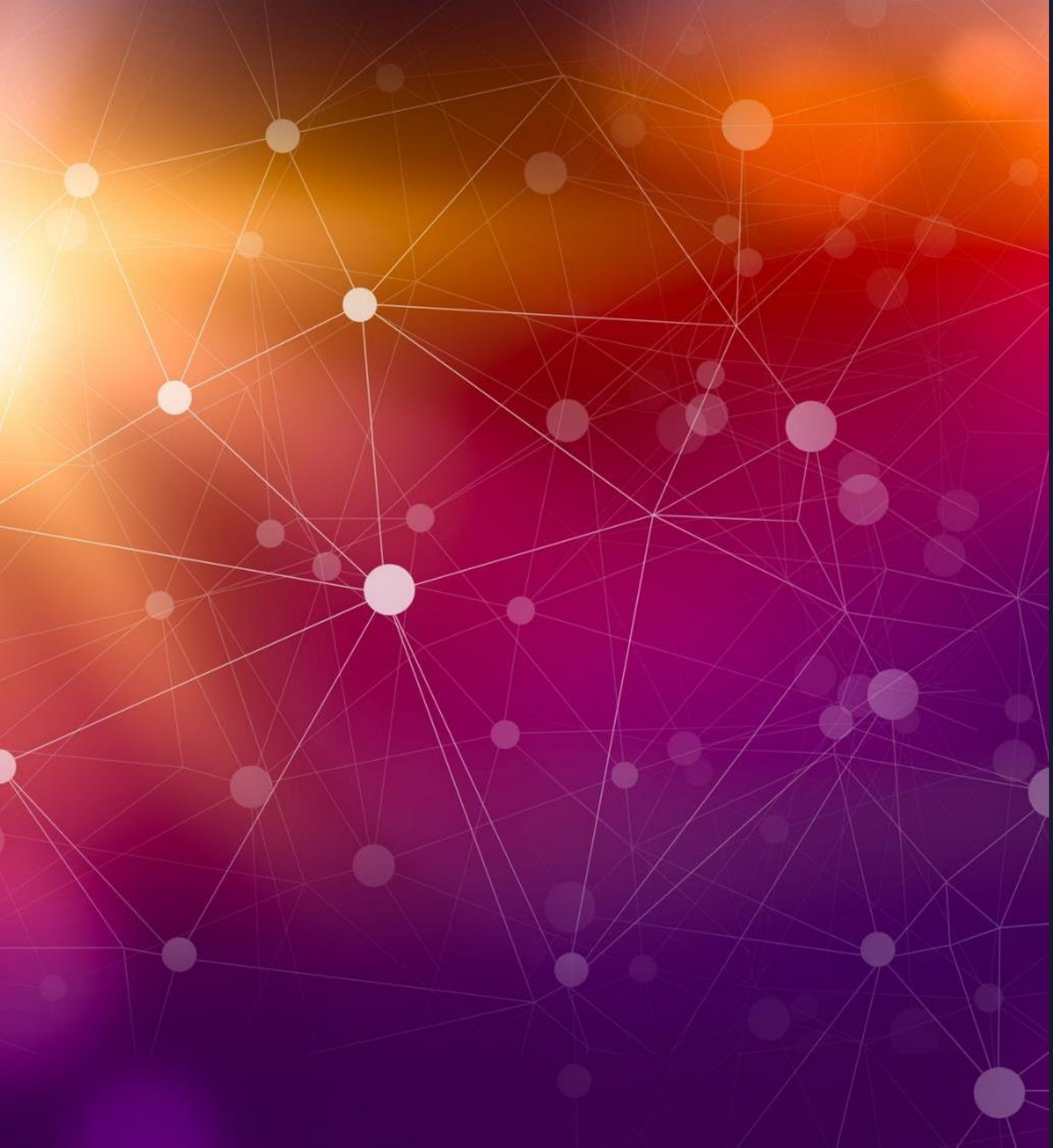


IEEE FCI - Helwan Student Branch

Calculus for ML

Hossam Ahmed

Ziad Waleed

Mario Mamdouh

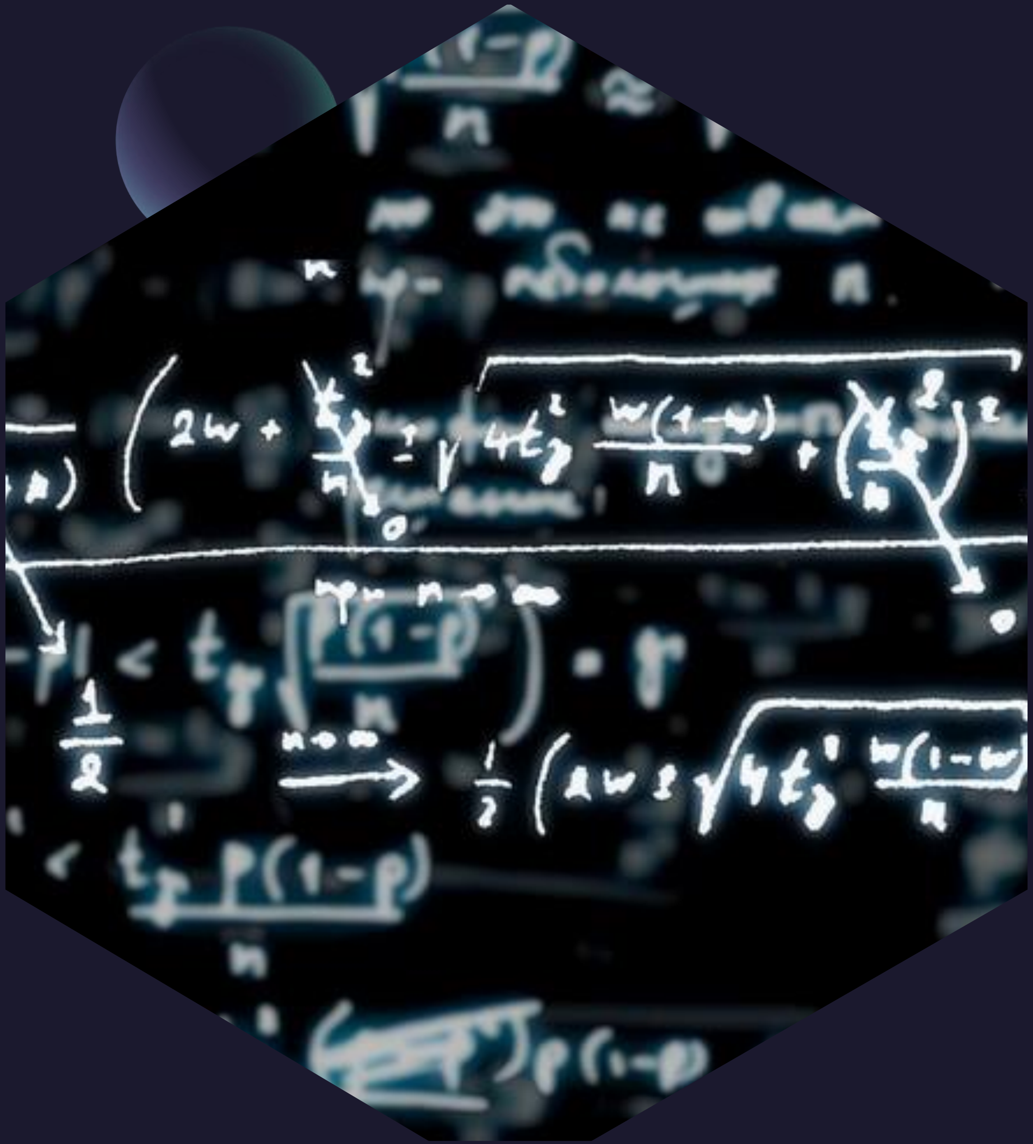
NumPy

Agenda

- Motivation
- Limits
- When doesn’t limits exist?
- Limits and continuity
- From limits to differentiation
- Numerical differentiation
- Differentiation Rules
- Partial derivative
- Chain Rule
- Gradient
- Beyond gradient
- Optimization
- Optimization and the learning problem
- Gradient Ascent for maximizing
- Gradient Descent for minimizing
- Gradient Ascent ⬆️ vs Gradient Descent ⬇️
- The effect of learning rate η

Motivation

- Calculus, is not an isolated mathematical field.
- Calculus is the engine that powers the machines with the ability to learn and adapt.
- Calculus provide a set of tools for optimizing our machine learning models.
- When ever you train ML model you probably are doing a lot of derivatives in the background.



Limits

- Where would you reach if you kept following this way?
- The answer doesn't mean you've fully reached it, and you may never get there, but you're getting as close as possible
- You can think of it as Zooming 🔍 a picture 🖼️, the more you zoom the more you see more details, until you reach a point you can't zoom any more, this would be limit of zooming the image, further zooming won't change the image.
- Example, consider a function $f(x) = 2x$, As x gets closer to 3, $2x$ would get closer to 6, the $\lim_{x \to 3} f(x) = 6$
- You won't always end with a finite number as a limit, for some functions you the limit may be infinity ♾️, for example $\lim_{x \to 0^+} \left(\frac{1}{x}\right) = \infty$, try inputting (0.1, 0.01, 0.001,…) .

Limits

- You won't always end with a finite number as a limit, for some functions you the limit may be infinity ∞, for example $\lim_{x\to 0^+}\left(\frac{1}{x}\right) = \infty$, try inputting (0.1, 0.01, 0.001,…) .

$f(0.1)$	$f(0.01)$	$f(0.001)$	$f(0.0001)$	$f(0.00001)$	…	$f(0.000001)$
10	100	1000	10000	100000	?	1000000

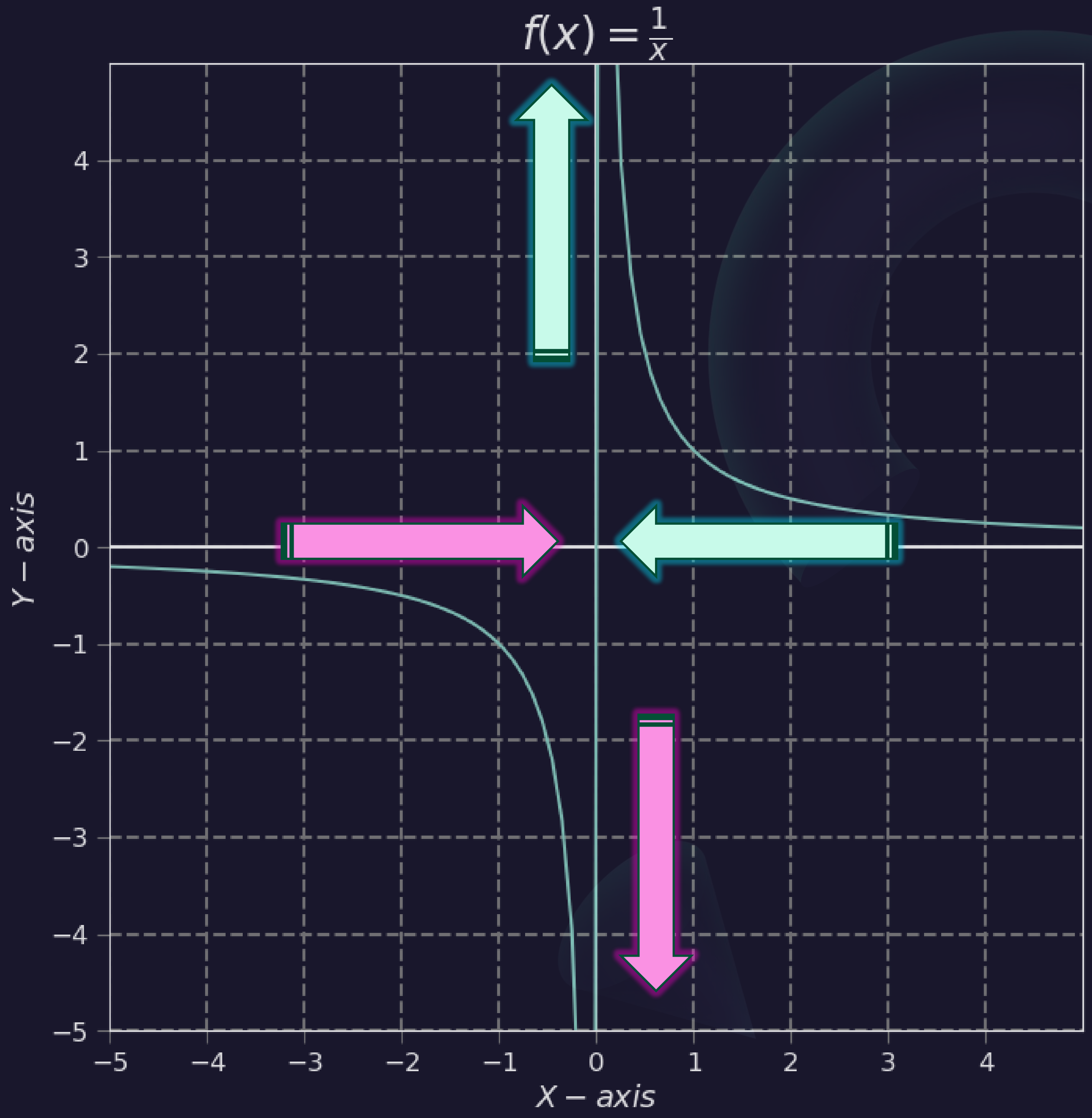
- How many zeros can you add before you reach zero ? (you try not to reach)

- You can add infinity zeros so that the output of the function is going towards infinity.

- $\lim_{x\to c} f(x) = L$, the limit of f as x approaches c equals L, this mean the value (output) of the function can be made arbitrarily close to L by choosing x sufficiently close to c.

