LaTeX Made Simple for Science Students

Personal Project

Simon Schmit — Physics Student

May 2025

Contents

1 Introduction

1.1 What is LaTeX?

If you're studying science, chances are you'll need to write reports that look professional and are easy to read, and that's exactly where LaTeX shines. LaTeX is the most common tool used by researchers and scientists around the world to produce high-quality scientific documents, especially when dealing with complex formulas, figures, and references.

This guide, "LaTeX Made Simple for Science Students", is designed specifically for students who are new to LaTeX and want to get familiar with this tool. Its purpose is to simplify learning by providing practical examples and straightforward explanations. No specific programming knowledge is required.

Throughout this tutorial, you will learn how to:

- Structure your documents with chapters, titles, and tables of contents
- Format text, create lists, and write important information
- Write mathematical formulas and equations clearly and correctly
- Include figures, tables, and graphics in your documents
- Manage references and citations professionally

By the end of this guide, you will be able to create clean, well-organized scientific documents that meet the standards expected in academic and research environments. Whether you are writing lab reports, project summaries, or research papers, this tutorial will give you the tools to present your work confidently and clearly.

1.2 Why use LaTeX for scientific documents?

As a science student, you're often asked to write lab reports, project summaries, or research documents. LaTeX is especially useful for that because it produces clean, structured, and professional results, and even more so when it comes to scientific content.

Writing equations in LaTeX is much easier and more complete than in regular word processors. You can include mathematical symbols, fractions, matrices, vectors, integrals, and more, with precision and consistency. It's also ideal for creating structured tables, systems of equations, or complex expressions, all with a clear and elegant layout. Once you understand the basics, LaTeX saves you time by automatically managing the layout, numbering, and references. That's why it is the standard tool used by researchers, teachers, and journals worldwide.

2 The basics

2.1 Using LaTeX: how?

LaTeX is not a traditional text editor like Word or Google Docs. Instead, it is a typesetting system where you write your document using plain text files with special commands. This approach gives you full control of the appearance of your document, especially for scientific writing.

To use LaTeX, you need two things:

1. A LaTeX editor where you write your code
2. A LaTeX compiler that processes the code and generates a PDF

But note that today, many code editors enable you to compile your document as well. You can even have a look at your compiled file while writing the code.

There are many tools available to write and compile LaTeX documents:

- **Overleaf** : Probably the most famous online platform that lets you write, compile, and share LaTeX documents directly in your web browser. It requires no installation and includes many templates, all for free.
- **TeXstudio, TeXworks, Texmaker**: desktop applications that can be installed on your computer, offering a user-friendly environment for writing and compiling LaTeX locally. They all have their advantages, but they are generally known for offering more advanced features
- **MiKTeX, MacTeX, TeX Live**: These are LaTeX distributions that include all the tools and packages necessary to compile LaTeX documents. Usually installed alongside desktop editors.

About this document :
This guide has been written and compiled using Overleaf, which makes it easy to focus on learning LaTeX without worrying about installation or setup. The intuitive interface of Overleaf and cloud storage also makes it ideal for students who want to work on their documents anywhere, on any device.

You should also be aware that this document contains a lot of information, but does not list all the existing commands and mathematical notations. So, if you're completely new to LaTeX, I recommend using other resources that list all the essential commands. That's why you'll find some underlined text throughout the document; these are clickable links that lead to relevant resources related to the corresponding section.

2.2 Writing your first document

Before starting writing, you should know that every LaTeX document begins with a document class that defines the basic style of the document. The most common classes are article, book, and report.

So first, you'll need to write a command like this:

\documentclass{article}

Once you've decided the type of document you want to create, you must write your content inside a special environment called document. Everything between those two lines is what will appear in the final PDF. To do this, you simply need to write:

\begin{document} to start
\end{document} to end

In the next chapter you will learn about the structure of a document, to complete the basic knowledge you need to write simple articles by yourself.

3 Structure of a document

Understanding the structure commands is essential for organizing your content clearly and efficiently.

3.1 Class of documents

In LaTeX, a **document class** defines the general structure of the document. Choosing the right class is important to tailor the document for its purpose.

The document class is declared in the very first line of the source file using:

\documentclass[options]{...}

Here are some of the most commonly used document classes:

- **article**: for short documents, papers, and notes.
- **report**: for longer documents.
- **book**: for books or more structured documents.
- **memoir**: a flexible class for custom reports and books.
- **beamer**: used to create presentation slides.

