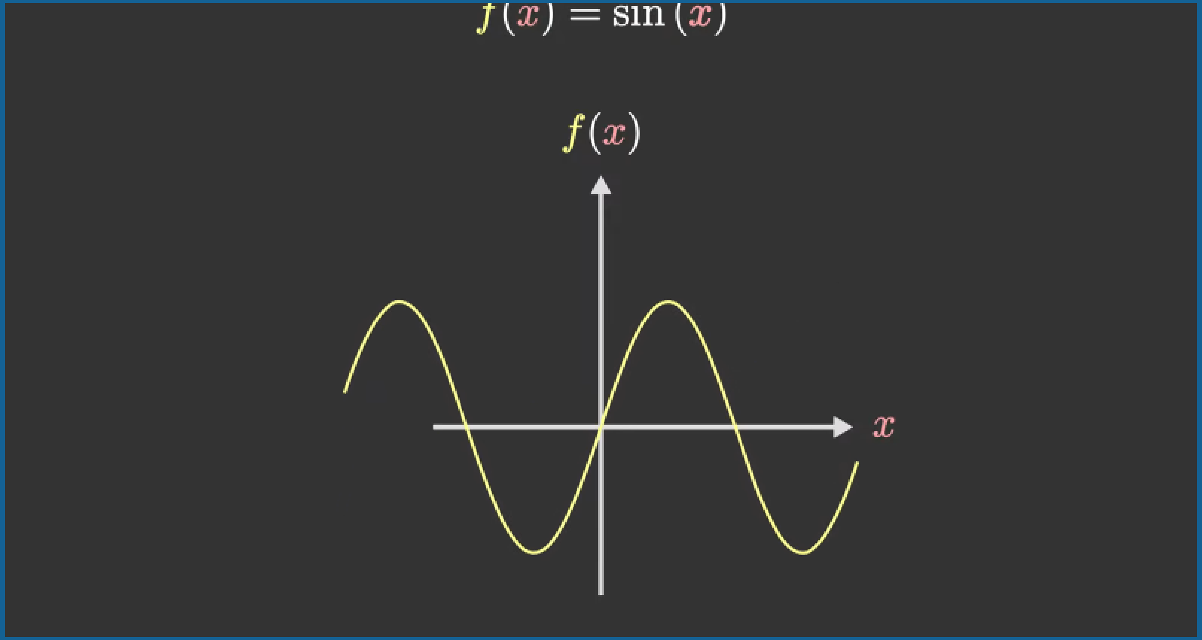
y
x

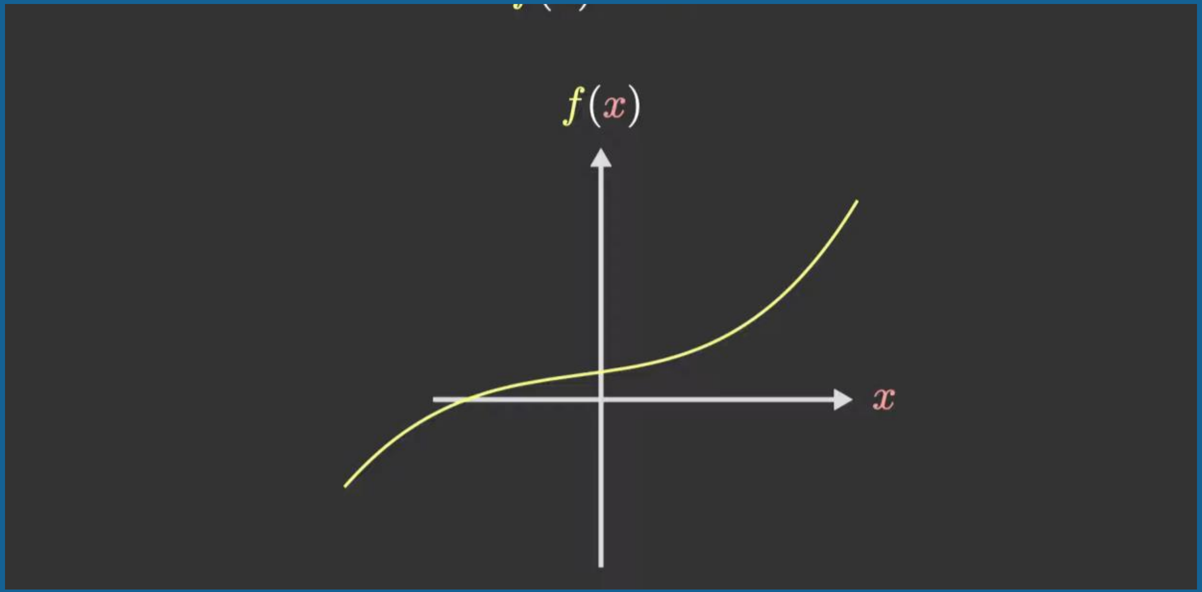

$f(x) = x$
$f(x)$
x

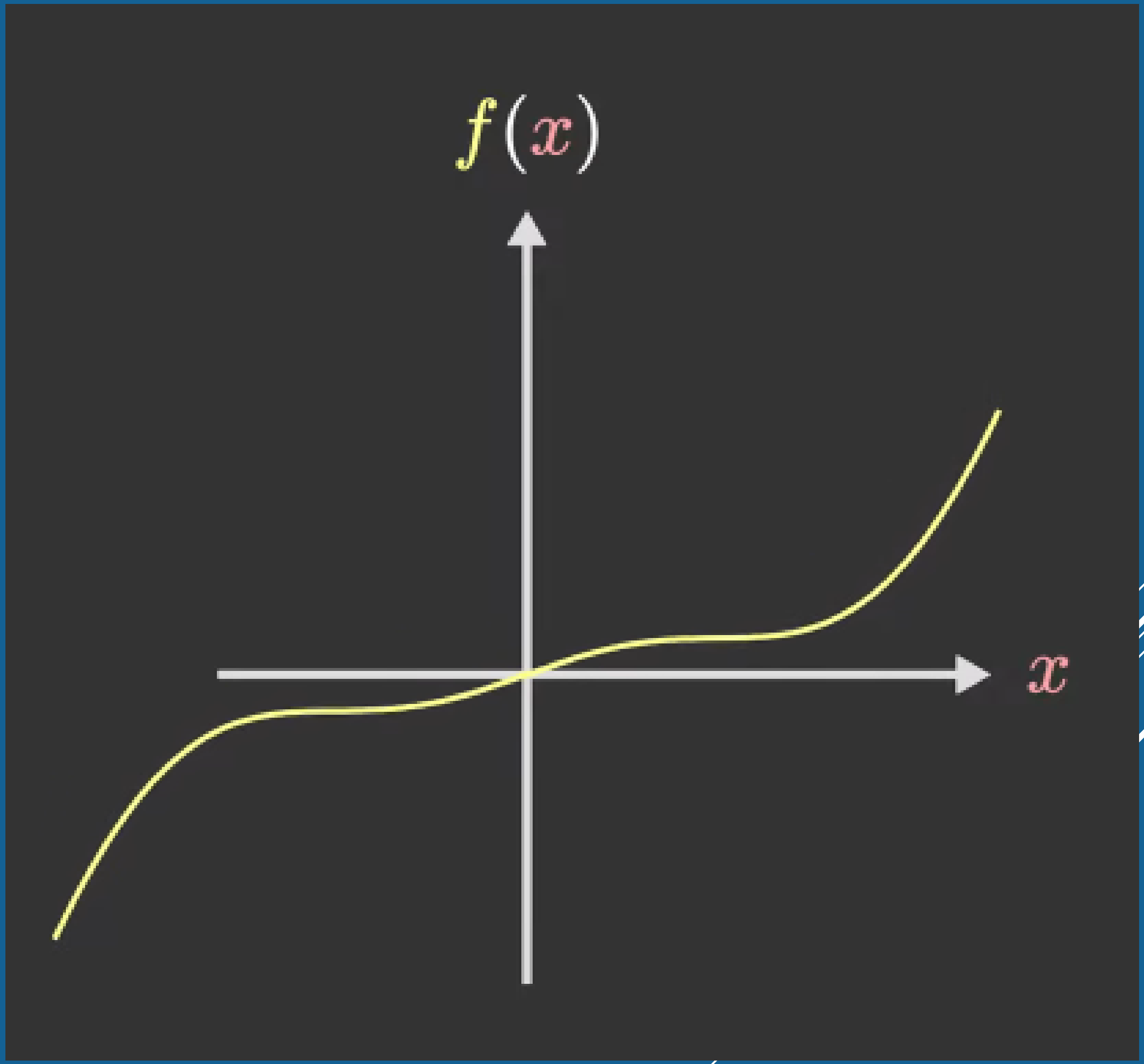

$f(x)$
x


$f(x)$
x


$f(x)$
x


f(x) = sin(x)
f(x)
x


f(x)
x


f(x)
x

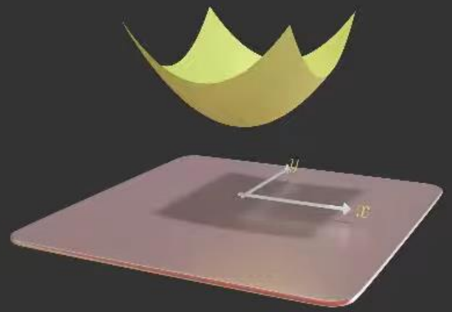
$$x^2$$

$$3\,x^2$$

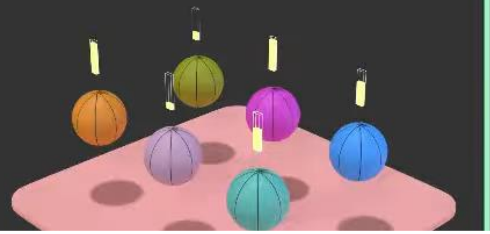
$$3\,x^2 + e^x$$

Sets

Functions



Optimization



“

... in fact, the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity.

”

- R. Tyrrell Rockafellar, 1993

"… in fact, the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity."

- R. Tyrrell Rockafellar, 1993

Easy	Hard
Linear	Nonlinear

> "... in fact, the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity."