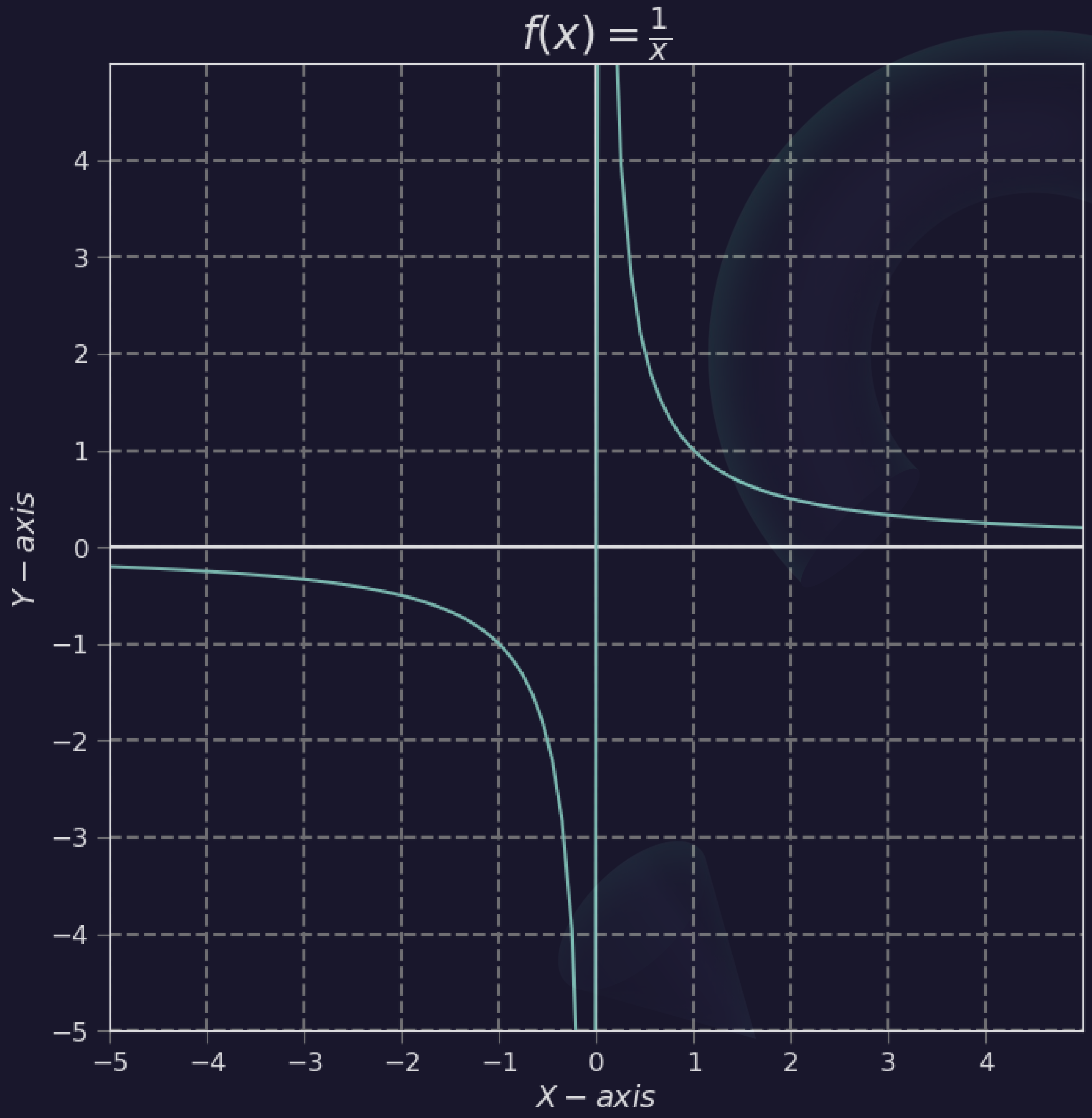
Limits

- Till now we treated this function from the positive side only, but it has two parts.
- When we inputted numbers that are close to zero in positive side it goes to infinity.
- Let's input number that are closer to zero but for the negative part of the function.
- For the negative part of this function it would go to negative infinity.



Limits

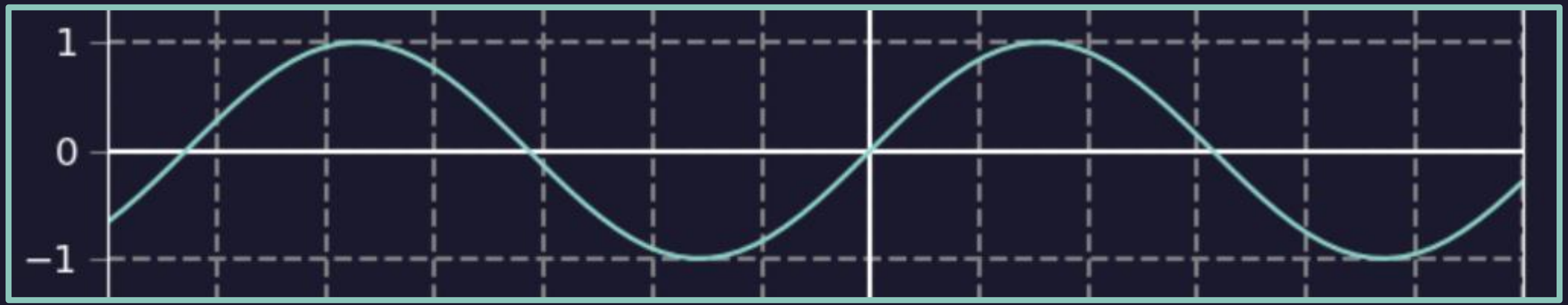
- $\lim_{x\to 0^+}\left(\frac{1}{x}\right) = \infty$ **for the positive side**
- $\lim_{x\to 0^-}\left(\frac{1}{x}\right) = -\infty$ **for the negative side**
- **What is the limit** $\lim_{x\to 0}\left(\frac{1}{x}\right) =?$
- **This *<u>limit doesn't exist</u>* for this function as the two parts of the functions are not approaching the same value.**



IEEE FCI - Helwan Student Branch

When doesn't limits exist?

- A limit doesn't exist if :
- The function behaves different from left and right.
- If the function oscillate between several values like (sin) and (cos)



- That only mean that the overall function like sin or cos doesn't converge to a single number, it oscillate between 1 and -1.

- This mean that the limit of the sin function as x approach infinity doesn't exist, but if x approach another value say a, the limit in this case would exist (equals sin(a))

Limits and continuity

- We can use the limits to decide if a function is conditions, that mean all inputs have possible outputs (defined everywhere).

- Function $f(x)$ is continuous if it satisfy three conditions.

- Function $f(a)$ is defined for all values of a.

- $\lim_{x \to \infty} f(x)$ exist, the function should approach the same value from both sides.

- $\lim_{x \to a} f(x) = a$, the limit and the actual value of the function should be the same.

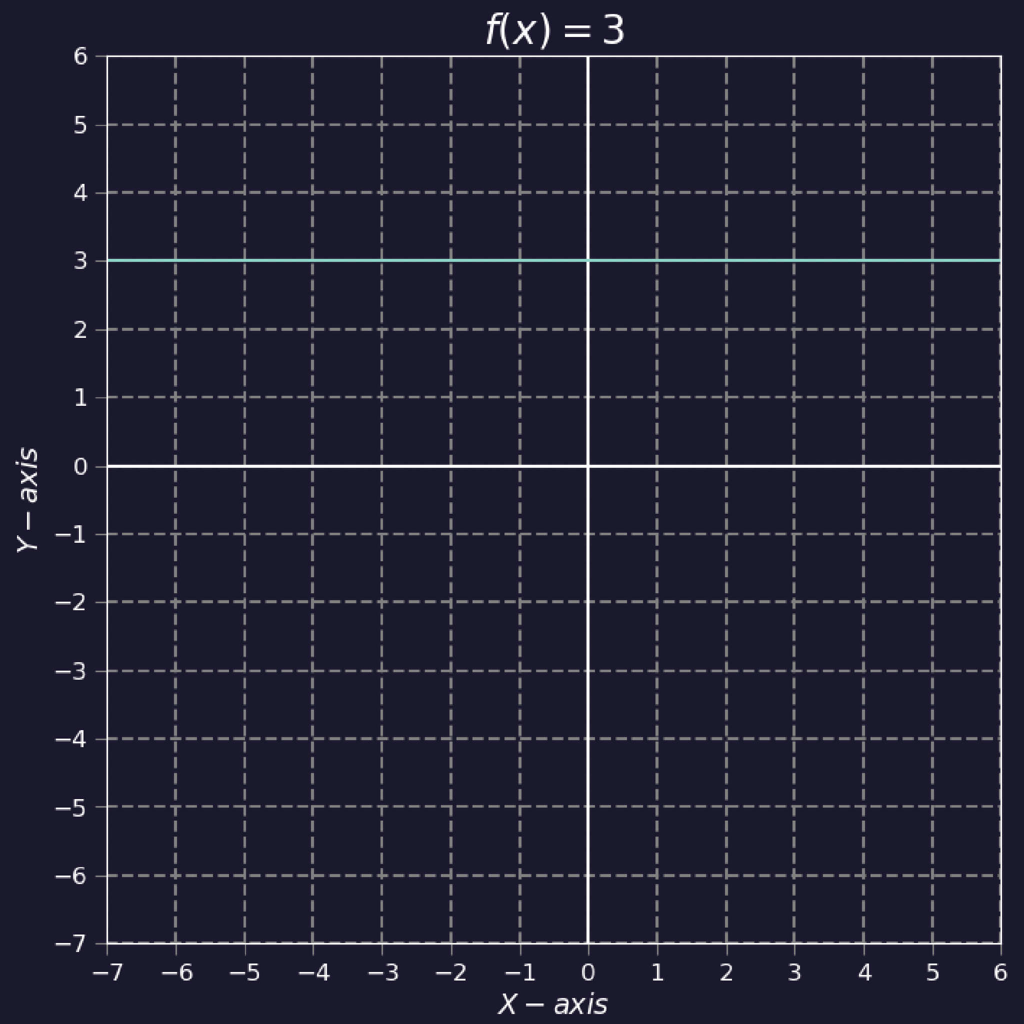
From limits to differentiation

- The derivative (differentiation) of a function $f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$

- Let's start by a simple function $f(x) = 3$ constant function.

- $f(x) = 3$ is given but $f(x+h)$ what is the value of h, it would be value that approach zero, it doesn't matter for this case as this function is a constant.

$$\lim_{h \to 0} \frac{f(x+h) - f(x)}{h} = \frac{3-3}{h} = \frac{0}{h} = 0$$



From limits to differentiation

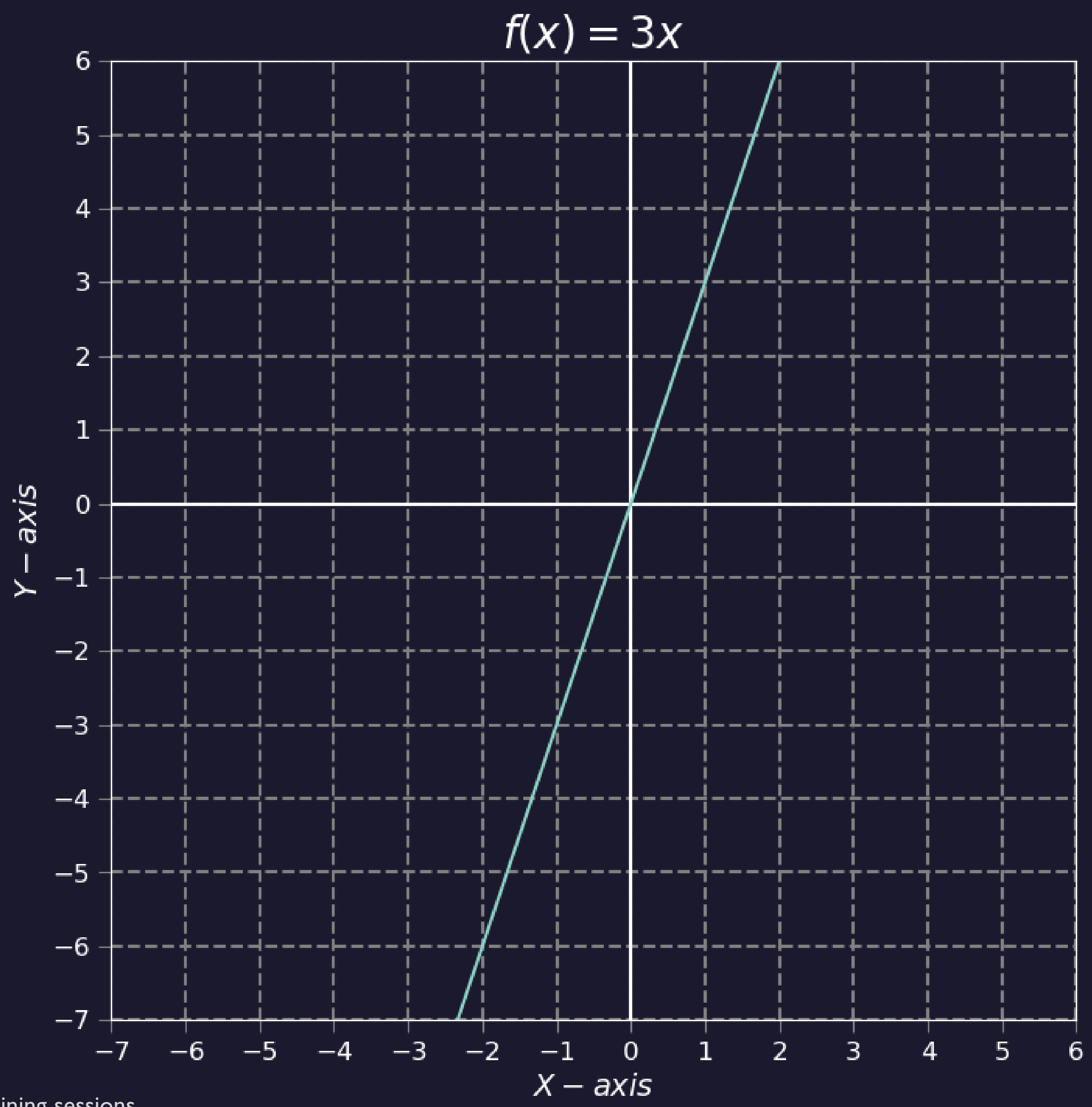
$$f'(x) = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$$

- function $f(x) = 3x$ constant function.