In scientific writing, the *article* class is the most commonly used because it suits papers, lab reports, and publications. Its simple structure makes it ideal for clear and formal documents.

As you can see in the command above, you can also add options to the document class. These options allow to make your creation more personal and customized so that they fit your expectations. Some of the most useful class options are:

- `12pt`, `11pt`, `10pt` — sets the base font size
- `twocolumn` — formats the document in two columns
- `twoside` — enables mirrored margins for printing on both sides
- `titlepage` — forces the title to appear on its page
- `a4paper` — sets the paper size to A4 (common in Europe)

Yet, if you need other options, you might have to take a look at the multiple resources available online. Here's a good one.

3.2 Organizing Content

When creating documents, organizing them into a clear hierarchy is easy and essential for readability. It's important to structure content logically. Fortunately, LaTeX is one of the best tools for this purpose.

Title of the document

The first information you want to show in any type of document is the title, and for that, you only have to define it before the document environment, at the top of the code, by writing:

\title{*your title*}

Additionally, you can also define the author's name and the date of creation by adding, at the same place, these commands:

\author{*your name*}
\date{*date*}

To display it on your creation, you have different options, you can use this command:

\maketitle

It must be the first thing you write in the body of the document if you want it to be on the first page, which is almost always what you want to do.
Now, if you want this information to be in the middle of the first page, as in this document, you can use:

```
\begin{titlepage}
    \centering
    \vspace*{\fill}

    {\Huge \textbf{your title} \par}
    \vspace{2em}
    {\Large your name \par}
    \vspace{1em}
    {\large date \par}

    \vspace*{\fill}
\end{titlepage}
```

You don't need to understand all of this for the moment, you can just copy and paste. However, if you want to understand, here is an explanation :

- \centering centers everything horizontally that follows within the current environment.

- \vspace*{\fill} adds vertical flexible space.
- \par indicates the end of a paragraph, forcing a line break
- We will see the other commands in the next chapters.

Chapters and sections

When structuring a document, it's important to choose the right level of division depending on the type of document you're writing. In LaTeX, the two main top-level divisions are chapters and sections, but they are not interchangeable.

Chapters

To define a new chapter, you only have to use the command:

\chapter{...}

Once you have defined it, you can write anything you want to be in this chapter, right below the command. Then, when you're done with a chapter, to begin a new one, you just need to use the command again. A chapter is the highest-level division in LaTeX, but it's only available if you're using certain document classes like:

- book
- report

Each one starts on a new page and is automatically numbered. Chapters are ideal for writing books, dissertations, longer reports, etc.

Sections

Sections are the main divisions used in shorter documents and are available in all document classes, including article. They do not start a new page, and are automatically numbered as well.
To define a new section, you only have to use the command:

\section{...}

Then write whatever you want to belong to this section.

What I use in this document

In this document, I'm using the article class. This means that the chapter command won't work.

Here's a quick visual of the structure I'm using:

```
\section{Main Topic}
\subsection{Subtopic}
\subsubsection{Detail}
```

Subsections

To further divide your sections, you can use \subsection{...} the same way you used \chapter{...} and \section{...}. These commands let you organize your content more precisely.

Subsubsections

A subsection goes inside a section and creates a second level of structure, which is also numbered. If needed, you can divide a subsection even further using \subsubsection{...}. It's a more detailed level of structure. In this document, we're using it mainly for short technical explanations, like this paragraph.

Table of contents

To help people who read your documents find the information they need, you may want to show the *Table of contents*. And nothing is easier than this, because you just have to use:

```
\newpage
\tableofcontents
\newpage
```

It will automatically show a full page with all your chapters, sections, subsections, and subsubsections in the right order.

4 The packages

4.1 What's a package and how to use one?

Packages are extensions that add new features and possibilities to classic LaTeX options. They let you:

- Enhance document formatting
- Add special symbols and math functions
- Include graphics and tables
- Customize almost anything in your document

How to Use a Package
To use any package:

1. Declare it in the preamble (before \begin{document})
2. Use the command:

\usepackage{package-name}

Example of how to use a package:
To enable advanced math features:

```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
...
\end{document}
```

Most LaTeX distributions include common packages. If missing, your editor will show an error during compilation.