- R. Tyrrell Rockafellar, 1993

Easy	Hard
Linear	Nonlinear
Convex	Nonconvex

Local $\longrightarrow$ Global

(through convexity)

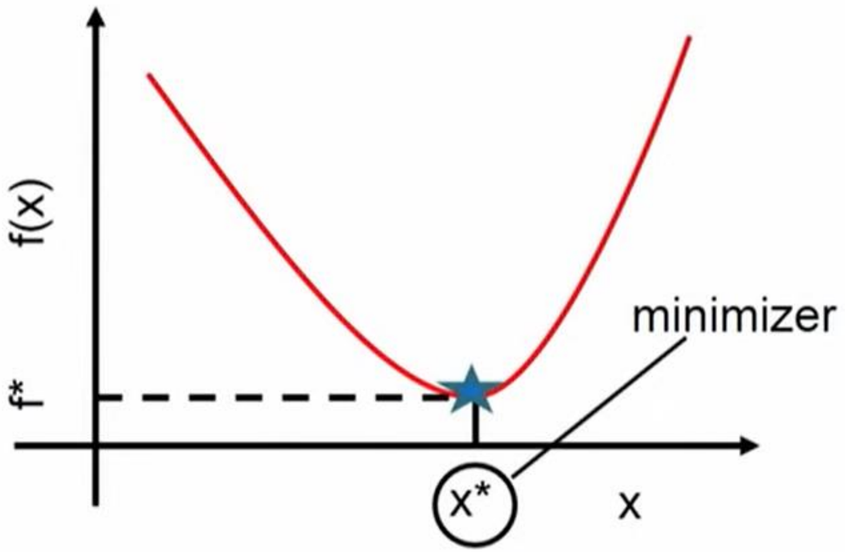
Univariate Optimization – Local and Global Optimum

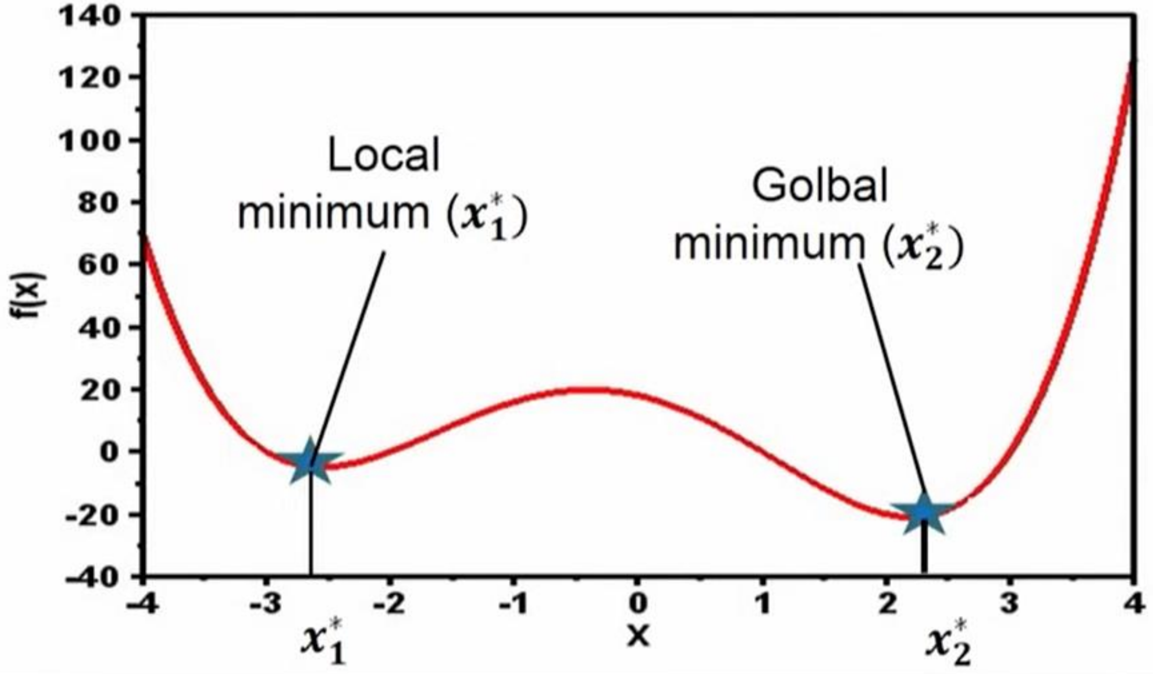
Univariate optimization

$$\boxed{\begin{array}{c} \min_{x} \quad f(x) \\ x \in R \end{array}}$$

Decision variable

Objective function





Univariate Optimization – Summary

Univariate optimization

$$\boxed{\begin{array}{c} \min\limits_{x} \quad f(x) \\ x \in R \end{array}}$$

Necessary and sufficient conditions for x^* to be the minimizer of the function $f(x)$