$$\lim_{h\to 0}\frac{3(x+h)-3x}{h}$$

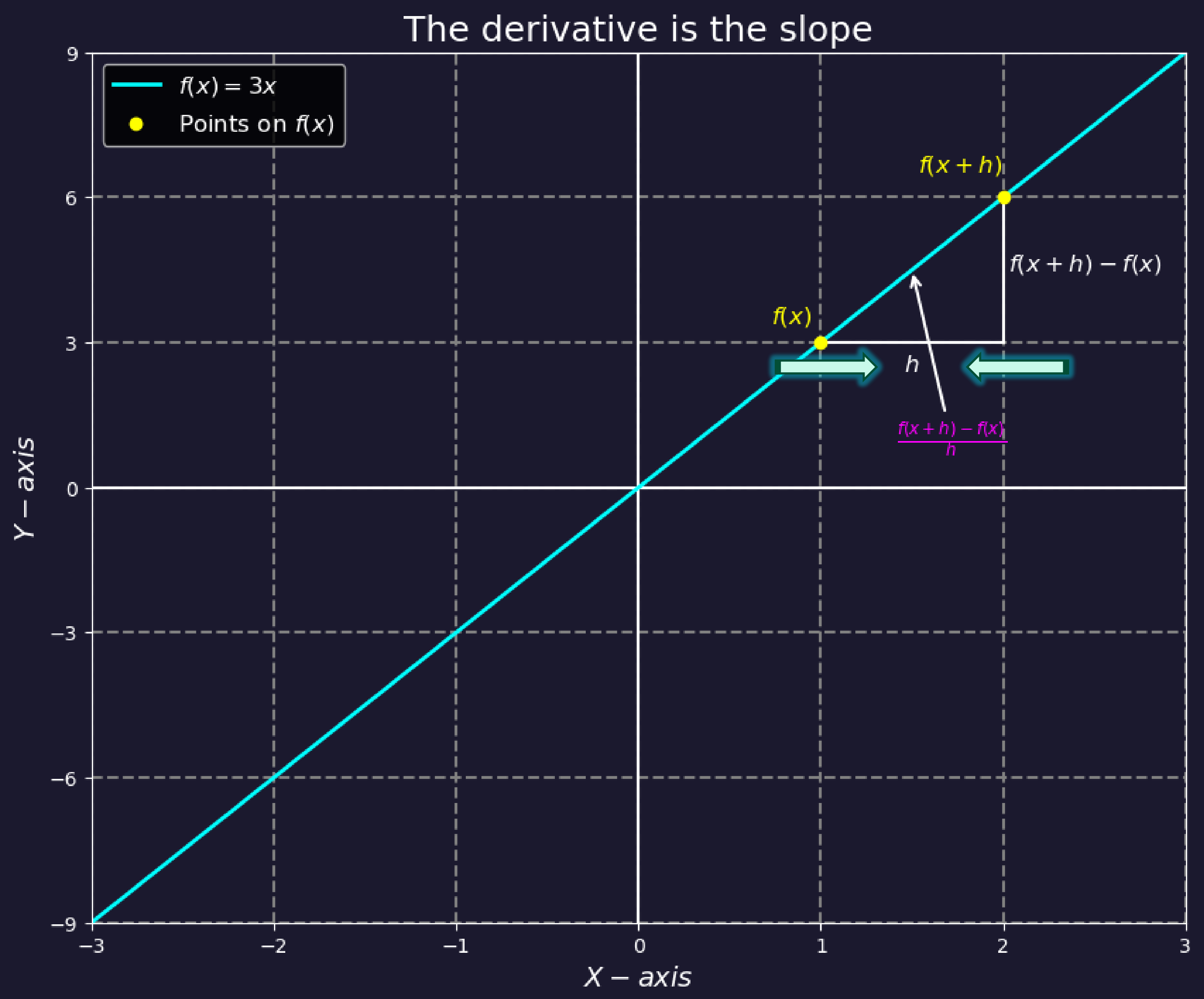
$$= \lim_{h\to 0}\frac{3x+3h-3x}{h}$$

$$= \lim_{h\to 0}\frac{3h}{h} = 3$$



- Derivative is the rate of change (difference) between the output of two inputs for the function these inputs are spaced by h.

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$



- We are trying to get the rate of change between two outputs.
- h is the distance between the inputs that produced these two outputs.
- h is a value that approach zero but would never be a zero.
- That is why the derivative is the slope.

From limits to differentiation

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

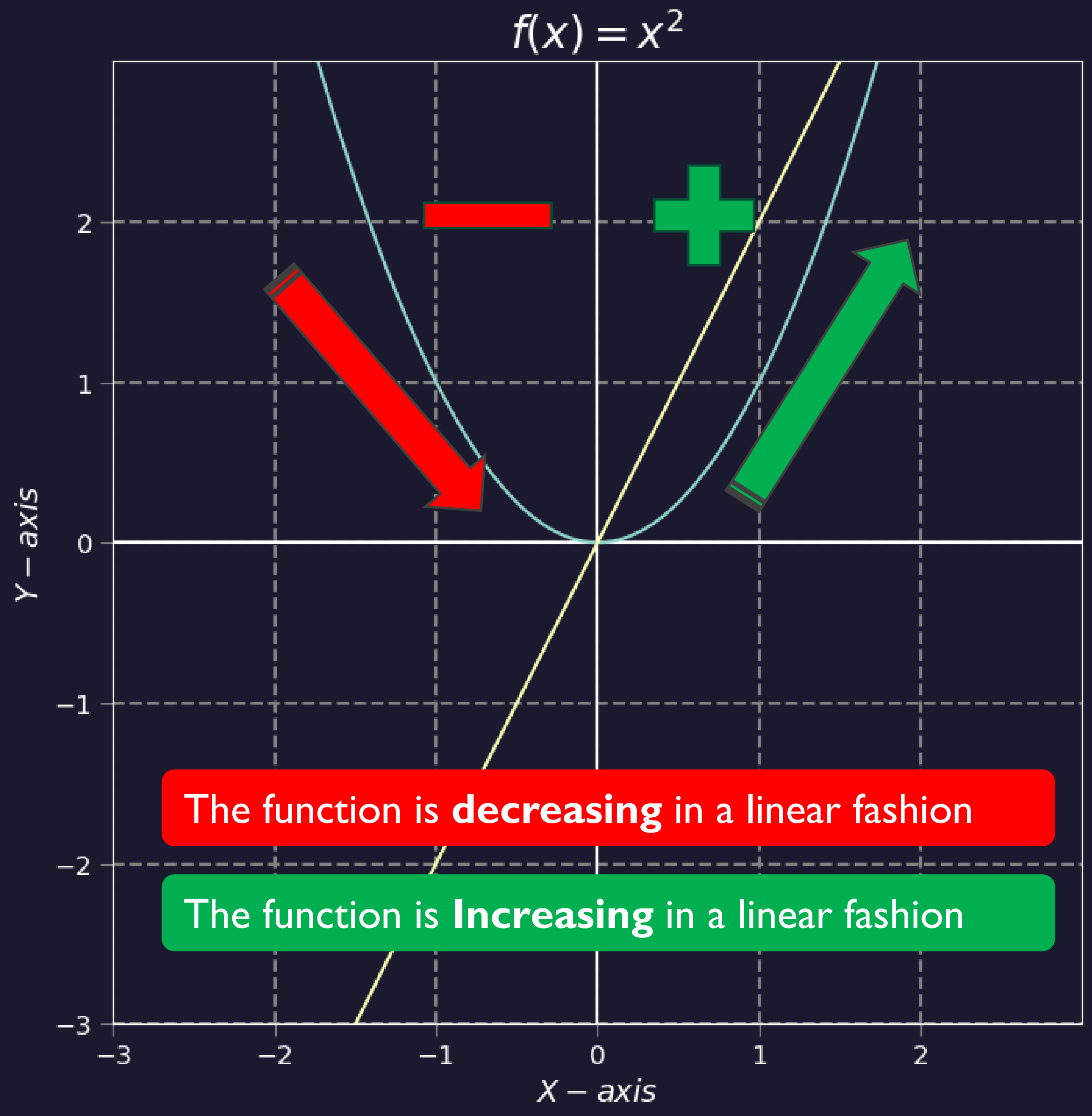
- function $f(x) = x^2$ quadratic equation.

$$\lim_{h \to 0} \frac{(x+h)^2 - x^2}{h}$$

$$= \lim_{h \to 0} \frac{x^2 + 2xh + h^2 - x^2}{h}$$

$$= \lim_{h \to 0} \frac{2xh + h^2}{h} = \frac{h(2x+h)}{h}$$

$$\frac{df(x)}{dx} = \lim_{h \to 0} 2x + h = 2x$$



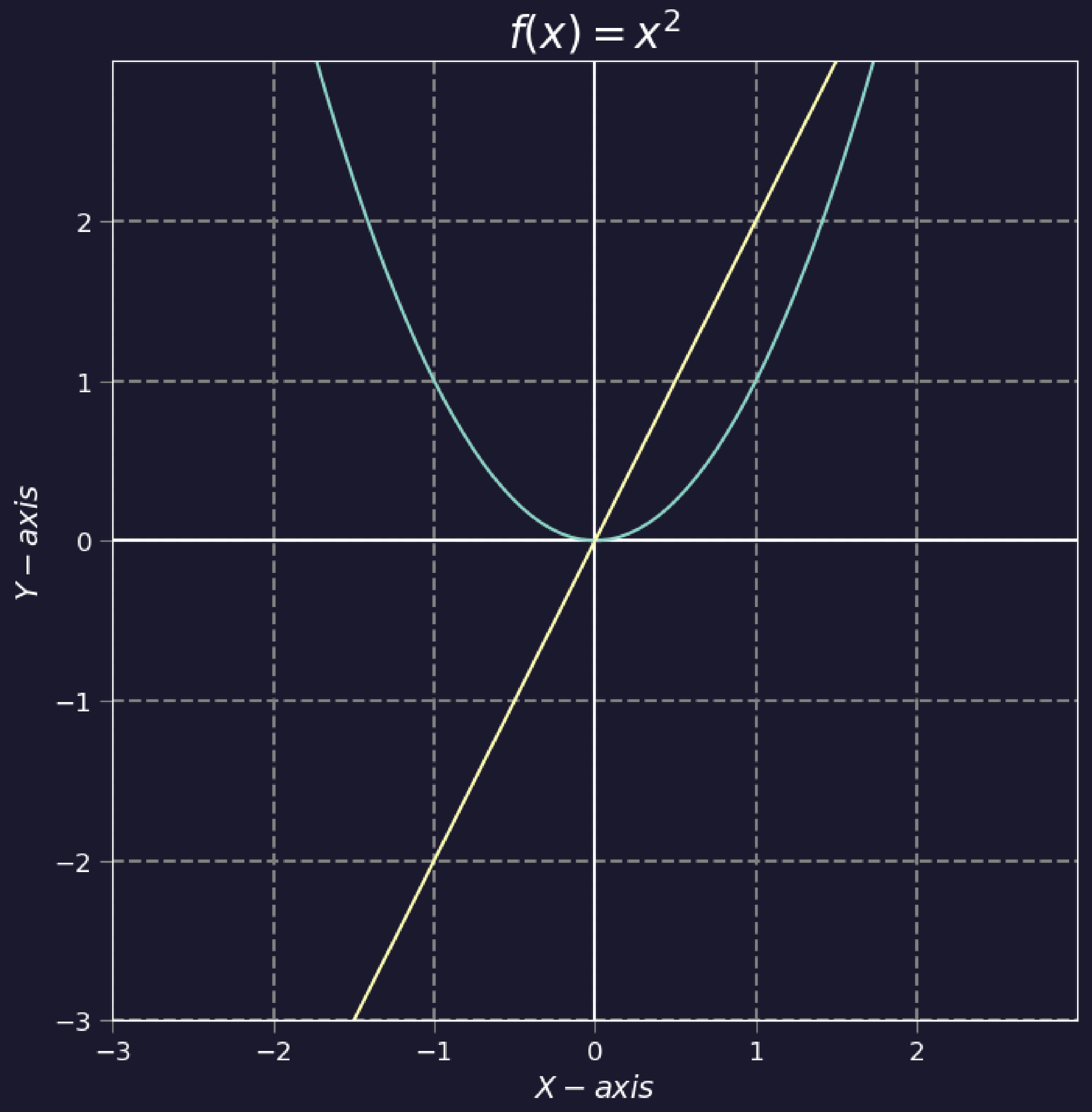
From limits to differentiation

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- function $f(x) = x^2$ quadratic equation.

```python
x = 3
def f(x): # x^2 -> 2x
    return x**2
h_1 = 0.01
h_2 = 0.001
h_3 = 0.0001
h_4 = 0.00001
print( (f(x+h_1) - f(x)) / h_1 )
print( (f(x+h_2) - f(x)) / h_2 )
print( (f(x+h_3) - f(x)) / h_3 )
print( (f(x+h_4) - f(x)) / h_4 )
```

```
6.009999999999849
6.000999999999479
6.000100000012054
6.000009999951316
```



Numerical differentiation

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

```
x = 3

def f(x): # x^2 -> 2x

    return x**2

h_5 = 0.0000000000000001

print( (f(x+h_5) - f(x)) / h_5 )
```

```
0.0
```

- This formula has an issue when handled to a computer.
- Computer represent numbers with finite precision (bits), when h is very small, $f(x+h)$ and $f(x)$ can become very close in value.

- **Subtracting two nearly equal values** leads to **catastrophic cancellation**, significant digits are lost, and the result is prune to round-offs errors.
- We can use something called Machine Epsilon $\epsilon_{machine} = (2.22 \times 10^{-16})$ which is smallest meaningful increment you can add to 1 without it being lost due to precision limitation.

Numerical differentiation

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- Machine Epsilon $\epsilon_{machine} = (2.22 \times 10^{-16})$ we can find it in NumPy.

```
epsilon_machine = np.finfo(float).eps
```

NumPy

- Let's check the definition, it's the smallest number we can add to one in the floating-point system.

```
print(1+h_5) #h_5 = 0.0000000000000001

print(1+epsilon_machine)
```

```
1.0
1.0000000000000002
```

- There are 4 types of errors that can interrupt our results and ruing it due to the limitations of machine representation of numbers.
  - Round-off error : computer can't represent numbers like π which has infinity digits.
  - Truncation error : the error when computers (or we) approximate numbers .
  - Underflow/ overflow : it's decreasing / growing beyond the representation capabilities.

Numerical differentiation

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- The first 2 errors may be ignorable at first, but they can **accumulate** causing more issues.

- **Round-off error,** computers use fixed number of bits to store real numbers
  - When you try to **represent a number with more precision** than the computer can handle, **it rounds the number** to fit into the available space.
  - This rounding process introduces small differences between the real number and its computer representation.

- **Truncation error,** Truncation error arises because you're cutting off part of the calculation to make the problem solvable.
  - This what happen in our derivative law, so we use a limited precision h.

