Limits

In this lesson, we will learn about computing limits in SymPy.

WE'LL COVER THE FOLLOWING

SymPy calculates symbolic limits with the `limit()` function using the Gruntz algorithm.

The input arguments of the `limit()` function are the mathematical function itself, the variable for which the limit is to be computed, and the point of evaluation:

```
limit(f(x), x, x0)
```

Computing limits

Let's compute a famous sinc function example and see if the `limit()` function returns the correct value.

$$\lim_{x \to 0} \frac{sin(x)}{x} = 1$$

```
from sympy import *

def f(x):
    return sin(x) / x

x = Symbol('x')
print(limit(f(x), x, 0))
```

As seen in line 7, the `limit` function correctly computes the value for the expression. This was simple, wasn't it? Let's compute the limit for a more complex example now:

$$\lim_{x \to 0} \left(2e^{\frac{1-cos(x)}{sinx}} - 1\right)^{\frac{sinh(x)}{atan^2 x}}$$

```
from sympy import *

def f(x):
    return ((2 * exp((1 - cos(x)) / sin(x)) - 1)**(sinh(x) / atan(x)**2))

x = Symbol('x')
print(limit(f(x), x, 0))
```

As you can see, the answer is e.

The direction of a limit

The direction of limit can be specified using the optional `dir` argument.

```
limit(f(x), x, x0, dir)
```

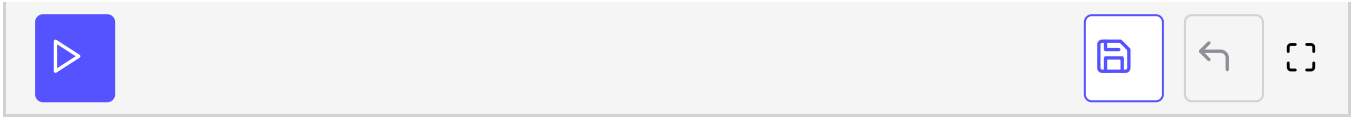
- The limit is bi-directional if the value of `dir` is `'+-'`.
- The limit is from the right if the value of `dir` is `'+'`.
- The limit is from the left if the value of `dir` is `'-'`.

> 💡 The default is from the right.

Let's look at an implementation of all three of them:

```
from sympy import *

def f(x):
    return 1 / x

def g(x):
    return 1 / x**2

x = Symbol('x')
```

```
print("Bi-directional for g(x):", limit(g(x), x, 0, '+-'))


print("From right for f(x)", limit(f(x), x, 0, '+'))
print("From left for f(x)", limit(f(x), x, 0, '-'))
```



Let’s test your understanding of the learned concepts in differentiation, integration, and limits with a short quiz.