First order necessary conditon: $f'(x^*) = 0$

Second order sufficiency condition: $f''(x^*) > 0$

Univariate Optimization – Numerical Example

$$\min_{x} \; f(x)$$

$$f(x) = 3x^4 - 4x^3 - 12x^2 + 3$$

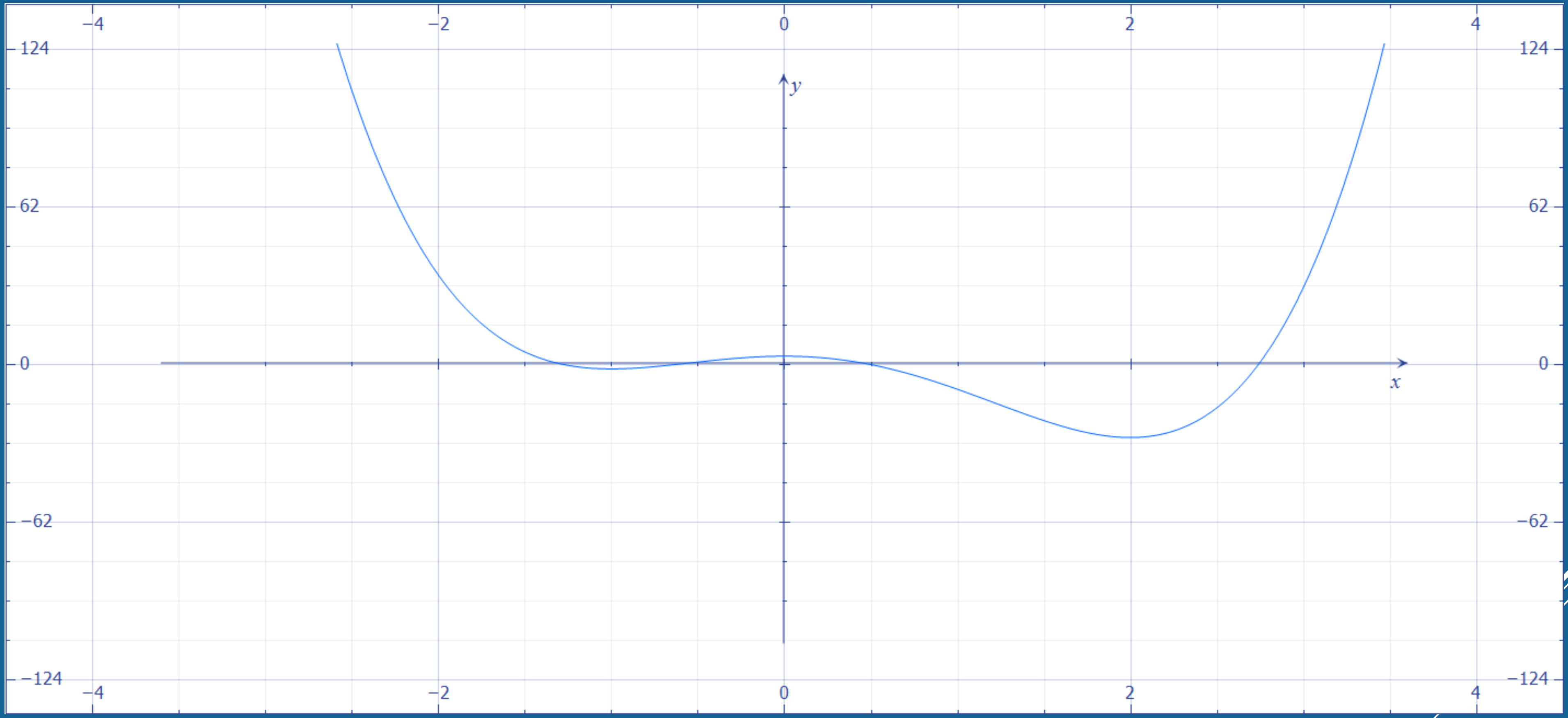
First order condition	Second order condition
$f'(x) = 12x^3 - 12x^2 - 24x = 0$	$f''(x) = 36x^2 - 24x - 24$
$= 12x(x^2 - x - 2x) = 0$	$f''(x)\big\vert_{x=0} = -24$
$= 12x(x+1)(x-2) = 0$	$f''(x)\big\vert_{x=-1} = 36 > 0$
$x = 0, x = -1, x = 2$	$f''(x)\big\vert_{x=2} = 72 > 0$