- **How these Errors affect our derivative law?**

Numerical differentiation

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- **How these Errors affect our derivative law?**
- The difference $f(x+h) - f(x)$ can be smaller than the Machine Epsilon $\epsilon_{machine}$.
- When this happen, the computer might round the difference to zero or very small value, causing the derivative approximation to be incorrect (Round-off error).
- The result of the division $\frac{f(x+h)-f(x)}{h}$ may be truncated if h is too small. (Truncation error)
- Truncation error scale with h and at very small h, truncation error becomes negligible.
- Round-off error scale with $\frac{1}{h}$ and at very small h, round-off error dominates.
- So there is a trade-off, so we would use the **square root** of $\epsilon_{machine}$ as the h.

Numerical differentiation

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

```python
# Machine epsilon for double-precision floats
epsilon_machine = np.finfo(float).eps

# Optimal h
h_optimal = np.sqrt(epsilon_machine)

# Forward difference method
forward_diff = (f(x + h_optimal) - f(x)) / h_optimal

# Print the result
print(f"Forward Difference Approximation: {forward_diff}")
print(f"Expected Derivative: {2 * x}")  # Analytical derivative of x^2 is 2x
```

Forward Difference Approximation: 6.0

Expected Derivative: 6

Numerical differentiation

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- We can ***reduce*** the error also by modifying the law, this variation is called **symmetric difference formula.**

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x-h)}{2h}$$

- This $f(x+h) - f(x-h)$ this produce less error (truncation) but aren't immune to representational errors.

```
x = 3

def f(x): # x^2 -> 2x
    return x**2

h_5 = 0.000000000000001

print( (f(x+h_5) - f(x)) / h_5 )

print((f(x+h_5)-f(x-h_5))/ (2*h_5))
```

6.21724893790087 7

6.128431095930864

Differentiation Rules

- Constant rule : If $f(x) = C$, where C is a constant then $f'(x) = 0$
  - $f(x) = 6$ then $f'(x) = 0$
- Power rule : If $f(x) = x^n$, where n is a constant then $f'(x) = n.x^{n-1}$
  - $f(x) = x^4$ then $f'(x) = 4x^3$
- Constant Multiple rule: If $f(x) = C.g(x)$, where C is a constant then $f'(x) = C.g'(x)$
  - $f(x) = 3x^4$ then $f'(x) = 12x^3$
- Sum rule : If $f(x) = g(x) + h(x)$, then $f'(x) = g'(x) + h'(x)$
  - $f(x) = 3x^4 + 4x$ then $f'(x) = 12x^3 + 4$
- Product rule : If $f(x) = g(x).h(x)$, then $f'(x) = g'(x).h(x) + h'(x).g(x)$
  - $f(x) = (x-2)(x+3)$ then $f'(x) = (1)(x+3) + (1)\ (x-2) = 2x + 1$

Differentiation Rules

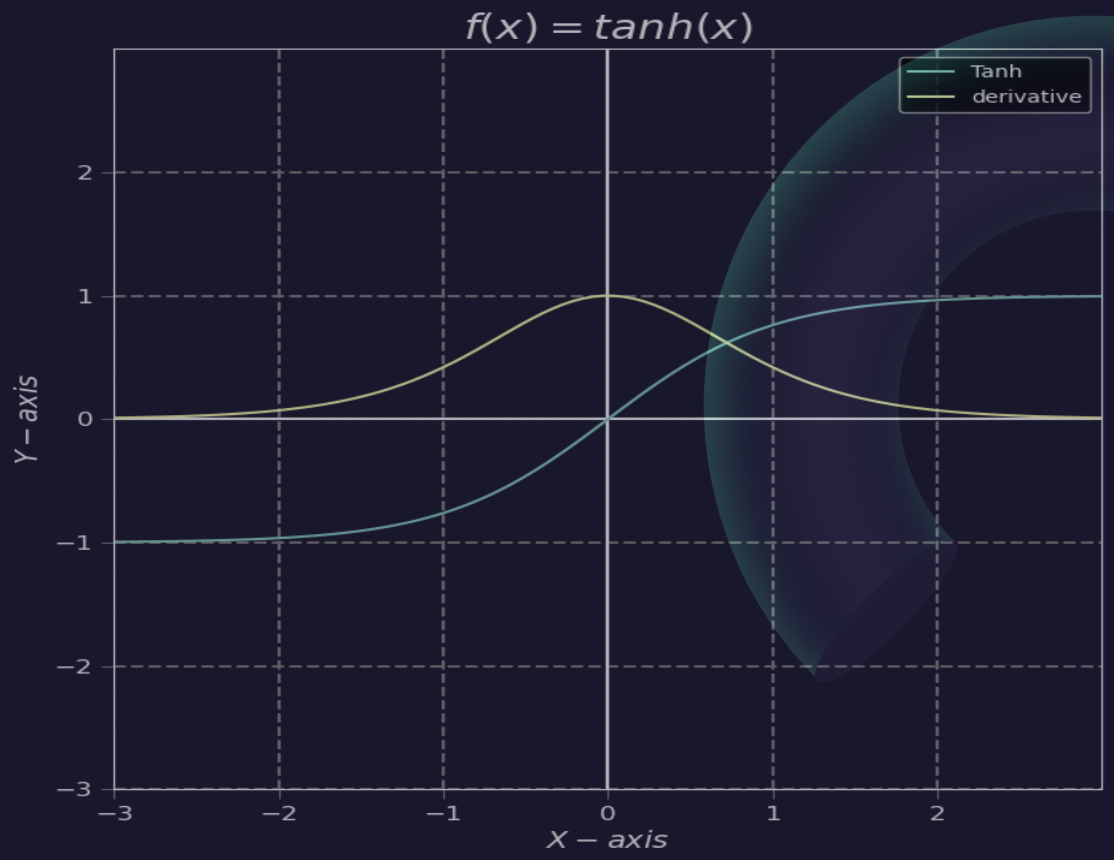
- Quotient rule : If $f(x) = \frac{g(x)}{h(x)}$, then $f'(x) = \frac{g'(x)h(x) - h'(x)g(x)}{h(x)^2}$
  - $f(x) = \frac{x^2}{x+1}$ then $f'(x) = \frac{2x(x+1) - 1x^2}{(x+1)^2}$

- Exponential functions : If $f(x) = e^x$, then $f'(x) = e^x$
- If $f(x) = a^x$, where C is a constant then, $f'(x) = a^x \, ln(a)$
  - If $f(x) = 2^x$ then $f'(x) = 2^x \, ln(2)$

- Logarithmic functions : If $f(x) = ln(x)$, then $f'(x) = \frac{1}{x}$
- If $f(x) = log_a(x)$, where a is a constant then $f'(x) = \frac{1}{x ln(a)}$
  - If $f(x) = log_a(2x^2 + 4x)$, then $f'(x) = \frac{4x+4}{(2x^2+4x)ln(a)}$

Differentiation Rules

- $\frac{d}{dx}\left(\sin(x)\right) = \cos(x)$
- $\frac{d}{dx}\left(cos(x)\right) = -\sin(x)$
- $\frac{d}{dx}\left(tan(x)\right) = \sec^2(x)$



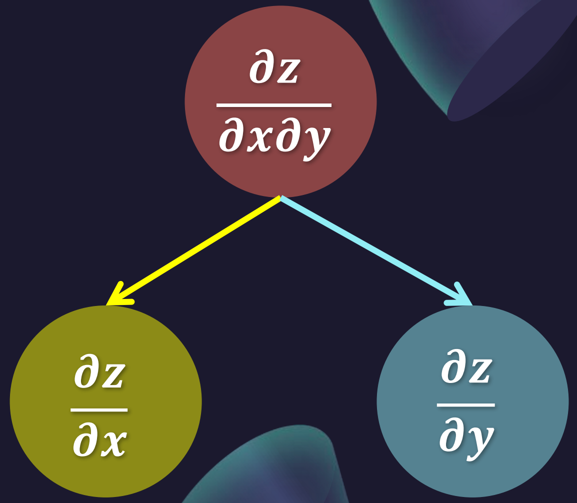
- $\frac{d}{dx}\left(tanh(x)\right) = \frac{d}{dx}\left(\frac{sinh(x)}{cosh(x)}\right) = 1 - \tanh^2(x) = \text{sech}^2(x)$

- Remember that the range of the output for the trigonometric functions is between -1 and 1.

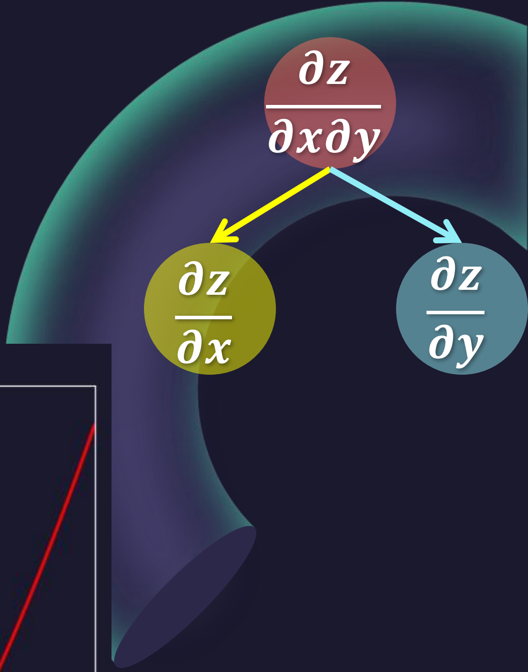
Partial derivative

- Partial derivative of a function of several variables is its derivative with respect to one of those variables, with the other held constant.
  - Notation $\frac{\partial f}{\partial x}$

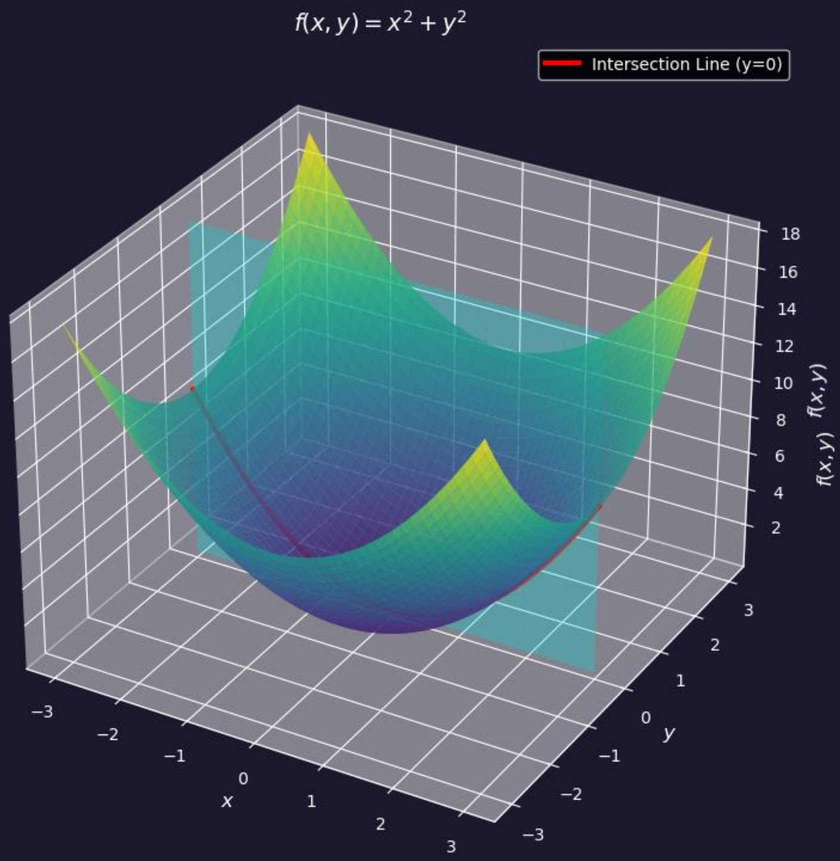
- $z = f(x, y) = x^2 + y^2$
  - $\frac{\partial z}{\partial x} = 2x + \frac{\partial}{\partial x}(y^2) = 2x + 0 = 2x$
  - $\frac{\partial z}{\partial y} = \frac{\partial}{\partial y}(x^2) + 2y = 0 + 2y = 2y$
  - $\frac{\partial z}{\partial x \partial y} = \frac{\partial z}{\partial x} + \frac{\partial z}{\partial y}$

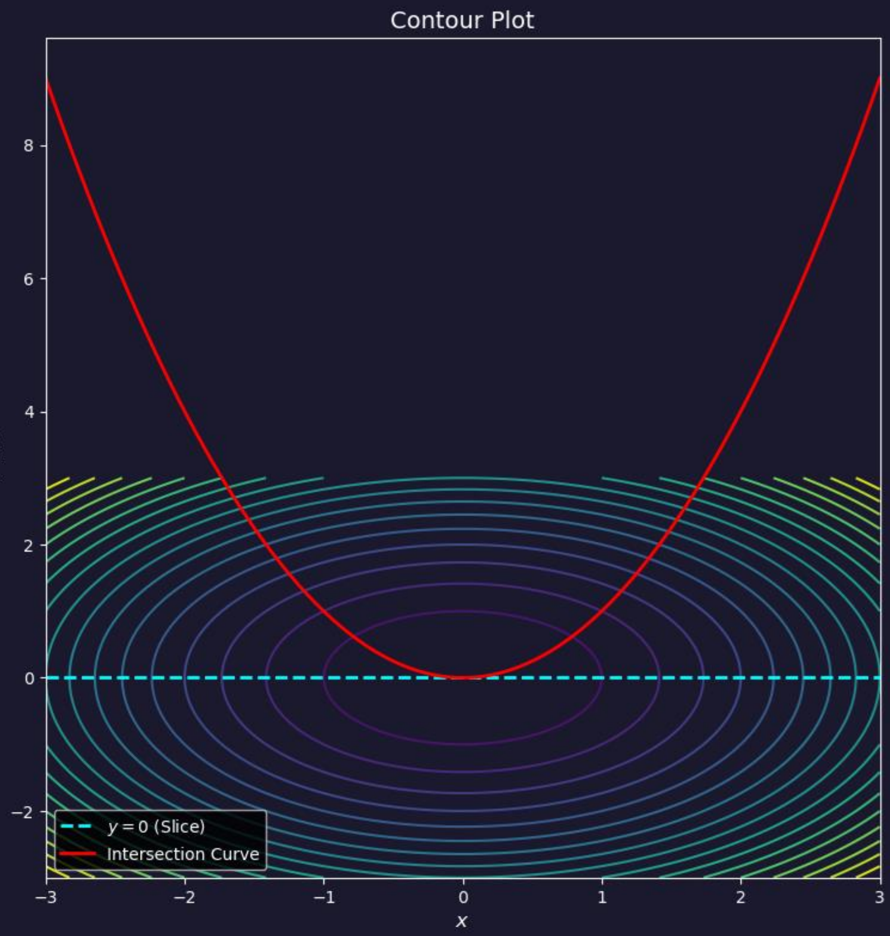


Partial derivative



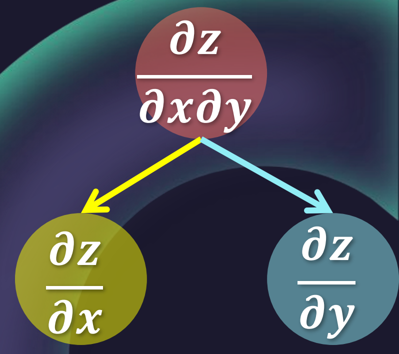
- When you set $y = 0$ (as we are considering it constant), you are fixing one variable in the function $z = f(x, y)$ reducing it to single variable function $f(x, y) = x^2 + y^2 = x^2, \rightarrow f(x, 0) = x^2$