4.2 Necessary packages for scientific documents

These packages are essential for most scientific documents:

- **amsmath** - Advanced math typesetting
- **graphicx** - Image inclusion
- **siunitx** - Proper unit formatting
- **hyperref** - Interactive document (makes references and table of contents clickable)

- **`biblatex`** - Bibliography management

In the following chapters, you will learn to:

- Use **`amsmath`** for including equations, matrices, and advanced notation.
- Master **`graphicx`** to integrate and customize figures, with precise control over sizing, positioning, and formatting.

Languages

To change the language of your document (which affects hyphenation, date formatting, and some automatic texts like "Contents" or "Chapter"), you should use the `babel` package.

For example, to write in French, include the following line in your preamble:

- \usepackage[french]{babel}

If you are writing in English, it's automatic; you don't need this package.

5 Body of the document

Now that the document framework is ready, we'll focus on content creation and formatting techniques.

5.1 Italics, bolds, underline, etc.

Basic Text Styles

Here are the most useful commands for quick formatting:

- `\textbf`{*bold text*} → **bold text**
- `\textit`{*italic text*} → *italic text*
- `\underline`{*underlined*} → underlined
- `\texttt`{*monospace*} → `monospace`
- `\textsc`{*Small Caps*} → SMALL CAPS

Advanced Tips

For scientific writing:

- Math mode has separate commands:
  - `$\mathbf{bold}$` → $\mathbf{bold}$
  - `$\mathit{italic}$` → italic

5.2 Lists

Two Main List Types

There are two main types of lists that you may want to use, depending on the purpose:

- `itemize`: For unordered lists
- `enumerate`: For numbered lists

How to add a list to your document?

When creating lists, you'll typically use either:

```
\begin{itemize}
    \item First item
    \item Second item
    \item Third item
\end{itemize}
```

The code above produces the following output:

- First item
- Second item
- Third item

Ordered lists are made the same, just by replacing **itemize** by **enumerate**:

```
\begin{enumerate}
    \item First point
    \item Second point
    \item Third point
\end{enumerate}
```

5.3 Other tips

After mastering basic formatting, these commands will help you refine your document's layout with precision:

Spacing Commands

- `\vspace{length}`: Vertical spacing
  - `\vspace{1em}` after headings
  - `\vspace*{\fill}` to fill page space
- `\hspace{length}`: Horizontal spacing
  - Align equations: `$E=mc2\hspace{2em}$`
  - Create indents: `\hspace{3em}Text`

Paragraph Formatting

- `\noindent`: Suppresses paragraph indentation: no space between the margin and the first word of the paragraph.
- `\par`: Forces paragraph break and is also an alternative to blank lines.

6 Writing Mathematics using LaTeX

One of LaTeX's greatest strengths is its ability to easily and beautifully render equations, matrices, and other mathematical objects. This section may seem a little long, but it represents the most important part of the specific knowledge required to write scientific documents that contain equations properly.

6.1 Essentials

Including an Equation

LaTeX allows you to include mathematical expressions either inline or as displayed equations.

To insert a formula within a sentence, enclose it between dollar signs. For example, typing `$E=mc^2$` produces the inline result $E = mc^2$.

For a standalone centered equation, you can use either:

`\[E=mc^2\]` or `$$E=mc^2$$`

Both render as:

$$E = mc^2$$

Exponents and Subscripts

Use the caret `^` for superscripts and the underscore `_` for subscripts.

- `x^{2}` $\rightarrow x^2$
- `e^{-kt}` $\rightarrow e^{-kt}$
- `v_{0}` $\rightarrow v_0$
- `C_6H_12O_6` $\rightarrow C_6H_{12}O_6$

Mathematical Operators

Here are a few useful operators:

Command	Output
`\frac{a}{b}`	$\frac{a}{b}$
`\sqrt{x}`	$\sqrt{x}$
`\sqrt[n]{x}`	$\sqrt[n]{x}$

Calculus and Linear Algebra

To write integrals, use `\int` and add limits with subscripts/superscripts:

- `\int_{a}^{b}` $\rightarrow \int_a^b$
- `\oint` $\rightarrow \oint$

- `\iint_{a}^{b}` → $\iint_a^b$

For sums and products:

- `\sum_{n=1}^{\infty}` → $\sum_{n=1}^{\infty}$
- `\prod` → $\prod$

Important Considerations

Always use math mode, even for basic symbols:

- Wrong: `a > b` → a > b
- Correct: `$a \gt b$` → $a > b$

If a variable has multiple letters, use `\mathrm` to avoid awkward spacing:

- `$\mathrm{Re}$` → Re

Additionally

To help you go further, I'm providing this external reference: the *Wikibooks* page about mathematics in LaTeX. It gathers many mathematical commands and symbols, with very useful examples when you are looking for a specific notation. To find what you need, look at the content on the left of the page, and you will probably find anything you want.