$f(-1) = -2$

$f(2) = -29$

$x^* = -1$, is a local minimizer of $f(x)$

$x^* = 2$, is a global minimizer of $f(x)$



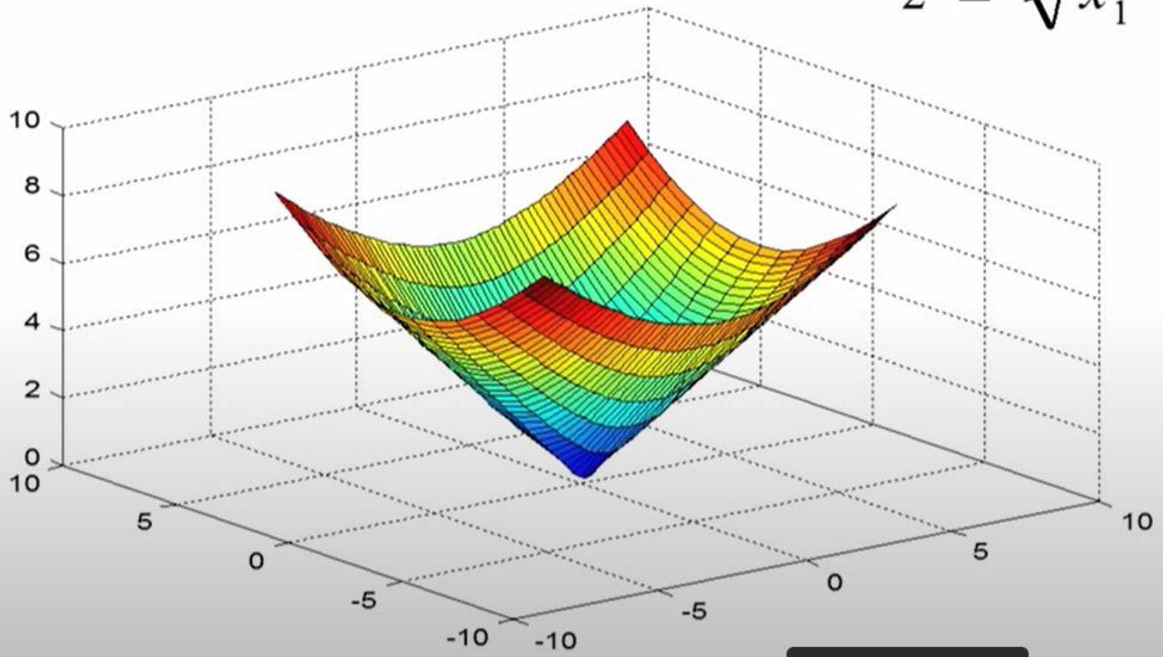
−4
−2
0
2
4
124
62
0
−62
−124
y
x

Multivariate optimization – Contour plots

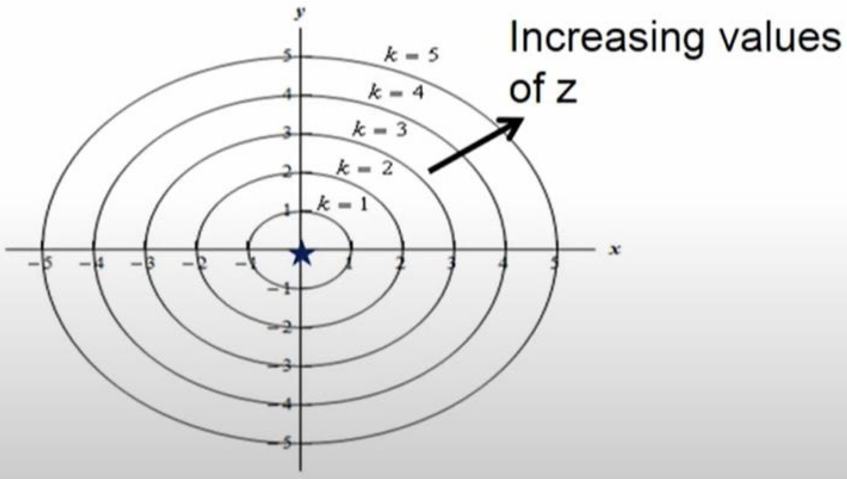
Multivariate optimization

$$z = f(x_1, x_2 \ldots. x_n)$$

$$z = \sqrt{x_1^2 + x_2^2}$$



Contour plot



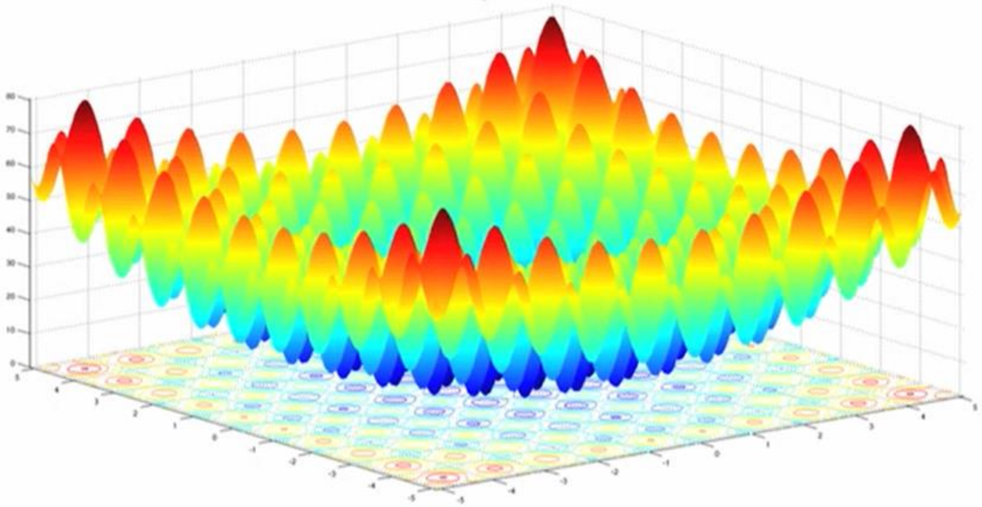
View chapter

Multivariate optimization – Local and global optimum

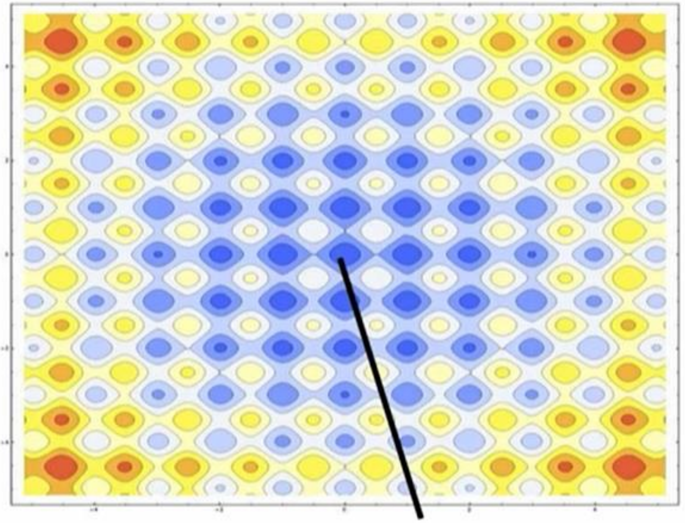
Multivariate optimization

Rastrigin function

$$f(x_1, x_2) = 20 + \sum_{i=1}^{2} [x_i^2 - 10\cos(2\pi x_i)]$$



Contour plot



Global minimum at [0,0]

Multivariate optimization

$$z = f(x_1, x_2 \ldots. x_n)$$

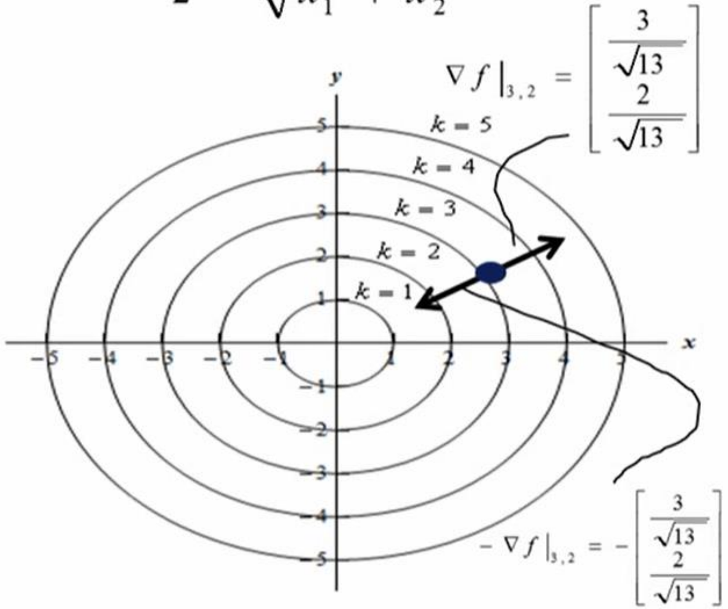
$$z = \sqrt{x_1^2 + x_2^2}$$

Gradient

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \cdots \\ \cdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

Hessian

$$\nabla^2 f = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$



- Gradient of a function at a point is orthogonal to the contours
- Gradient points in the direction of greatest increase of the function
- Negative gradient points in the direction of the greatest decrease of the function
- Hessian is a symmetric matrix

