

Using LaTeX in Jupyter Notebooks

Contents

Overview	1
Inline Mathematics	1
Common Mathematical Functions	2
Square Roots	2
Integrals and Sums	3
Accents	3
Displaying Equations	3
Sizing Parentheses or Brackets	4
Invisible Delimiters	5
Cases and Matrices	5
Piecewise Functions	5
Matrices	5
Text Inside Math	6
Common Gotchas	6
Curly braces are grouping characters	6
Spaces in math mode	6
Using LaTeX in Figures	6

This tutorial was inspired and draws from Justin Bois' tutorial.

Overview

LaTeX is a typesetting system for creating professional-quality mathematical notation. You don't need to install anything for this class — LaTeX math works inside Quarto, Jupyter notebooks, and most Markdown editors by enclosing expressions in `$` signs. However, if you want to use LaTeX outside of these environments, you will need to install a LaTeX distribution. For example, MacTeX for MacOS, TeX Live for Linux, or MiKTeX for Windows. You can also use online editors like Overleaf.

In this tutorial, you will learn how to typeset mathematics and equations using LaTeX.

Inline Mathematics

To include mathematical notation within text, enclose the LaTeX within dollar signs `$`. For example, to obtain the output

the objective function is $4x + 7x$,

you would enter

```
the objective function is $4x + 7x$.
```

You can enter subscripts and superscripts with `_` and `^`, respectively; to get

the function is $f(x_i) = x_i^2$,

type

the function is $f(x_i) = x_i^2$.

If you want multiple characters to be enclosed in a subscript or superscript, enclose them in braces {}:

$e^{i\pi} - 1 = 0$ is produced by

`$e^{i \pi} - 1 = 0$` .

To get special characters like π (or other Greek letters), precede their name (or sometimes a code) with a backslash: `$\pi$` . There are a number of special characters like this, which you can find in cheatsheets like this one.

Bold characters, which you might use to denote vectors, can be rendered using `$\mathbf{a}$` :

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i \times b_i$$

`$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i \times b_i$`

Fractions can be displayed using `$\frac{}{}$` , where the first bracket encloses the numerator and the second the denominator, as in

$$\frac{1}{2}$$

`$\frac{1}{2}$`

Common Mathematical Functions

Names of standard functions should be preceded by a backslash so they render in upright (roman) type, not italic. This is the #1 beginner mistake:

Incorrect	Correct	Renders as
<code>$\sin(x)$</code>	<code>$\sin(x)$</code>	$\sin(x)$
<code>$\cos(x)$</code>	<code>$\cos(x)$</code>	$\cos(x)$
<code>$\log(x)$</code>	<code>$\log(x)$</code>	$\log(x)$
<code>$\ln(x)$</code>	<code>$\ln(x)$</code>	$\ln(x)$
<code>$\exp(x)$</code>	<code>$\exp(x)$</code>	$\exp(x)$
<code>$\max(x,y)$</code>	<code>$\max(x,y)$</code>	$\max(x,y)$
<code>$\min(x,y)$</code>	<code>$\min(x,y)$</code>	$\min(x,y)$

Square Roots

Use `$\sqrt{}$` :

$$\sqrt{x} \text{ and } \sqrt[3]{x}$$

`\sqrt{x}` and `\sqrt[3]{x}`

The optional `[3]` specifies the root degree.

Integrals and Sums

Integrals use `\int` with optional bounds via `_` and `^`:

$$\int_0^\infty e^{-x} dx$$

`\int_0^\infty e^{-x} \, dx`

The `\,` adds a thin space before `dx`, which is conventional.

Sums use `\sum`:

$$\sum_{i=1}^n x_i$$

`\sum_{i=1}^n x_i`

Accents

Common for denoting vectors, estimates, or time derivatives:

Code	Renders as	Use
<code>\hat{x}</code>	$\hat{x}$	Unit vector or estimator
<code>\bar{x}</code>	$\bar{x}$	Sample mean
<code>\tilde{x}</code>	$\tilde{x}$	Tilde variable
<code>\dot{x}</code>	$\dot{x}$	Time derivative
<code>\ddot{x}</code>	$\ddot{x}$	Second time derivative
<code>\vec{x}</code>	$\vec{x}$	Vector (alternative to <code>\mathbf{}</code>)

Displaying Equations

To place equations or other mathematics on their own line(s), enclose the entire block in *two* dollar signs `$$`. For example, the prior dot-product definition could be displayed as