6.2 Vectors, Matrices, and Tables

Vectors

In LaTeX, there are two common ways to write vectors, depending on the notation style you prefer or that is required by the content of what you're doing. The most common visual representation is a letter with an arrow above it, written using the `\vec` command. For example, to write vector $\vec{v}$, use:

`$\vec{v}$` → $\vec{v}$

If you want to write the vector as a list of components, like a column vector, you will need to use the `bmatrix` or `pmatrix` environments. Here's how you could define a simple column vector:

```
\[
\vec{v} =
\begin{bmatrix}
1 \\
2 \\
3
\end{bmatrix}
\]
```

This will give you:

$$\vec{v} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

As you can see, you must use `\\` to write a new component; this "sub-command" goes below. This shows the vector both symbolically and by its coordinates. You can use `bmatrix` instead of `pmatrix` if you want square brackets instead of parentheses.

Matrices

Matrices in LaTeX are written using environments such as `matrix`, `bmatrix`, `pmatrix`, and others provided by the `amsmath` package. The most common ones are:

- `matrix` for a plain layout
- `bmatrix` for square brackets
- `pmatrix` for parentheses

For example, to define a 2×2 matrix, you can use the following syntax:

```
\[
A =
\begin{bmatrix}
a & b \\
c & d
\end{bmatrix}
\]
```

This will give you:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

In this example, the symbol `&` separates columns, and the double backslash `\\` moves to the next row as for vectors. You can extend this to any matrix size, even non-square ones. As you can see, in LaTeX, matrices are just vectors with additional columns.

Tables

While matrices and vectors are typeset in math mode, general tables are created using the `tabular` environment. This environment is very flexible and allows you to define columns (left, center, or right aligned), insert lines, and even merge cells with extra packages.

Here's a basic table with three columns:

```
\begin{tabular}{|c|c|c|}
\hline
x & y & z \\
\hline
1 & 2 & 3 \\
4 & 5 & 6 \\
\hline
\end{tabular}
```

This will give you:

x	y	z
1	2	3
4	5	6

Once again, the way to write content in a new column or move to the next row is the same as for vectors and matrices.
In the argument `{|c|c|c|}`, the vertical bars `|` (AltGr + 6) create column borders, and `c` stands for "centered". You can also use `l` (left) or `r` (right) for alignment.

To manage horizontal lines in your table:

- `\hline` adds a horizontal line across the full width of the table.
- `\cline{i-j}` draws a line from column `i` to column `j`, useful for partial lines.
- `\midrule`, `\toprule`, and `\bottomrule` (from the `booktabs` package) give cleaner spacing and are better for professional-looking tables.

For more advanced tables with merged cells or custom formatting, you might want to explore the `booktabs`, `multirow`, or `tabularx` packages. Here's a more detailed page about tables : Click Here

6.3 Mathematical symbols

Here is a small selection of frequently used math symbols. For a full list, refer to the Wikibooks page mentioned above.

Command	**Output**
`\alpha`, `\beta`, `\gamma`, etc.	α, β, γ
`\pi`, `\Delta`, `\Omega`, etc.	π, Δ, Ω
`\leq`, `\geq`, `\neq`	$\leq$, $\geq$, $\neq$
`\infty`, `\approx`, `\sim`	∞, $\approx$, $\sim$
`\rightarrow`, `\Leftarrow`, `\mapsto`	$\rightarrow$, $\Leftarrow$, $\mapsto$
`\cdot`, `\times`	$\cdot$, $\times$
`\div`, `\pm`	$\div$, $\pm$
`\wedge`, `\vee`	$\wedge$, $\vee$
`\nabla`, `\partial`	∇, ∂
`\forall`, `\exists`	$\forall$, $\exists$

7 Including plots and graphics

Scientific and technical documents often require the inclusion of plots, diagrams, or images to support the text and illustrate key results. LaTeX provides powerful tools to import, scale, and position graphics with precision. In this section, we will cover the essential commands to include external images and discuss how to integrate plots.