Overall Summary – Univariate and multivariate local optimum conditions

Multivariate optimization

$$\min_{x} \; f(x)$$
$$x \in R$$

$$\min_{\bar{x}} \; f(\bar{x})$$
$$\bar{x} \in R^{n}$$

Necessary condition for x^* to be the minimizer

$$f'(x^*) = 0$$

Sufficient condition

$$f''(x^*) > 0$$

Necessary condition for $\overline{x^*}$ to be the minimizer

$$\nabla f(\bar{x}^*) = 0$$

Sufficient condition

$\nabla^2 f(\overline{x^*})$ has to be positive definite

Multivariate optimization – Numerical example

Multivariate optimization

$$\min_{x_1, x_2} \quad x_1 + 2x_2 + 4x_1^2 - x_1 x_2 + 2x_2^2$$

First order condition

$$\nabla f = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 1 + 8x_1 - x_2 \\ 2 - x_1 + 4x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

solving

$$\begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} -0.19 \\ -0.54 \end{bmatrix}$$

Second order condition

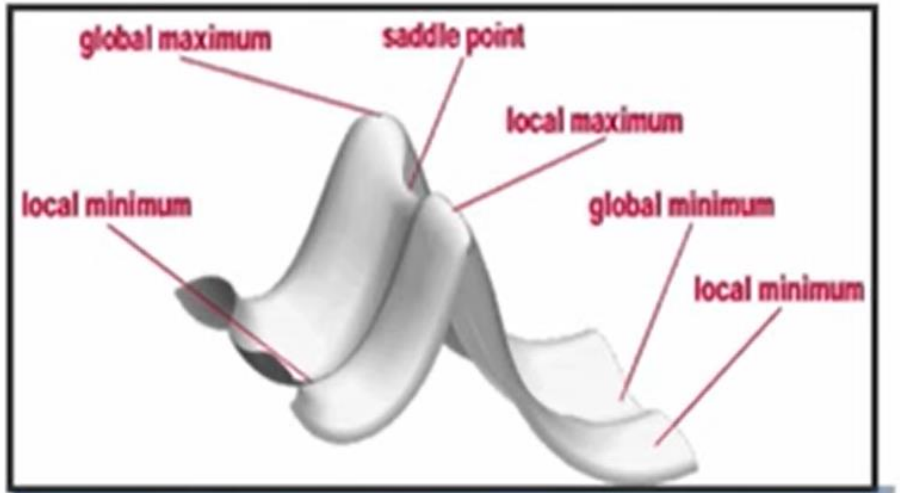
$$\nabla^2 f = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 x_2} \\ \dfrac{\partial^2 f}{\partial x_2 x_1} & \dfrac{\partial^2 f}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 8 & -1 \\ -1 & 4 \end{bmatrix}$$

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 3.76 \\ 8.23 \end{bmatrix}$$

$$\begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} -0.19 \\ -0.54 \end{bmatrix}$$

Unconstrained multivariate optimization - Directional search

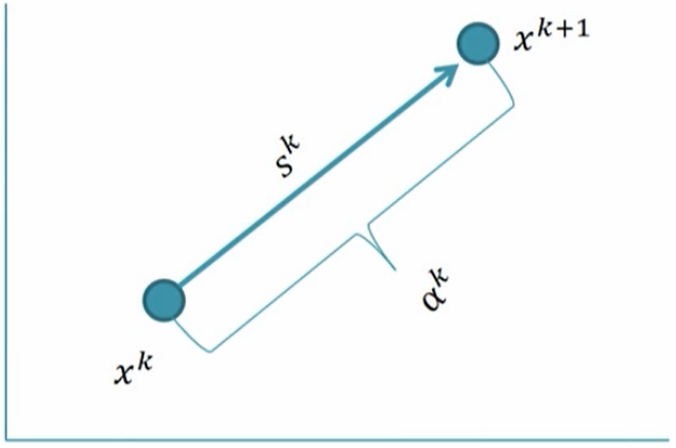
- Aim is to reach the bottom most region
- Directions of descent
- Steepest descent
- Sometimes we might even want to climb the mountain for better prospects to get down further



Unconstrained multivariate optimization - Descent direction and movement

- Iterative

$$x^{k+1} = \underbrace{x^k}_{\text{Starting point}} + \underbrace{\alpha^k}_{\text{Step length}} \underbrace{s^k}_{\text{Search direction}}$$



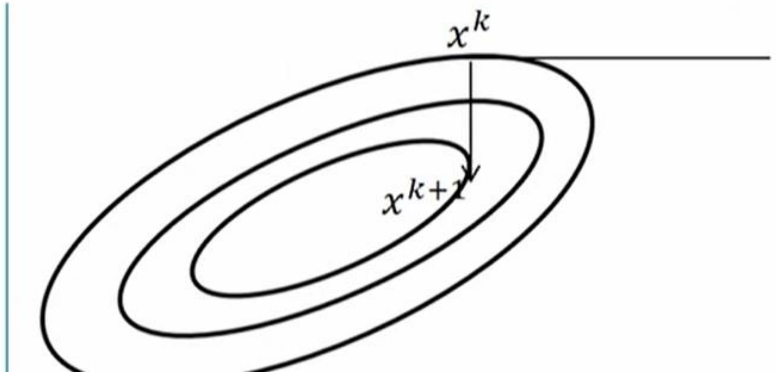
Multivariate Optimization

- Search direction computation
- Step length computation = Univariate optimization

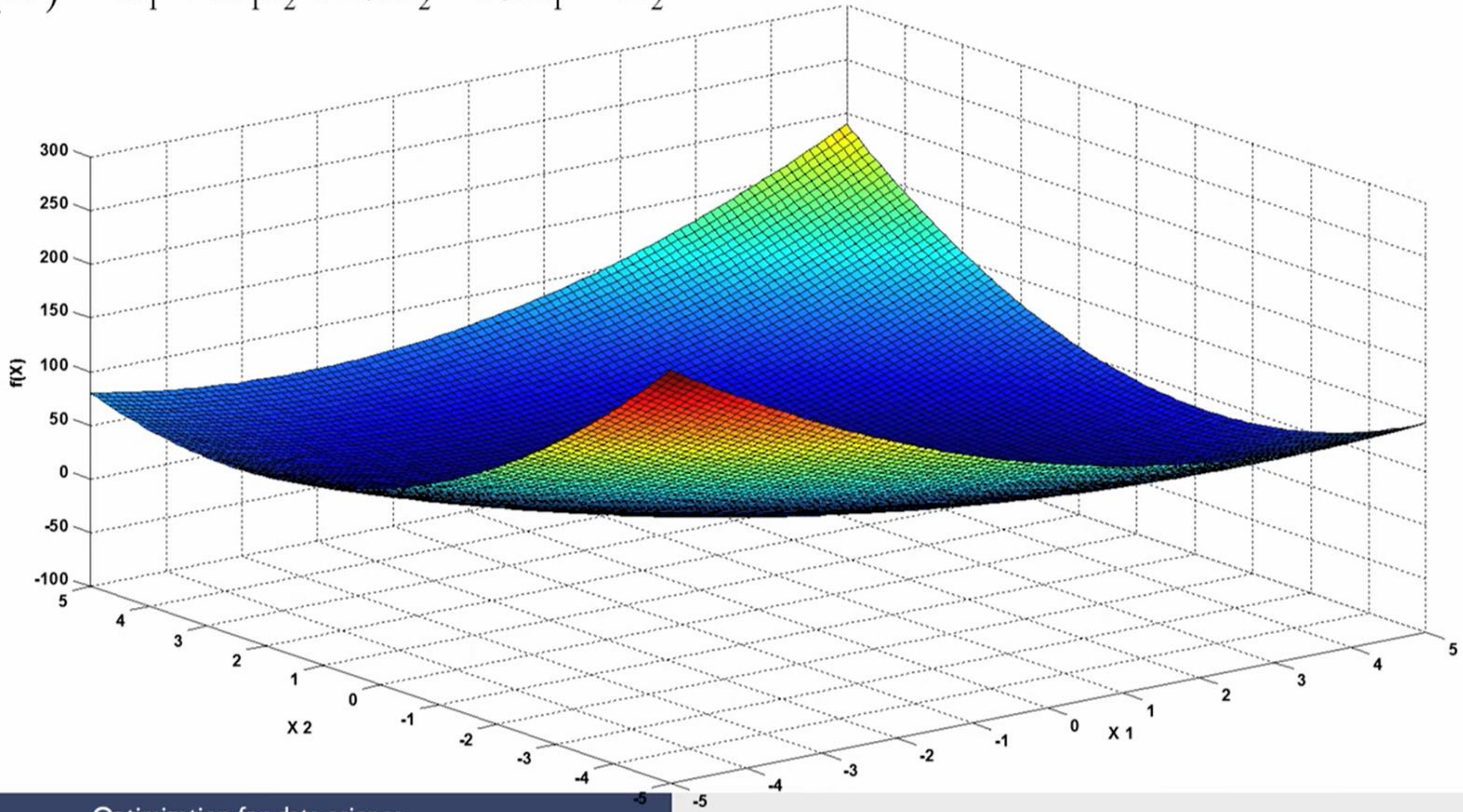
- In ML techniques, this is called as the learning rule
- In neural networks
  - Back-propagation algorithm
  - Same gradient descent with application of chain rule
- In clustering
  - Minimization of an Euclidean distance norm

