

# Differentiation

This lesson discusses first and higher order derivatives.

## WE'LL COVER THE FOLLOWING ^

- First-order derivatives
  - Value of the derivative
- Higher-order derivatives
- Partial derivatives
  - First-order
  - Higher-order

Differentiation is the process of finding a function that outputs the rate of change of one variable with respect to another.

SymPy provides the functionality of symbolically calculating the derivatives of a function.

## First-order derivatives #

Derivatives are computed with the `diff` function, which recursively uses the various differentiation rules.

$$y = x^4 + x^3 + x^2$$

$$\frac{dy}{dx} = 4x^3 + 3x^2 + 2x$$

We will be exploring different types of derivatives using SymPy. The SymPy `diff()` function takes a minimum of two arguments: the function to be differentiated and the variable with respect to which the differentiation is performed.

```
diff(y, x)
```

Let's look at an implementation of this:

```
from sympy import *

def f(x):
    return (x**4 + x**3 + x**2)

def g(x):
    return (sin(x) + cos(x))

def h(x):
    return (sin(x)*exp(x) + x)

x = Symbol('x')

print("Derivative of f(x):", diff(f(x), x))
print("Derivative of g(x):", diff(g(x), x))
print("Derivative of h(x):", diff(h(x), x))
```

## Value of the derivative #

The value of the first derivative determines whether the function is increasing or decreasing. A positive first derivative shows the function is increasing and a negative first derivative shows that the function is decreasing.

To find the value of the derivative, we use the `subs()` method. The first argument of `subs()` is the variable to be substituted and the second argument is its value. The following syntax is used:

```
diff(f(x), x).subs(x, value)
```

## Higher-order derivatives #

Higher-order differentials are the differentials of the first differential. Each of these higher-order differentials has importance. A common example is in kinematics.

- Displacement of a body is defined by  $x(t)$ .
- Its velocity is defined by its first derivative,  $\frac{dx}{dt}$ .

- Its acceleration is defined by its second derivative,  $\frac{d^2x}{dt^2}$ .

As mentioned above, the SymPy `diff()` function takes a **minimum** of two arguments. The remaining arguments can be used for higher-order derivatives.

```
diff(f(x), x, x)
```

Each subsequent argument after the second argument will differentiate the function with respect to that variable.

If the third variable is an integer, it will denote the order of differentiation. Let's look an example of this below:

```
from sympy import *

def f(x):
    return (x**4 + x**3 + x**2)

def g(x):
    return sin(5*x)

x = Symbol('x')

print("Second derivative of f(x):", diff(f(x), x, x))
print("Second derivative of f(x):", diff(f(x), x, 2))    # specifying the value of order of d
print("-----")
print("Third Derivative of g(x):", diff(g(x), x, x, x))
print("Third Derivative of g(x):", diff(g(x), x, 3))    # specifying the value of order of diffe
```



## Partial derivatives #

### First-order #

In mathematics, a partial derivative of a function with several variables is its derivative with respect to one of those variables, with the others held constant. Below are the examples for first-order partial derivatives:

$$z = y^2x + x^2y$$

$$\frac{\partial z}{\partial y} = 2yx + x^2$$

$$\frac{\partial z}{\partial x} = y^2 + 2xy$$

Partial derivatives are used in vector calculus and differential geometry. In SymPy, these are calculated for multivariable functions:

```
from sympy import *

def f(x, y):
    return (y**2)*x + (x**2)*y

x = Symbol('x')
y = Symbol('y')
z = f(x, y)

print("dz/dy = ", diff(z, y))
print("dz/dx = ", diff(z, x))
```



## Higher-order #

For higher-order partial derivatives, see the example below:

$$z = y^2x + x^2y$$

$$\frac{\partial z}{\partial y} = 2yx + x^2$$

$$\frac{\partial^2 z}{\partial y \partial x} = 2y + 2x$$

Higher-order partial derivatives are computed using a similar method as the higher-order derivatives except that in partial derivatives, we have to specify the degree of two variables. We can either specify the variable and the order of its derivative to be computed or keep on adding subsequent arguments.

```
from sympy import *

def f(x, y):
    return (y**4)*(x**3) + (x**4)*(y**3)

x = Symbol('x')
y = Symbol('y')
z = f(x, y)

print("d^3z/dy^2dx = ", diff(z, y, 2, x, 1)) # or could use diff(z, y, y, x)
```





---

In the next lesson, we will learn about integration in SymPy.