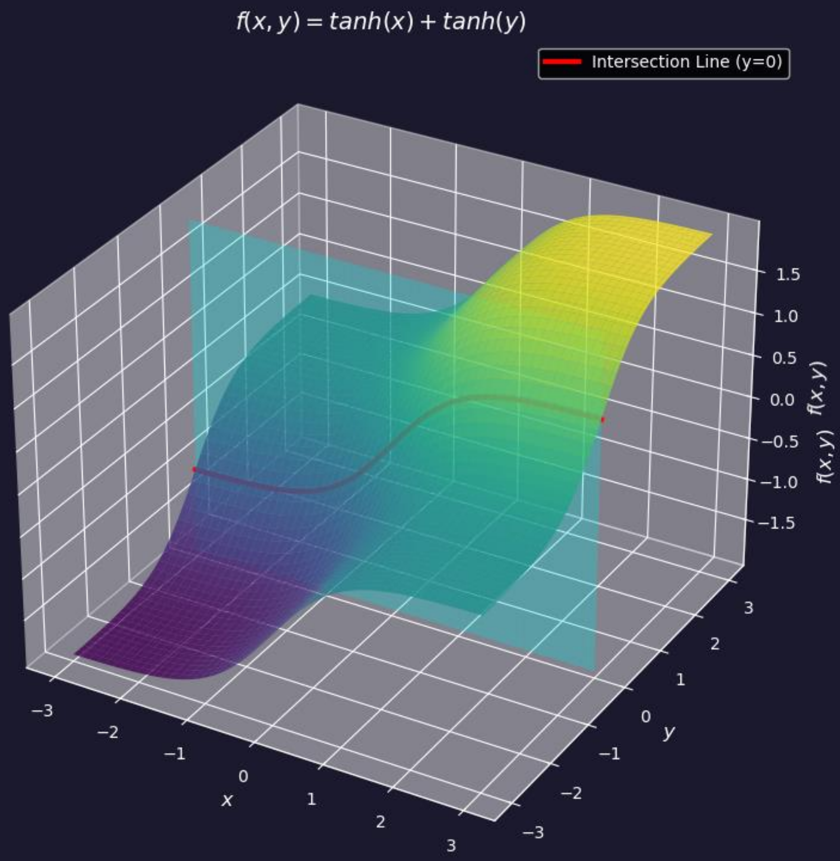


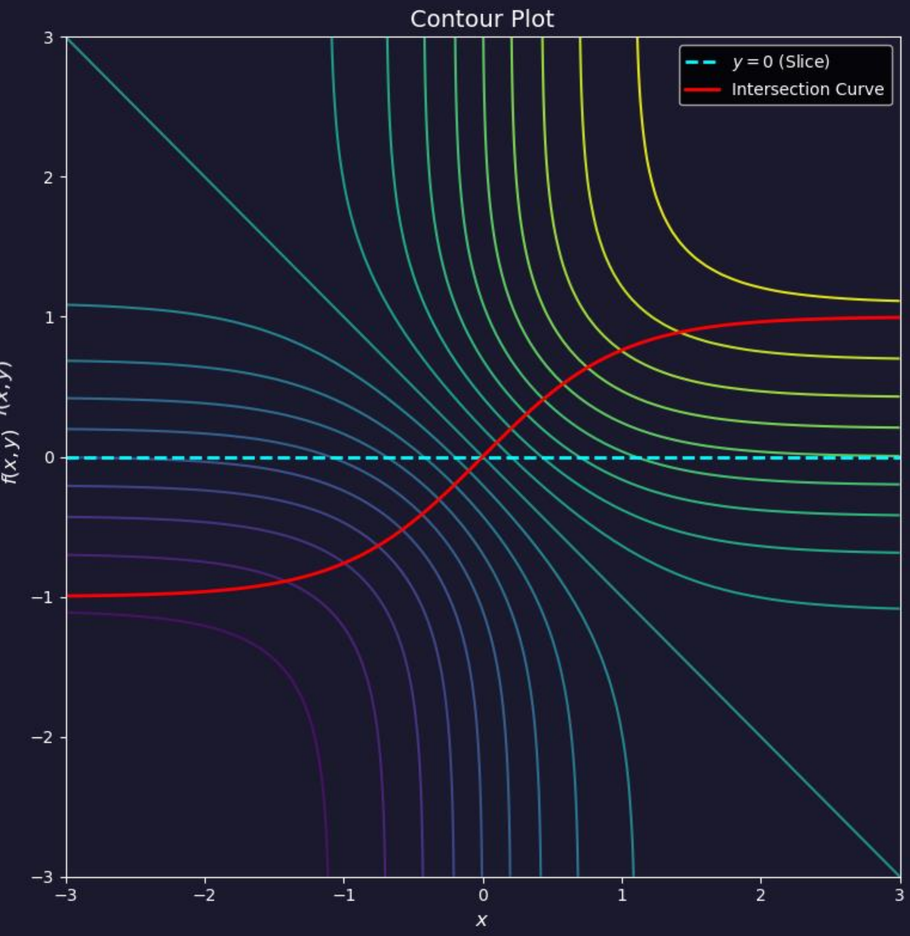


Partial derivative

- Another example on the function $f(x, y) = tanh(x) + tanh(y)$

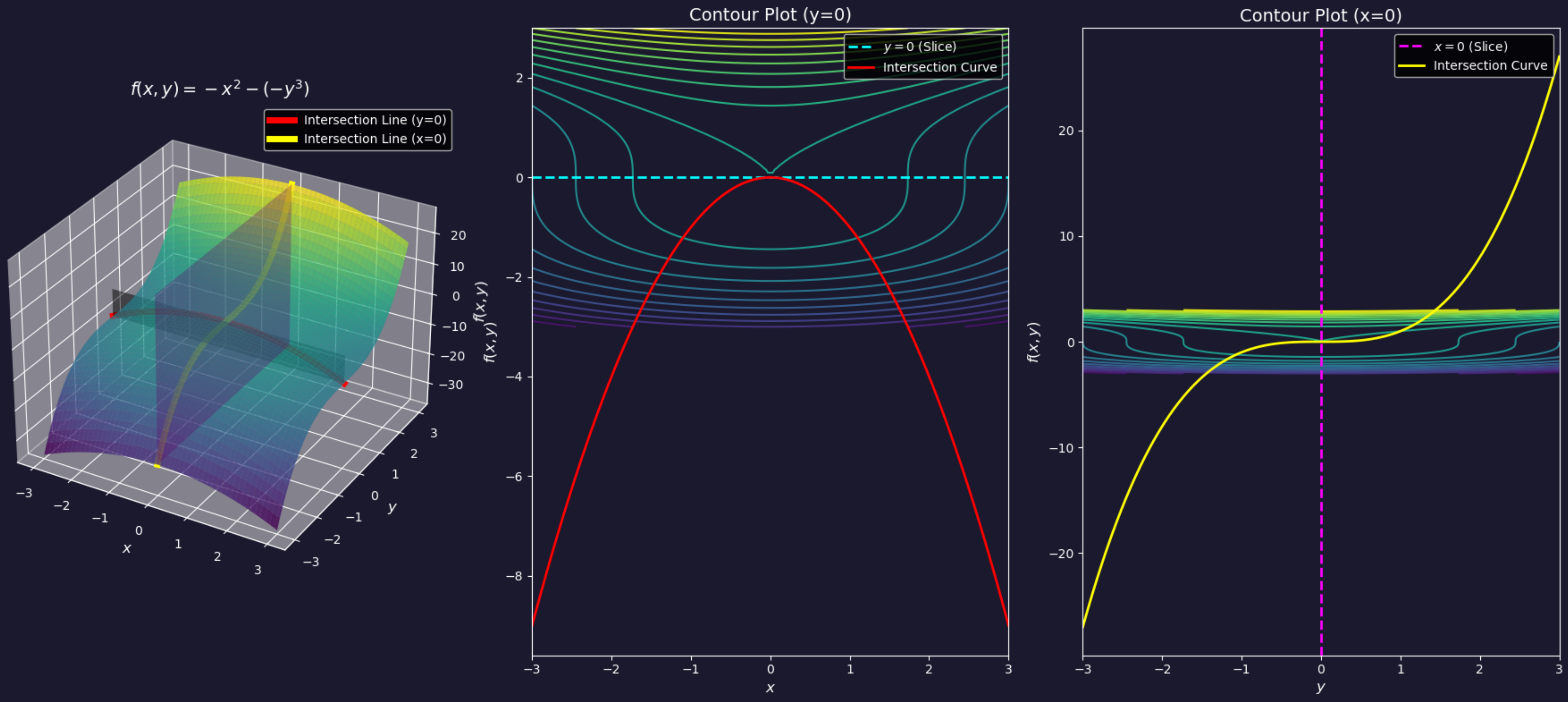






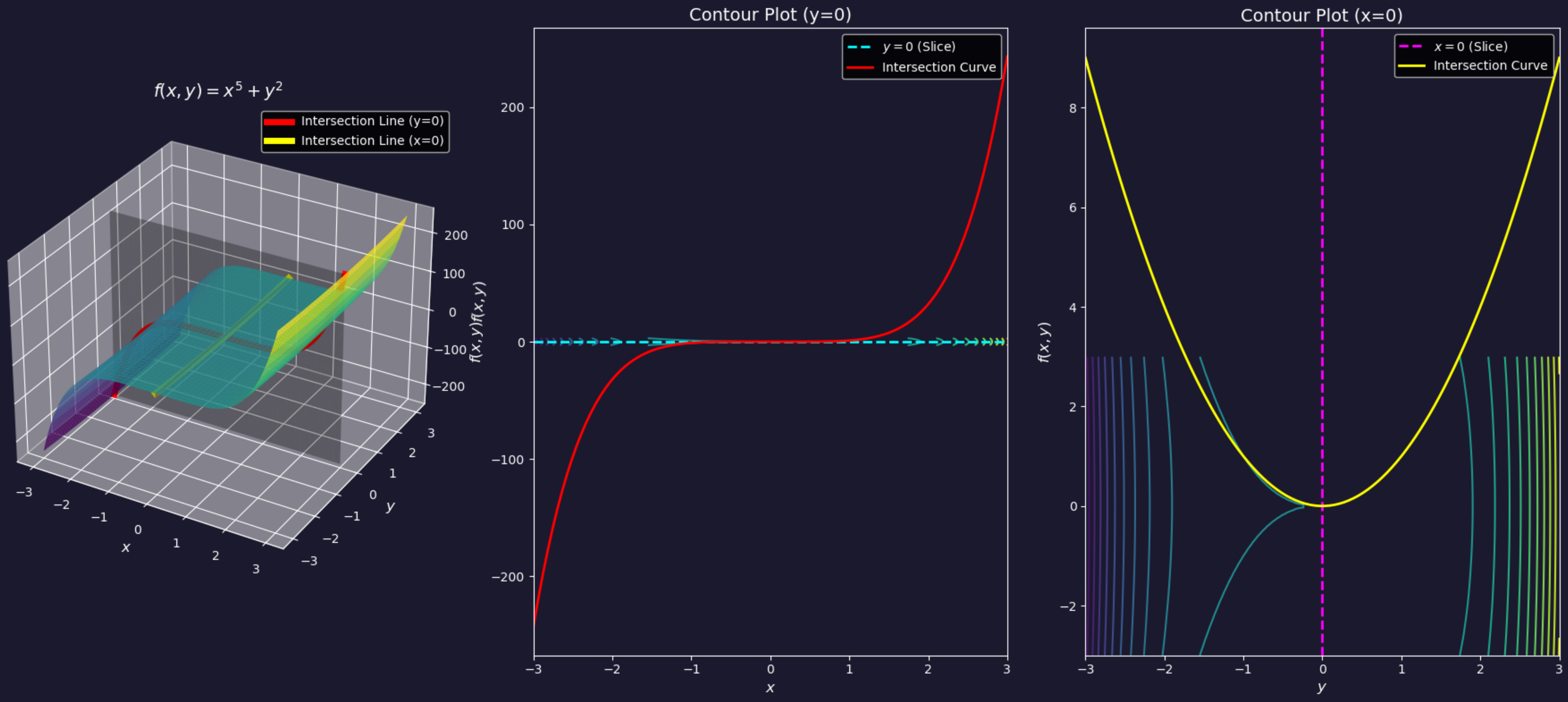
Partial derivative

- Another example on the function $f(x, y) = -x^2 - (-y^3)$



Partial derivative

- Another example on the function $f(x, y) = x^5 + y^2$



Chain Rule

- Chain rule is a formula that express the derivative of the composition of two differentiable functions f and g in terms of the derivatives of f and g.
  - $h(x) = f(g(x))$
  - $h'(x) = f'(g(x))g'(x)$ another notation $\frac{d}{dx}[f(g(x))] = \frac{df}{dg} * \frac{dg}{dx}$
- $h(x) = sin(3x)$, the outer function $f(u) = sin(u)$,the inner function $g(x) = 3x$
  - $\frac{d}{dx}[sin(3x)] = \frac{d}{du}[sin(u)] * \frac{d}{dx}[3x] = cos(u) * 3 = cos(3x) * 3$
- The chain rule arise from the existence of chain of dependencies between some functions, x depends on y and y depends on z and so on.
- Let's expand our notation, if we have a composition of many functions $f_1\left(f_2(\ldots f_n(x))\right)$, the derivative is $\frac{d}{dx}\left[f_1\left(f_2(\ldots f_n(x))\right)\right] = \frac{d}{d\,f_1} * \frac{df_1}{df_2} * \cdots * \frac{df_{n-1}}{df_n}$