Steepest descent and optimum step size

- Minimize $f(x_1, x_2, \ldots x_n) = f(x)$
- **Steepest descent**
  - At iteration k starting point is x^k
  - Search direction s^k = Negative of gradient of $f(x) = -\nabla f(x^k)$
  - New point is $x^{k+1} = x^k + \alpha^k s^k$ where α^k is the value of α for which $f(x^{k+1}) = f(\alpha) =$ is a minimum (univariate minimization)



$$f\left(X\right)=4x_1^2+3x_1x_2+2.5x_2^2-5.5x_1-4x_2$$



$$f'(X) = \begin{bmatrix} 8x_1 + 3x_2 - 5.5 \\ 3x_1 + 5x_2 - 4 \end{bmatrix}$$

Learning parameter $(\alpha) = 0.135$

Initial guess $(X_0) = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ $\quad f(X_0) = 19$

Gradient Descent (or) Learning Rule in ML

Step 1: $X_1 = X_0 - \alpha f'(X_0)$

$$X_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - 0.135 \begin{bmatrix} 8x_{0,1} + 3x_{0,2} - 5.5 \\ 3x_{0,1} + 5x_{0,2} - 4 \end{bmatrix}$$

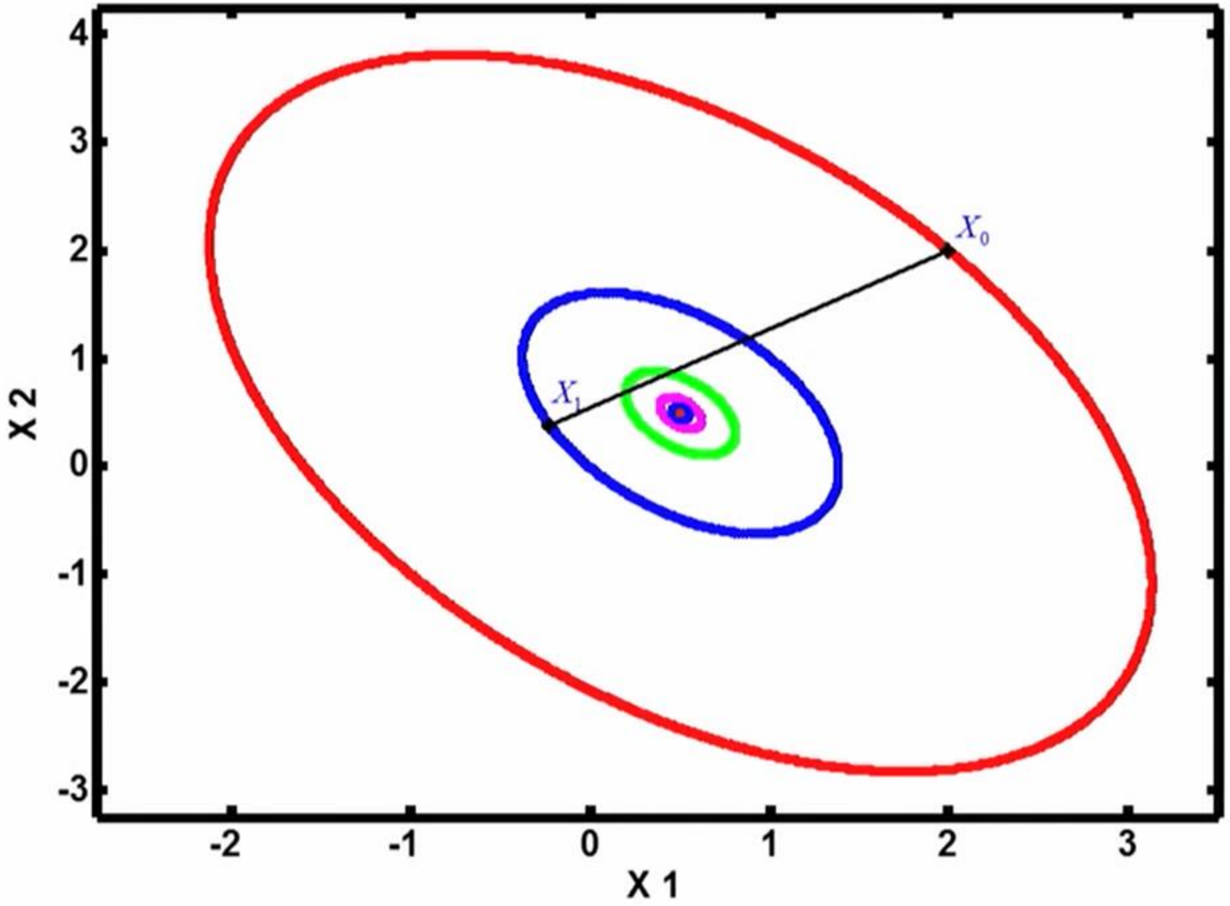
$$X_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - 0.135 \begin{bmatrix} 8(2) + 3(2) - 5.5 \\ 3(2) + 5(2) - 4 \end{bmatrix}$$