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i \times b_i$$

using

```
$$  
\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i \times b_i  
$$
```

Displaying equations on their own line(s) can improve the spacing of symbols like sums (as above) or fractions: compare the inline

$$x < \frac{1}{2}$$

to

$$x < \frac{1}{2}.$$

To display multiple related lines in a single block, there are two environments of note. The first will center all of the equations, and is obtained by enclosing the equations in `\begin{gather}` and `\end{gather}`. Each line should be separated with `\\`:

$$\begin{aligned} x_1 + x_2 &\leq 5 \\ y &\leq \frac{1}{2}. \end{aligned}$$

```
$$
\begin{gather}
x_1 + x_2 \leq 5 \\
y \leq \frac{1}{2}.
\end{gather}
$$
```

The second environment will let you align the equations as you wish instead of automatically centering them, and is used by enclosing the equations with `\begin{align}` and `\end{align}`, with an ampersand `&` in front of the characters which will be used on each line to align the equations:

$$\begin{aligned} x_1 + x_2 &\leq 5 \\ y &\leq \frac{1}{2}. \end{aligned}$$

```
$$
\begin{align}
x_1 + x_2 &\leq 5 \\
y &\leq \frac{1}{2}.
\end{align}
$$
```

Sizing Parentheses or Brackets

By default, parentheses and brackets are sized for simple characters, but will look bad when used to surround fractions or sums, particularly when they are not used in-line:

$$x_n = (\frac{1}{2})^n.$$

To make this look better, use `\left` and `\right` around the left and right parentheses or brackets:

$$x_n = \left(\frac{1}{2}\right)^n$$

```
$$
x_n = \left(\frac{1}{2}\right)^n.
$$
```

This is totally optional, but helps!

Invisible Delimiters

When splitting an expression across multiple lines, you sometimes need a bracket on one line without its partner. Use `\left.` and `\right.` (with a period) for invisible parentheses:

$$x = \left(\frac{1}{2} + \frac{1}{3} \right. \\ \left. + \frac{1}{4} \right)$$

```
$$
\begin{align}
x = \left( \frac{1}{2} \right. & \left. + \frac{1}{3} \right. \\
& \left. + \frac{1}{4} \right) \\
\end{align}
$$
```

Cases and Matrices

Piecewise Functions

Use `\begin{cases}` for piecewise definitions:

$$f(x) = \begin{cases} x^2 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

```
$$
f(x) = \begin{cases}
x^2 & \text{if } x \geq 0 \\
0 & \text{otherwise}
\end{cases}
\end{cases}
$$
```

The `&` separates the expression from the condition, and `\text{}` lets you write normal text inside math mode.

Matrices

Use `\begin{bmatrix}` for bracketed matrices or `\begin{pmatrix}` for parentheses:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

```
$$
A = \begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23}
\end{bmatrix}
\end{bmatrix}
$$
```

Columns are separated by `&`, rows by `\\`.

Text Inside Math

When you need words inside an equation, use `\text{}`:

$$\underbrace{x_1 + x_2 + \cdots + x_n}_{n \text{ terms}}$$

```
$$
\underbrace{x_1 + x_2 + \cdots + x_n}_{\text{$n$ terms}}
$$
```

Without `\text`, “n terms” would render as *nterms* (italic, no space).

Common Gotchas

Here are mistakes that trip up nearly every beginner.

Curly braces are grouping characters

To display an actual `{` or `}`, escape them by preceding them with a backslash:

$$\{x \mid x > 0\}$$

```
$\{x \mid x > 0\}$
```

Spaces in math mode

LaTeX ignores standard spaces in math mode. To add space, use:

Command	Width
<code>\,</code>	Thin space
<code>\:</code>	Medium space
<code>\;</code>	Thick space
<code>\quad</code>	One em (roughly the width of “M”)
<code>\qquad</code>	Two ems

Using LaTeX in Figures

You may want to use LaTeX in figures, for example if your x -axis should have a title like x . To do this, load the `LaTeXStrings` package and precede the relevant LaTeX-formatted string (within in-line dollar signs `$`) with `L`, as in:

```
using Plots
using LaTeXStrings

x = -2π:0.01:2π
plot(x, exp.(sin.(x)), xlabel=L"$x$", ylabel=L"$e^{\sin(x)}$", legend=false)
```