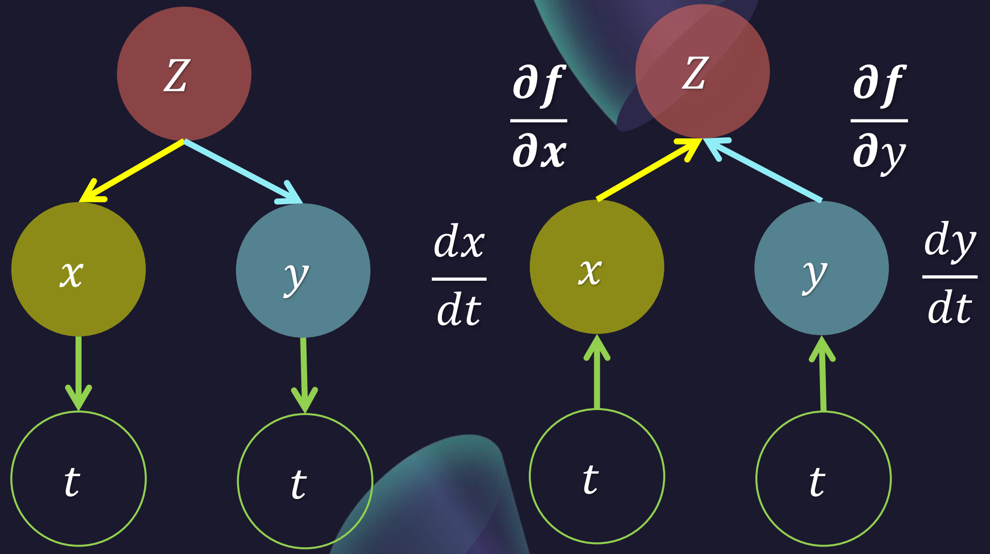
Chain Rule

- Calculate $\frac{dz}{dt}$ given the following functions, express the final output in terms of t
  - $z = f(x, y) = x^2 - 3xy + 2y^2$
  - $x = x(t) = 3\, sin(2t)$
  - $y = y(t) = 4\, cos\,(2t)$

$$\frac{dz}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt}$$

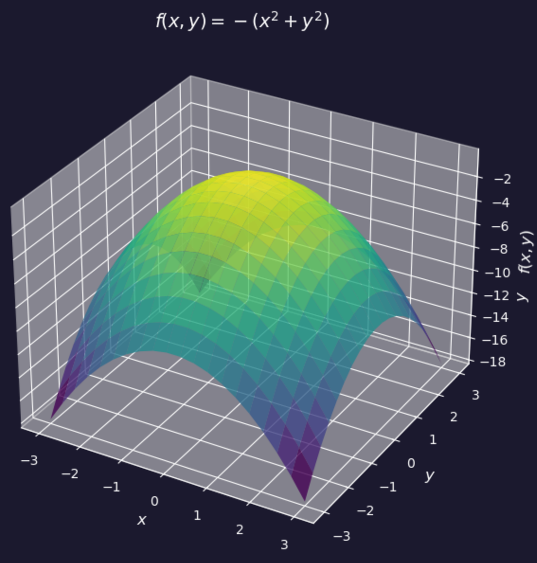
$$\frac{dz}{dt} = (2x - 3y)(6\, cos\, 2t) + (-3x + 4y)(-8\, sin\, 2t)$$

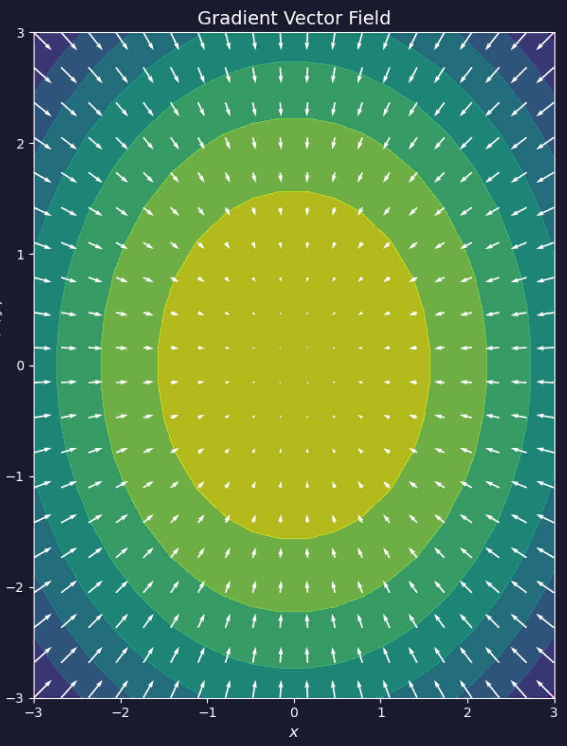
$$= -64\, sin\, 4t - 72\, cost\, 4t$$

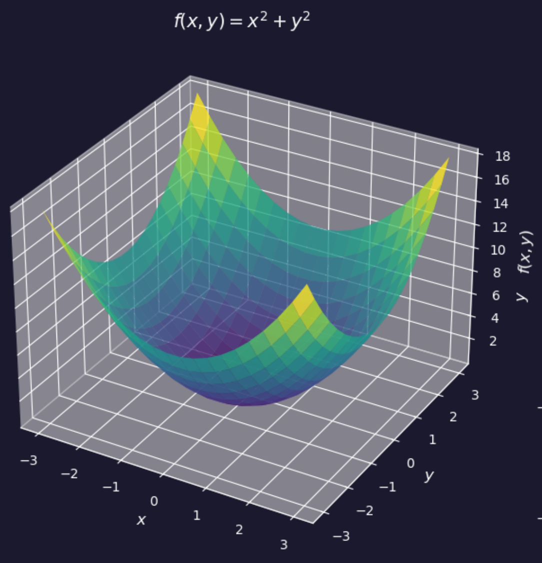


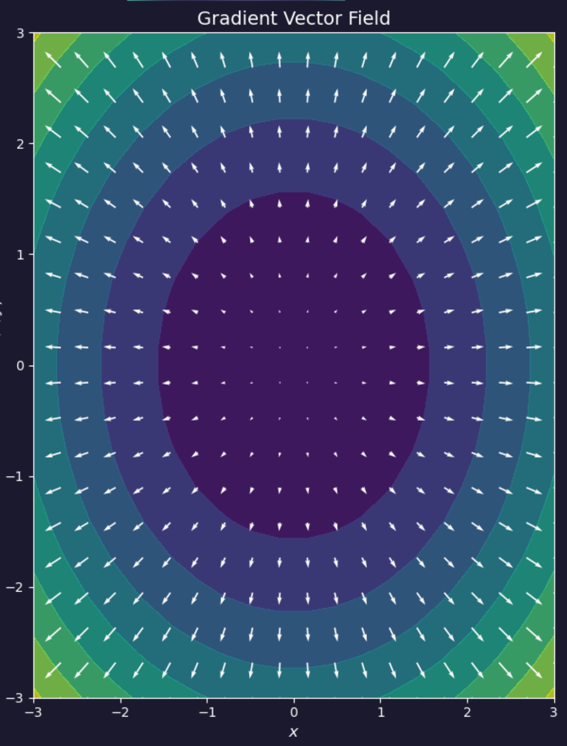
Gradient

- The gradient of a scaler function $f(x, y)$ is a vector field that points in the **direction of the greatest rate of change** of f , for a function $f(x, y)$ the gradient is defined as :
  - $\nabla f(x, y) = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y})$ vector of partial derivatives.

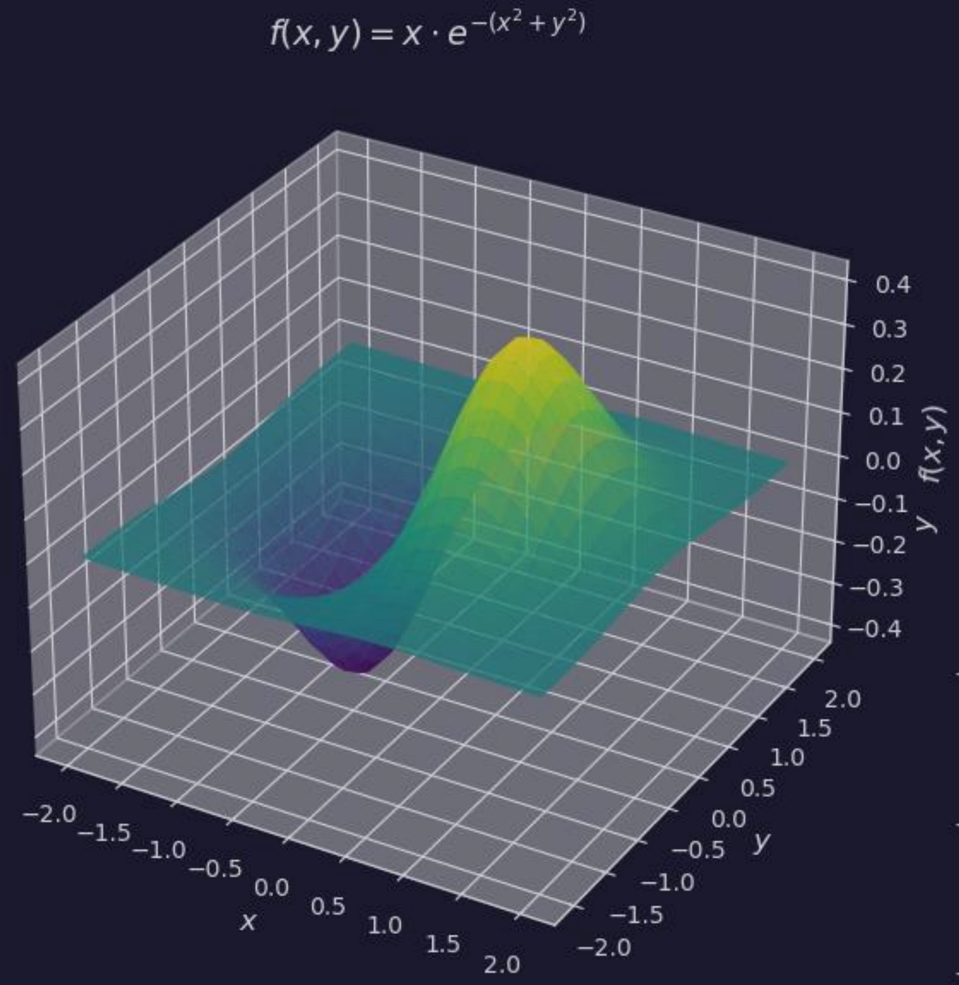


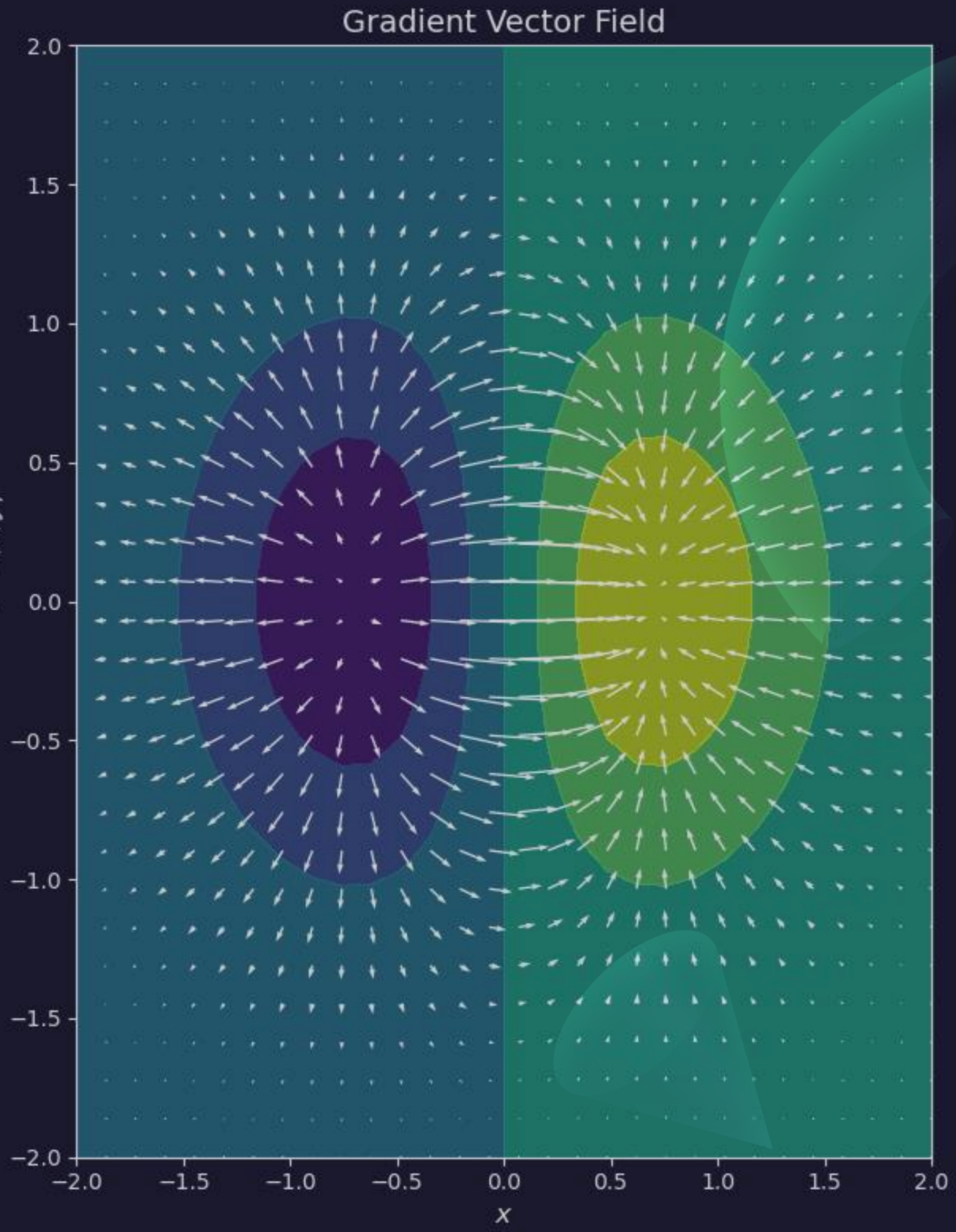






Gradient


$f(x,y) = x \cdot e^{-(x^2+y^2)}$
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
$f(x,y)$
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x
y