$$X_1 = \begin{bmatrix} -0.2275 \\ 0.3800 \end{bmatrix} \quad f(X_1) = 0.0399$$

Constant objective function contour plots

$$f(X) = 4x_1^2 + 3x_1x_2 + 2.5x_2^2 - 5.5x_1 - 4x_2 = K$$

Quadratic in this case - ellipse



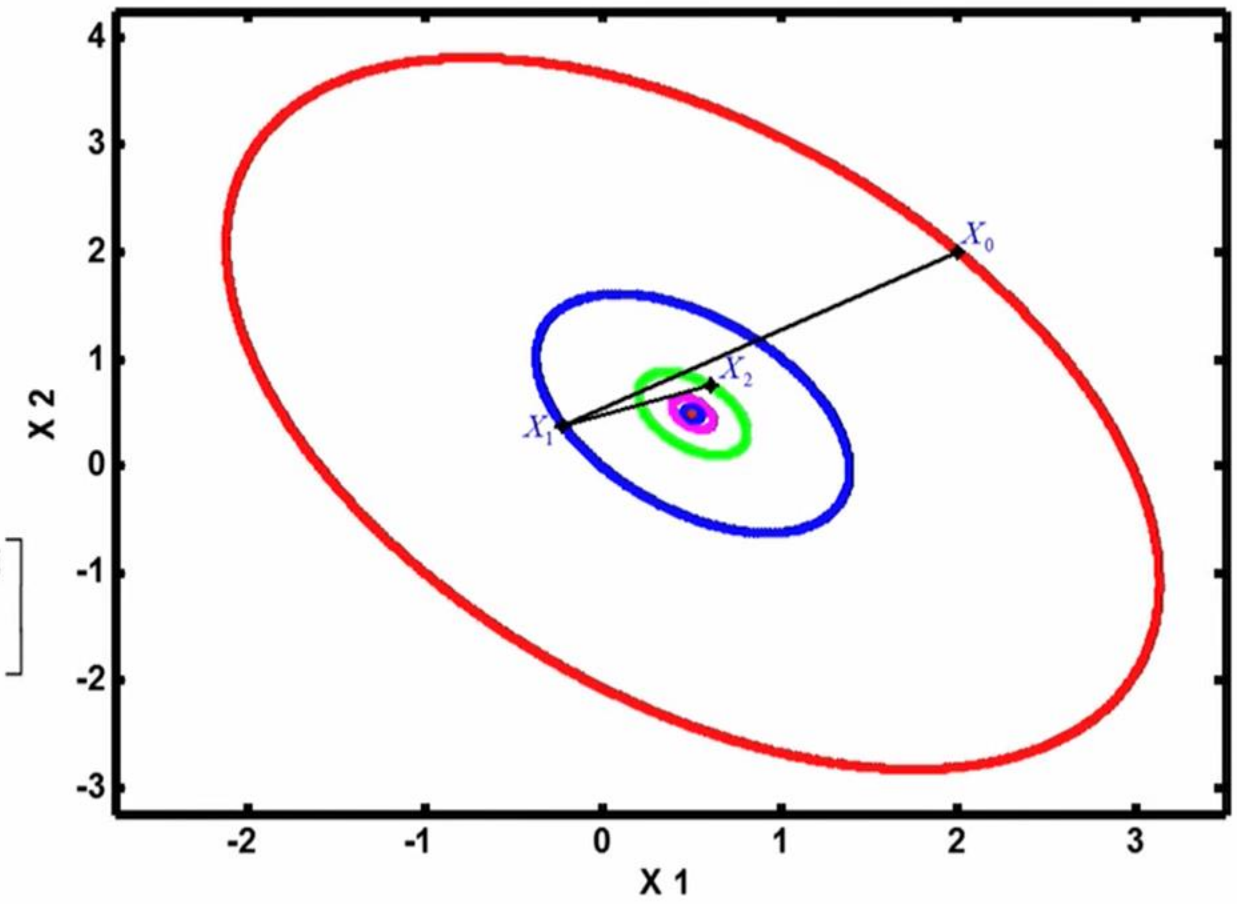
$$\text{First iteration } (X_1) = \begin{bmatrix} -0.2275 \\ 0.3800 \end{bmatrix}$$

$$\text{Step 2: } X_2 = X_1 - \alpha f'(X_1)$$

$$X_2 = \begin{bmatrix} -0.2275 \\ 0.3800 \end{bmatrix} - 0.135 \begin{bmatrix} 8x_{1,1} + 3x_{1,2} - 5.5 \\ 3x_{1,1} + 5x_{1,2} - 4 \end{bmatrix}$$

$$X_2 = \begin{bmatrix} -0.2275 \\ 0.3800 \end{bmatrix} - 0.135 \begin{bmatrix} 8(-0.2275) + 3(0.3800) - 5.5 \\ 3(-0.2275) + 5(0.3800) - 4 \end{bmatrix}$$

$$X_2 = \begin{bmatrix} 0.6068 \\ 0.7556 \end{bmatrix} \qquad f(X_2) = -2.0841$$



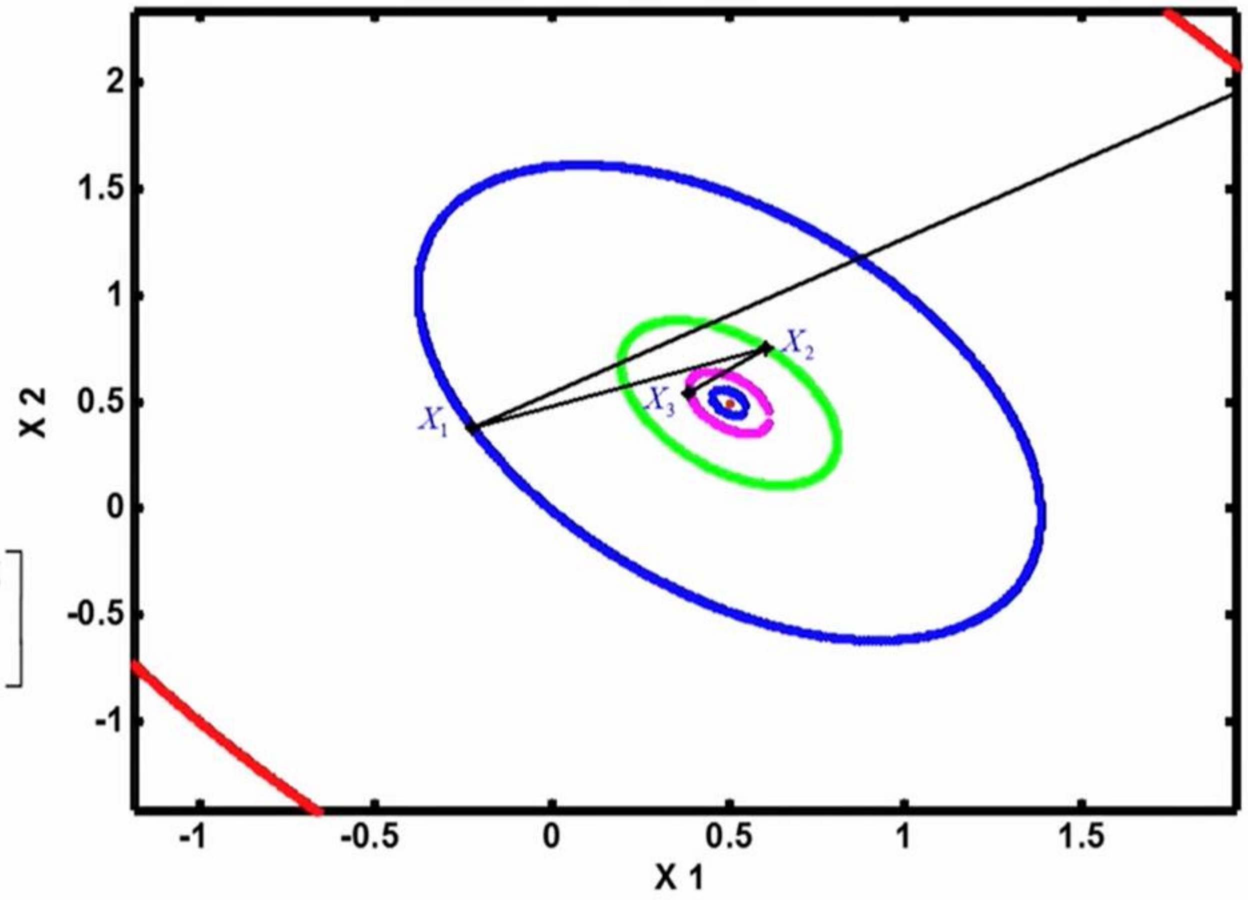
Second iteration $(X_2) = \begin{bmatrix} 0.6068 \\ 0.7556 \end{bmatrix}$