7.1 Adding and sizing an external document

In LaTeX, including external graphics such as plots, diagrams, or images is straightforward thanks to the `graphicx` package. This package supports many formats, including `.jpg`, `.png`, and `.pdf`.

To include an image, first ensure the file is in your project folder. If you are using Overleaf, you don't need to explicitly load the image; simply upload it or open it in the file tree, and it will be ready to use.

Here's the basic syntax to include an image:

```
\includegraphics[width=0.5\textwidth]{doc-name.png}
```

This inserts the image at half the text width, but you can personalize it.

For inserting entire pages or multiple pages from external PDF documents (such as multipage figures or reports), the `pdfpages` package is used:

```
\includepdf[pages=1]{doc-name.pdf}
```

This command inserts page 1 of `my_report.pdf` as a standalone page in your document, but you can specify ranges or all pages:

- `\includepdf[pages={2-4}]{...}` includes pages 2 to 4.
- `\includepdf[pages=-]{...}` includes all pages.

Including graphics and external documents this way gives you great flexibility in combining plots, images, and PDFs in your LaTeX documents.

Positioning and sizing images

To control the placement and size of your images in LaTeX, you can use various options and environments.

Sizing

You can specify width, height, or scale in `\includegraphics` options:

- `\includegraphics[width=0.5]{image.png}` scales the image to 50% of the text width.

- `\includegraphics[height=3cm]{image.png}` sets the height to 3 cm, scaling width proportionally.
- `\includegraphics[scale=0.8]{image.png}` scales the image to 80% of its original size.
- You can combine options like `width` and `height`, but it may distort the image.

Positioning

By default, images are placed inline. To better control placement:

- Use the `figure` environment to treat the image as a floating object that LaTeX will position intelligently.

```
\begin{figure}[htbp]
    \centering
    \includegraphics[width=0.6\textwidth]{image.png}
    \caption{A caption for the image}
    \label{fig:myimage}
\end{figure}
```

- The optional argument `[htbp]` suggests to LaTeX where to try placing the figure in order of preference:
  - `h` = here (approximate position in text)
  - `t` = top of the page
  - `b` = bottom of the page
  - `p` = on a dedicated page for floats
- `\centering` centers the image inside the figure environment.
- To align images left or right, use the `flushleft` or `flushright` environments or `\raggedright` and `\raggedleft`.

Full page images

To insert an image that occupies an entire page, combine `figure` with the `p` option or use the `pdfpages` package for PDFs:

```
\begin{figure}[p]
    \centering
    \includegraphics[width=\textwidth,height=\textheight,keepaspectratio]{bigimage.png}
\end{figure}
```

This tells LaTeX to place the figure on a dedicated page, scaling it to fit the page while keeping the aspect ratio.

7.2 Adding captions

To label and describe your figures, use the \caption command inside a figure environment. Captions automatically add a numbered label and allow you to refer to the figure later in your document.

Here's a basic example:

```
\begin{figure}[h]
\centering
\includegraphics[width=0.6\textwidth]{figures/image.png}
\caption{This is a description of the figure.}
\end{figure}
```

This will place a centered image with the caption "Figure X: This is a description of the figure" below it.

You can also add a label with \label... after the caption to refer to the figure elsewhere using \ref...:

```
\begin{figure}[h]
\centering
\includegraphics[width=0.6\textwidth]{figures/image.png}
\caption{Results of the simulation.}
\label{fig:simulation}
\end{figure}

As shown in Figure~\ref{fig:simulation}, ...
```

NOTE: Always place \label after \caption to ensure the correct figure number is recorded.

7.3 Example

Let's walk through a practical example: we want to insert an image called phase-diagram.png, scale it to 70% of the text width, center it, and add a caption and label for reference.