Gradient Vector Field
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
y
x

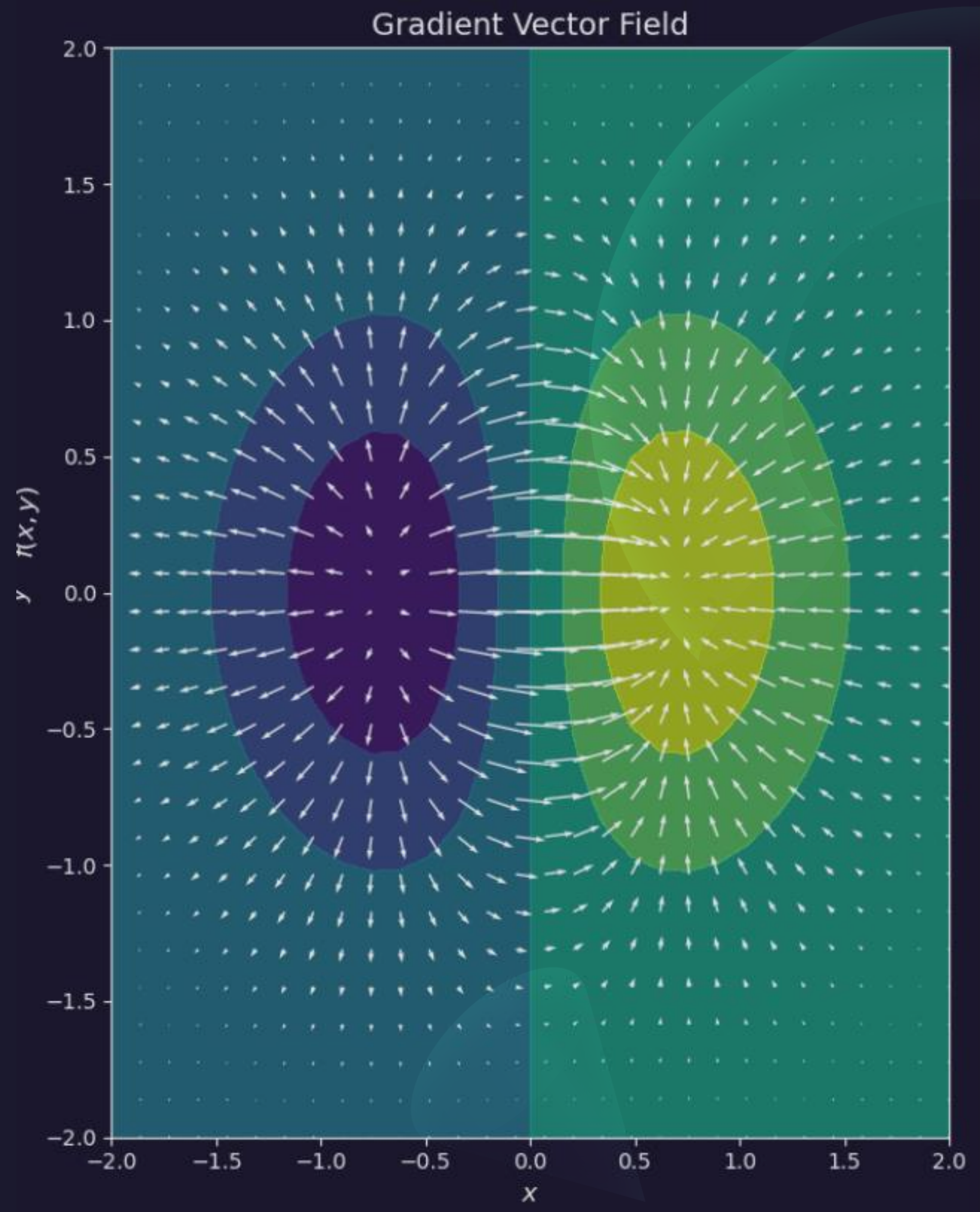
Gradient

$$f(x, y, z) = 2x + 3y^2 - sin(z)$$
$$\nabla f(x, y, z) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right)$$
$$\nabla f(x, y, z) = \big(2, 6y, -cos(z)\big)$$

- We can write the gradient as a row vector or column vector.
- To generalize the definition, for a function $f(x, y, \dots)$, the gradient ∇f is a vector that point towards the direction of the steepest increase of f.
  - $\nabla f(x, y, \dots) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \dots\right)$
  - Since the gradient is a vector, it has a direction and a magnitude represented by the arrows we plotted in the vector field.
  - These vectors represent the steepest ascent, and the magnitude tell us how fast the function increase in that direction.

Gradient

- How we know the direction of the arrow (gradient vector at specific point) ?
- The direction is $\theta = tanh^{-1}\left(\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}}\right)$
- The magnitude is $\|\nabla f\| = \sqrt{\sum_{i=1}^{n}\left(\frac{\partial f}{\partial x_i}\right)^2}$



Beyond gradient

- **Jacobian matrix** of vector valued function of several variables is the matrix of its all fist-order partial derivatives.

- If we have a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$, the Jacobian matrix $J \in \mathbb{R}^{n\times m}$ is defined as :

$$J_{i,j} = \frac{\partial}{\partial x_j} f(x)_i$$

$$J_{i,j} = \frac{\partial}{\partial x_j} f(x)_i = \left[\frac{\partial f}{\partial x_1} \cdots \frac{\partial f}{\partial x_n}\right] = \begin{bmatrix} \nabla^{\mathrm{T}} f_1 \\ \vdots \\ \nabla^{\mathrm{T}} f_m \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

- $\nabla^{\mathrm{T}} f_i$ is the transpose of (row vector) of the gradient of the $i-\mathrm{th}$ component .

Optimization

- Optimization is the selection of the best element, with regards to some criteria, from a set of available alternatives.
- Optimization problem consist of maximizing or minimizing a real function by systematically choosing input values from within allowed set and computing the value of the function.
- How it this related to machine learning and calculus ?
- In ML the model has some parameters (elements) we need to select the best that maximize its performance and minimizing its mistakes.
- What makes ML special is the use of data to approximate some function using the parameters that minimize the Error and maximize its accuracy.

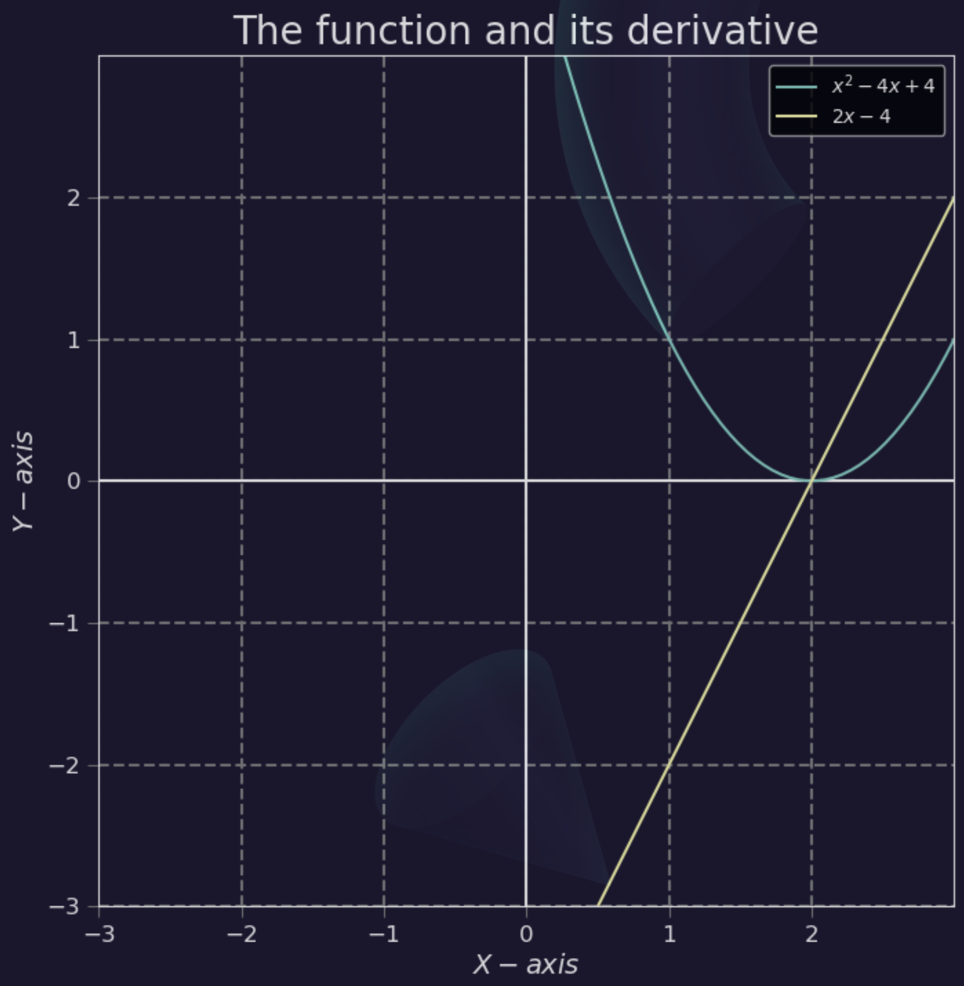
Optimization

- Derivative can be a good tool for finding the maximum or minimum values based on finding the points where the slope is zero, we call them critical points.

- $f(x) = x^2 - 4x + 4$, the derivative (slope) $f'(x) = 2x - 4$, when the slope would equal zero?

- $f'(x) = 2x - 4$, the question is when $f'(x) = 0$
  - $2x - 4 = 0$
  - $x = \frac{4}{2} = 2$

- For ML we are searching for a function, and we don't have the perfect function to minimize directly.



Optimization and the learning problem

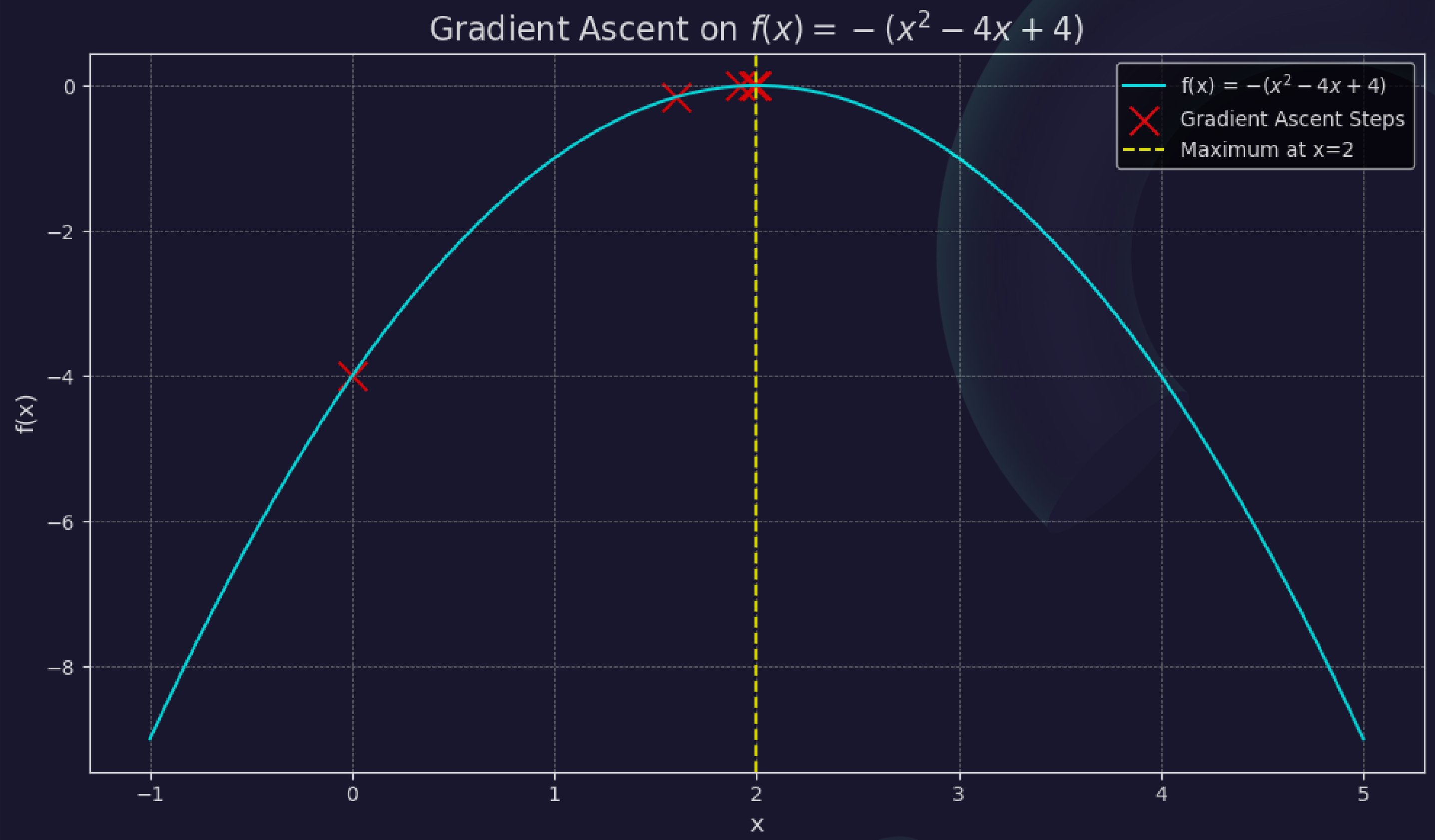
- In machine learning we try to minimize the error, and to minimize it we need to measure it.
- By the nature of the learning problem, you don’t have all the possible errors to represent as a function, so how we know the error?
- We can know or measure the error at some point using the training data.
- Error function is a way to quantify the error by comparing our model outputs to the data points we have.
- We have unknown function to learn and error function that is known only when we have a function and data points to compare to, this would enforce use to learn iteratively in most cases.

