

COMP270: Mathematics for 3D Worlds and Simulations

WEEK 2: GEOMETRY I
PART 3: FUNCTIONS AND DISCRETISATION

Objectives

- **Recall** the mathematical definition of a function
- **Understand** what it means to **discretise** a continuous function, e.g. for implementation in code

Recap: sets

- Definition: a set is a collection of objects (called **elements**) in which order has no significance, e.g.
$$A = \{1, 7, 3, 24, 999\}$$
 - The elements often have some **shared characteristics**, e.g.
 $B = \{r \in \mathbb{R} : r > 3\}$ is the set of all **real numbers** greater than 3
- The Cartesian product of n sets is the set of all n -tuples with one component from each set.

Functions

- In mathematics, a function is a mapping from one set to another
- If S and T are sets, then a function
$$f : S \rightarrow T$$
maps each element of S to an element of T
$$s \in S, \quad f(s) \in T$$
- S is called the domain of f , and T is the codomain
- Note: f maps each element of S to **one and only one** element of T ; however it could map multiple elements of S to the same element of T (a **many-to-one** mapping).

Functions vs functions

- In mathematics:

$$f : S \rightarrow T$$

- In code:

```
class S {...};  
class T {...};  
T f(S s) {...}
```

- (Under the assumption that `f` is implemented to always give the same return value given the same argument – e.g. no internal or external state)

Functions vs functions

- In mathematics:

$$f : \mathbb{R} \rightarrow \mathbb{Z}$$

- In code:

```
int f(float x) {...}
```

- OK, so \mathbb{Z} and `int` aren't really the same, nor are \mathbb{R} and `float`, but close enough for computing...

Multiple arguments

- The domain of a function could be a Cartesian product:

$$f : A \times B \rightarrow C$$

- In code:

```
C f(A a, B b) {...}
```

Continuous vs. discrete

- Much of traditional mathematics is **continuous**: functions can vary smoothly across the entire domain
- Computer science uses **discrete mathematics**, for objects that can only assume **distinct, separate values**
 - Computers can't represent every value exactly, e.g. floats vs. \mathbb{R}
 - Even if they could, we couldn't evaluate a function for e.g. every value of \mathbb{R} ...
- Need to **discretise** mathematical functions by evaluating them at representative intervals

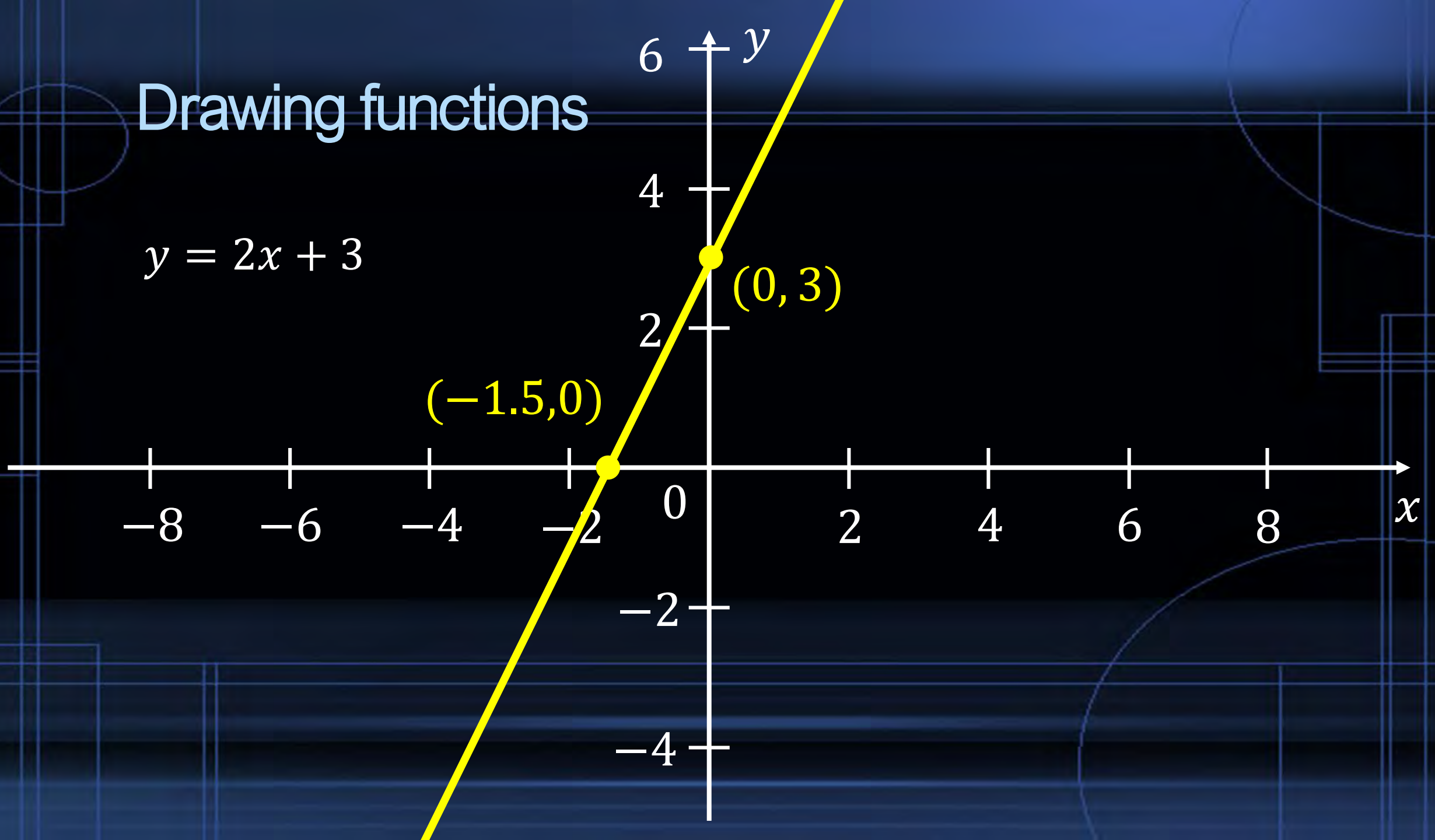
Similar to **sampling** in signal processing