Code used:

```
\begin{figure}[htbp]
\centering
\includegraphics[width=0.7\textwidth]{phase-diagram.png}
\caption{Phase diagram of the system.}
\label{fig:phase}
\end{figure}
```

Result:

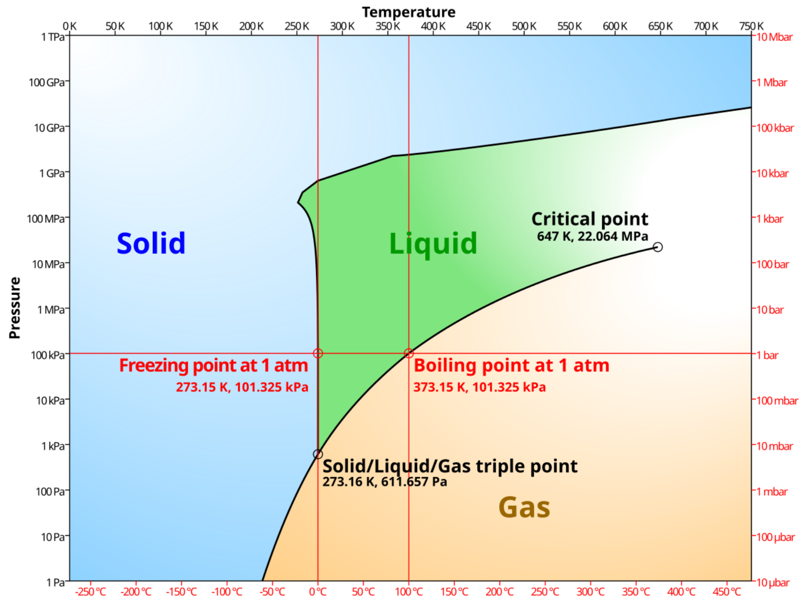


Figure 1: Phase diagram of the system.

7.4 Plotting with pgfplots

For scientific documents, it is often necessary to include precise, customizable plots. The `pgfplots` package is one of the most powerful tools in LaTeX to generate 2D and 3D plots directly from code. It uses the `TikZ` drawing library and allows you to plot functions, data points, and more.

As you will see, plotting on LaTeX is very similar to Python with matplotlib.pyplot collection in the code and the visuals. Like matplotlib.pyplot, pgfplots uses a declarative syntax where plot elements (axes, labels, styles) are defined before rendering. Key differences include native LaTeX text rendering and direct compatibility with your document fonts/math modes.

Installing the package

Before using `pgfplots`, make sure to include it in the preamble of your document:

```
\usepackage{pgfplots}
\pgfplotsset{compat=1.18}
```

The `compat` version ensures compatibility with the version of pgfplots installed on your system. You can adjust it depending on your setup.

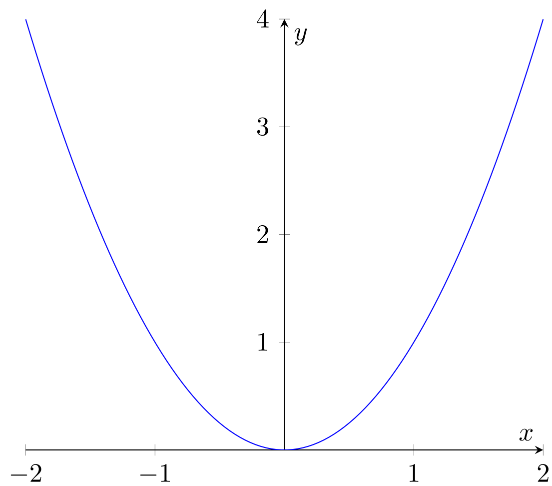
Plotting a function

Here is an example of how to plot a simple function like $y = x^2$:

```
\begin{tikzpicture}
  \begin{axis}[
    axis lines = center,
    xlabel = $x$,
    ylabel = $y$,
    title = {$y = x^2$}
  ]
    \addplot [
      domain=-2:2,
      samples=100,
      color=blue,
    ]
    {x^2};
  \end{axis}
\end{tikzpicture}
```

The \addplot command draws a curve in blue over the x-range [-2:2], using 100 sample points for smooth plotting.

Here's the result:



Plotting from data

You can also plot experimental data from a table using the `addplot table` command. Here's an example using inline data:

```
\begin{tikzpicture}
  \begin{axis}[
    xlabel=Time (s),
    ylabel=Voltage (V),
    title=Measurement Data,
    grid=major,
  ]
    \addplot[
      color=red,
      mark=*,
    ]
    table {
      0 0
      1 1.5
      2 2.7
      3 3.8
      4 4.2
    };
  \end{axis}
\end{tikzpicture}
```

This creates a plot of points with markers, suitable for representing discrete experimental measurements.

Tips

- Line colors, markers, legends, and styles can be customized as desired.
- For complex plots, consider using external data files, like Excel, with: `addplot table [x=X, y=Y] data.txt`.
- Visit the PGFPlots gallery for inspiration.