Step 3: $X_3 = X_2 - \alpha f'(X_2)$

$$X_3 = \begin{bmatrix} 0.6068 \\ 0.7556 \end{bmatrix} - 0.135 \begin{bmatrix} 8x_{2,1} + 3x_{2,2} - 5.5 \\ 3x_{2,1} + 5x_{2,2} - 4 \end{bmatrix}$$

$$X_3 = \begin{bmatrix} 0.6068 \\ 0.7556 \end{bmatrix} - 0.135 \begin{bmatrix} 8(0.6068) + 3(0.7556) - 5.5 \\ 3(0.6068) + 5(0.7556) - 4 \end{bmatrix}$$

$$X_3 = \begin{bmatrix} 0.3879 \\ 0.5398 \end{bmatrix} \qquad f(X_3) = -2.3342$$



Third iteration $(X_3)=\begin{bmatrix}0.3879\\0.5398\end{bmatrix}$

Step 4: $X_4 = X_3 - \alpha f'(X_3)$

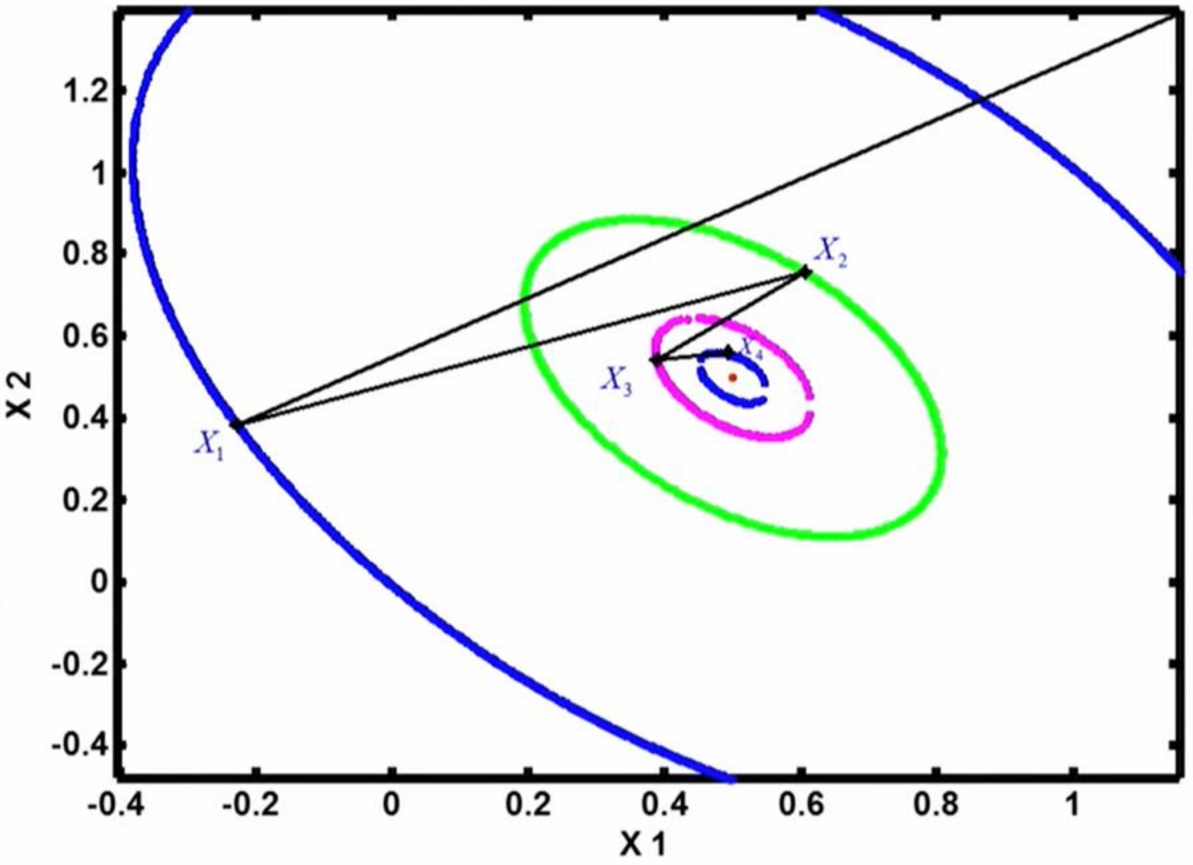
$$X_4 = \begin{bmatrix}0.3879\\0.5398\end{bmatrix} - 0.135\begin{bmatrix}8x_{3,1}+3x_{3,2}-5.5\\3x_{3,1}+5x_{3,2}-4\end{bmatrix}$$

$$X_4 = \begin{bmatrix}0.3879\\0.5398\end{bmatrix} - 0.135\begin{bmatrix}8(0.3879)+3(0.5398)-5.5\\3(0.3879)+5(0.5398)-4\end{bmatrix}$$

$$X_4 = \begin{bmatrix}0.4928\\0.5583\end{bmatrix} \qquad f(X_4) = -2.3675$$

Optimal solution $(X_{opti})=\begin{bmatrix}0.5\\0.5\end{bmatrix}$ $\quad f(X_{opti}) = -2.3750$

Gradient is zero at the optimum point



Third iteration $(X_3)=\begin{bmatrix} 0.3879 \\ 0.5398 \end{bmatrix}$ ✓

$x^{k+1} \leftarrow x^k$

$f(x^{k+1}) \leftarrow f(x^k)$

$\nabla f(x^{k+1}) \leftarrow \nabla f(x^k)$

Step 4: $X_4 = X_3 - \alpha f'(X_3)$

$$X_4 = \begin{bmatrix} 0.3879 \\ 0.5398 \end{bmatrix} - 0.135 \begin{bmatrix} 8x_{3,1} + 3x_{3,2} - 5.5 \\ 3x_{3,1} + 5x_{3,2} - 4 \end{bmatrix}$$

$$X_4 = \begin{bmatrix} 0.3879 \\ 0.5398 \end{bmatrix} - 0.135 \begin{bmatrix} 8(0.3879) + 3(0.5398) - 5.5 \\ 3(0.3879) + 5(0.5398) - 4 \end{bmatrix}$$

$$X_4 = \begin{bmatrix} 0.4928 \\ 0.5583 \end{bmatrix} \checkmark \qquad f(X_4) = -2.3675$$

Optimal solution $(X_{opti}) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$ $f(X_{opti}) = -2.3750$

Gradient is zero at the optimum point