Drawing functions

- For a function $f : \mathbb{R} \rightarrow \mathbb{R}$ we can plot the graph
 $y = f(x)$
- Formed of the points $(x, f(x))$ for x in some range
- Note: only one point per x value, since f maps each x to one and **only one** y value
- Build a picture of a continuous shape by considering the value of $f(x)$ for values of x at various intervals...

Drawing functions

$$y = 2x + 3$$



Drawing functions

$(-6, 5)$

$$y = \frac{1}{4}x^2 - 4$$

y

$(6, 5)$

$(-4, 0)$

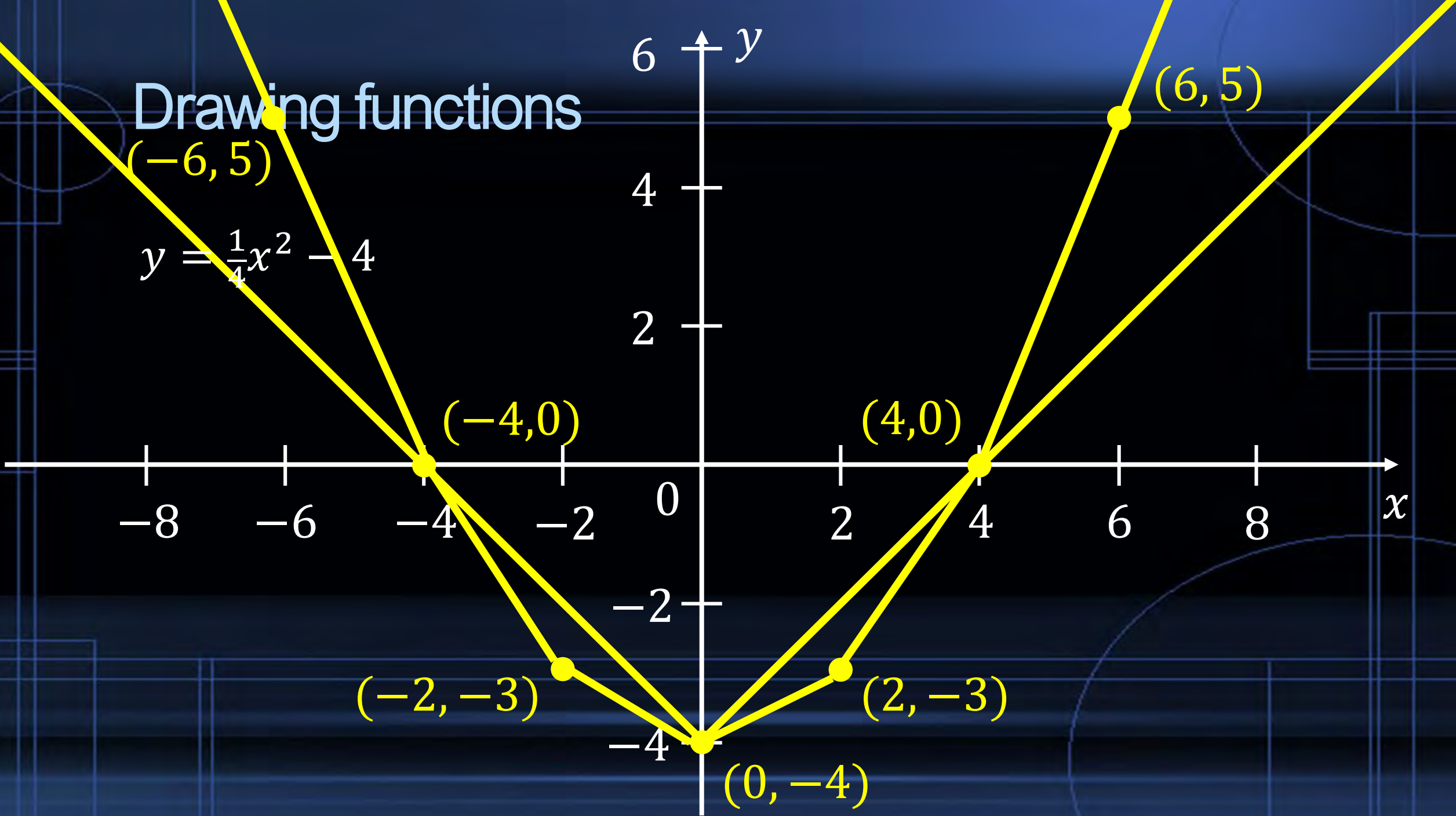
$(4, 0)$

x

$(-2, -3)$

$(2, -3)$

$(0, -4)$



Butterflies and beyond