Gradient Ascent for maximizing

- Do you remember how the vector field of the gradient was pointing to the maxima points?
- We would *maximize* this function $f(x) = -(x^2 - 4x + 4)$, its derivative is $f'(x) = -(2x - 4)$
- In the initial step we would guess a value for x let's say 0
- We would update the value of x using this formula $x_{new} = x_{old} + \eta\, f'(x_{old})$
  - η is the step size (learning rate), how much we want to go in this direction.
- This algorithm work with iterative approach so we can set number of steps (iteration) also the size of the step η in each iteration.
  - For simplicity we would simulate 5 iterations.
  - $\eta = 0.4$

Gradient Ascent for maximizing

$$x_{new} = x_{old} + \eta\, f'(x_{old})$$



- Step 1
  - $x_{new} = x_{old} + \eta\, f'(x_{old})$
  - $x_{new} = 0 + 0.4\,(-(2(0) - 4))$
  - $x_{new} = 1.6$
- Step 2
  - $x_{new} = 1.6 + 0.4\,(-(2(1.6) - 4))$
  - $x_{new} = 1.92$
- Step 3
  - $x_{new} = 1.92 + 0.4\,(-(2(1.92) - 4))$
  - $x_{new} = 1.984$
- Step 4
  - $x_{new} = 1.984 + 0.4\,(-(2(1.984) - 4))$
  - $x_{new} = 1.9968$
- Step 5
  - $x_{new} = 1.9968 + 0.4\,(-(2(1.9968) - 4))$
  - $x_{new} = 1.99936$

Gradient Descent for minimizing

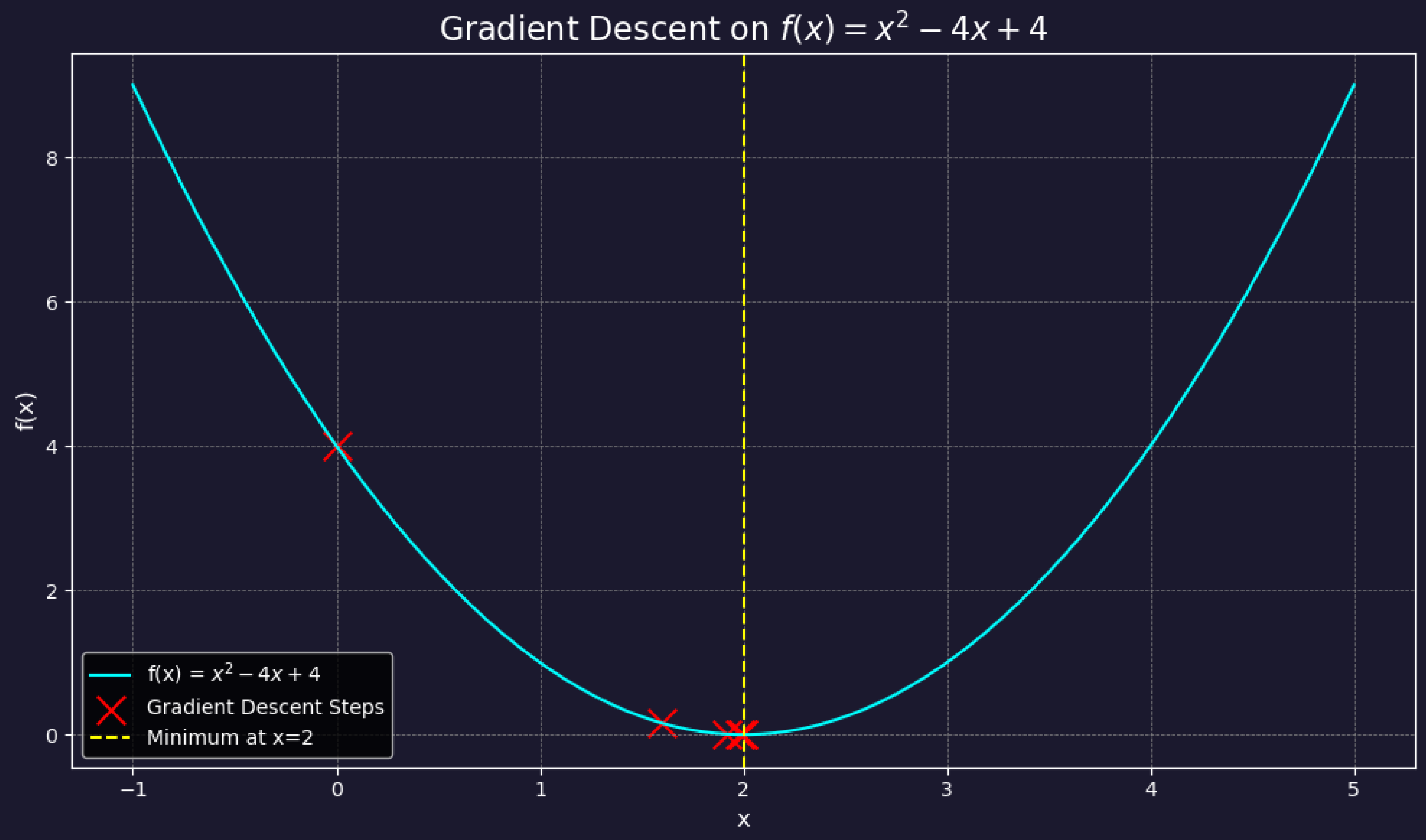
- We would *minimize* this function $f(x) = x^2 - 4x + 4$, its derivative is $f'(x) = 2x - 4$
- In the initial step we would guess a value for x let's say 0
- We would update the value of x using this formula $x_{new} = x_{old} - \eta\, f'(x_{old})$
  - η is the step size (learning rate), how much we want to go in this direction.
- This algorithm work with iterative approach so we can set number of steps (iteration) also the size of the step η in each iteration.
  - For simplicity we would simulate 5 iterations.
  - $\eta = 0.4$
- Notice the we are going in the negative direction of the gradient (derivative) to minimize the function.

IEEE FCI - Helwan Student Branch

Gradient Descent for minimizing

$$x_{new} = x_{old} - \eta\, f'(x_{old})$$

- Step 1
  - $x_{new} = x_{old} - \eta\, f'(x_{old})$
  - $x_{new} = 0 - 0.4\,(2(0) - 4)$
  - $x_{new} = 1.6$
- Step 2
  - $x_{new} = 1.6 - 0.4\,(2(1.6) - 4)$
  - $x_{new} = 1.92$
- Step 3
  - $x_{new} = 1.92 - 0.4\,(2(1.92) - 4)$
  - $x_{new} = 1.984$
- Step 4
  - $x_{new} = 1.984 - 0.4\,(2(1.984) - 4)$
  - $x_{new} = 1.9968$



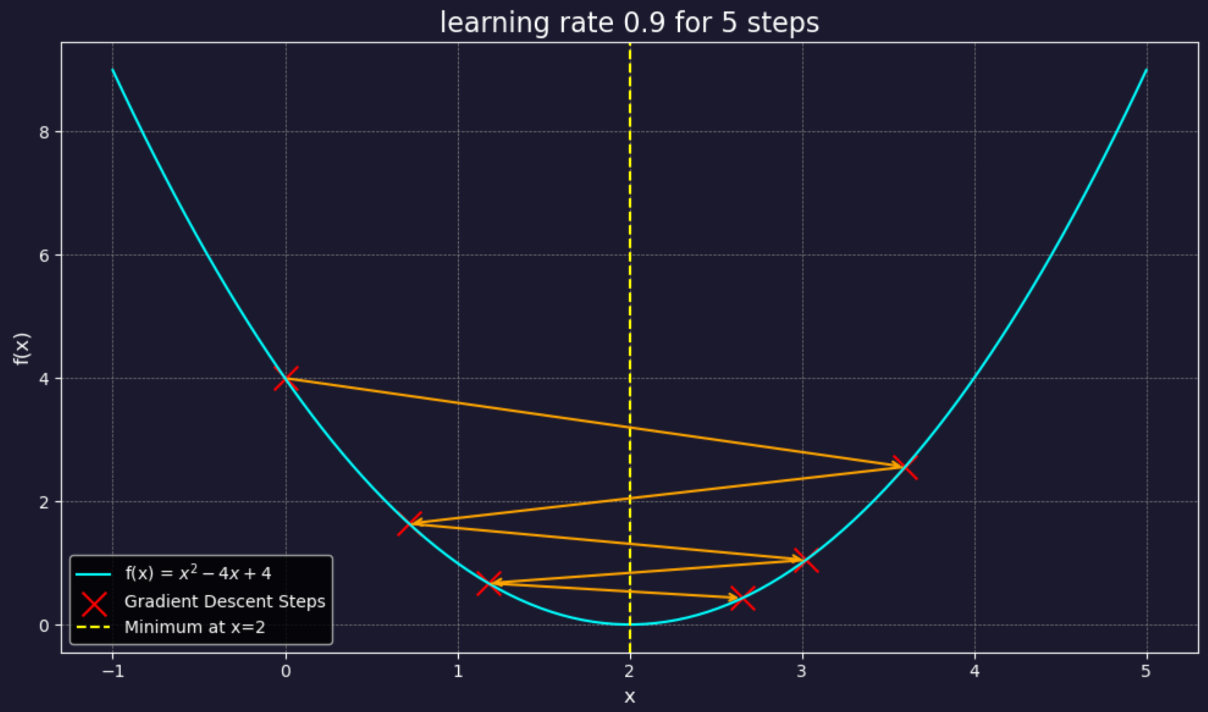
- Step 5
  - $x_{new} = 1.9968 - 0.4\,(2(1.9968) - 4)$
  - $x_{new} = 1.99936$

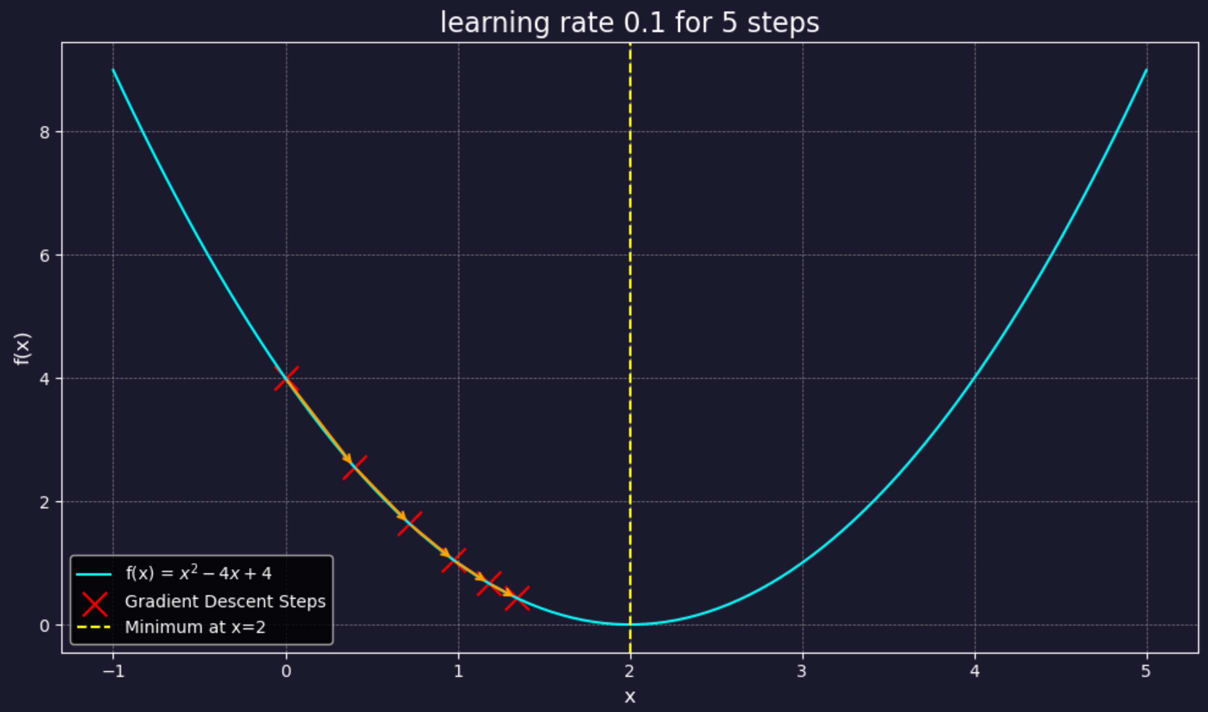
Gradient Ascent ⬆️ vs Gradient Descent ⬇️

Aspect	Gradient Descent	Gradient Ascent
Objective	Minimize a function	Maximize a function
Direction	Negative gradient	Positive gradient
Formula	$x_{new} = x_{old} - \eta f'(x_{old})$	$x_{new} = x_{old} + \eta f'(x_{old})$
in ML	Used to minimize the loss function.	Used to maximize the reward function in reinforcement learning.

The effect of learning rate η

- Learning rate (step size) is a critical parameter in gradient ascent/descent.
- We need to make the learning rate high so we can reach the best point in fewer steps.
- But if we made η a very large number we may miss our goal, too small number is bad also.





See 👀

- https://youtu.be/TglD4Y6lmQk?si=uLiClQDrSdXOB7oQ (7h ⏱️ of Limits problems, if you want to practice)
- https://youtu.be/kfF40MiS7zA?si=ZC9Uf-3G16PTK-JH (of course you would enjoy this 👀)
- https://home.iitk.ac.in/~pranab/ESO208/rajesh/03-04/Errors.pdf (Types of Errors 💻)
- https://zingale.github.io/comp_astro_tutorial/basics/floating-point/numerical_error.html (Types of Errors 💻)
- https://web.engr.oregonstate.edu/~webbky/ESC440_files/Section%201%20Roundoff%20and%20Truncation%20Error.pdf
- https://en.wikipedia.org/wiki/Round-off_erro
- https://youtu.be/03Lg60MTSdM?si=edZqtyywFMtlJNyD
- https://kapilcaet.wordpress.com/wp-content/uploads/2015/01/unit-4-round-off-and-truncation-errors.pdf
- https://en.wikipedia.org/wiki/Differentiation_rules
- https://www.khanacademy.org/math/multivariable-calculus/thinking-about-multivariable-function/ways-to-represent-multivariable-functions/a/contour-maps