The `pgfplots` package is a powerful way to bring your scientific results to life directly inside your LaTeX documents, without the need for external plotting tools.

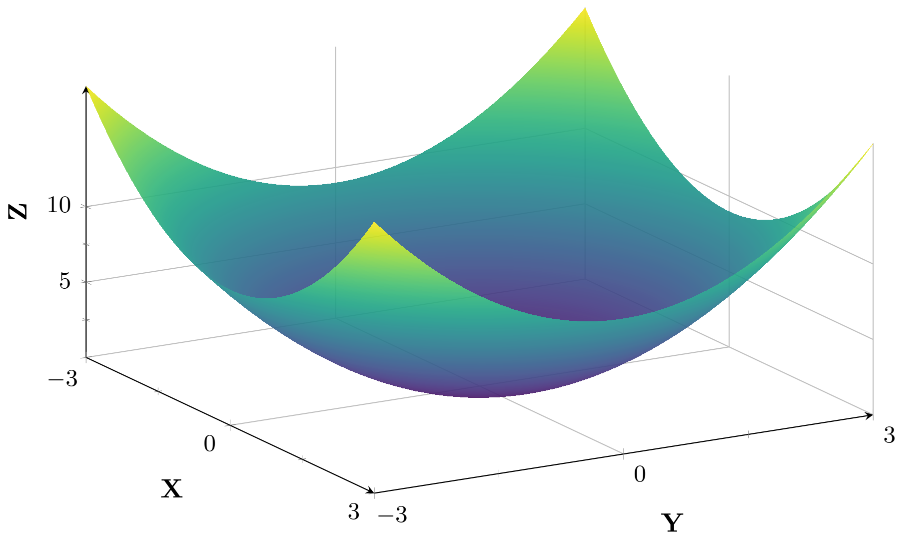
Example of a 3D plot using `pgfplots`

The 3D plot below illustrates a circular paraboloid defined by the function $z = x^2 + y^2$. Here are the key settings used:

- `view={60}{30}`: the angles of view

- `colormap/viridis`: Perceptually uniform colors
- `shader=interp`: Smooth color interpolation

Circular Paraboloid: $z = x^2 + y^2$



8 Final tips

8.1 Bibliography: cite sources

Properly citing sources is essential in any academic or scientific document. LaTeX offers robust tools for handling references in a clean and automated way.

Simple method with `thebibliography`

For shorter documents, you can manage references manually using the `thebibliography` environment:

```
\begin{thebibliography}{9}

\bibitem{bergson}
H. Bergson,
\textit{Creative Evolution},
Félix Alcan, 1907.

\bibitem{curie}
M. Curie,
\textit{Research on Radioactive Substances},
Gauthier-Villars, 1904.

\end{thebibliography}
```

Then, you can cite them in the text using: `\citecurie` → [2]

Using a `.bib` file (BibTeX)

For longer projects, it's recommended to use an external `.bib` file (e.g., `refs.bib`) to store your references. Here's an example:

```
@book{curie,
author = {Marie Curie},
title = {Research on Radioactive Substances},
year = {1904},
publisher = {Gauthier-Villars}
}
```

In your main LaTeX document, include:

```
\bibliographystyle{plain}
\bibliography{refs}
```

And cite it in the text using: `\citemonod` → [2]

Useful styles and packages

With `natbib` or `biblatex`, you can also use:
`\citetbergson` → Bergson (1907)
`\citepmonod` → (Monod, 1971)

Citing your sources correctly is a matter of academic integrity and rigor. LaTeX helps ensure that your bibliography is both accurate and beautifully formatted.

8.2 Commenting your code

Commenting your code is essential for keeping your LaTeX documents organized, especially as they grow in size and complexity. Comments can help you quickly locate sections, clarify the purpose of commands, and make it easier to understand your code when you return to it later.

How to comment?

In LaTeX, simply add a percent sign `%` before the text you want to treat as a comment. Everything after the `%` on that line will be ignored during compilation.

Example:

```
% This section defines the document title
\title{My Report on Phase Diagrams}
```

Using comments like this improves readability and saves time when debugging or updating your LaTeX file.

Conclusion

In this guide, we have explored the essential tools and techniques needed to begin working with LaTeX. From structuring documents and formatting text to including mathematical equations and inserting figures, you now have a solid foundation to create clear and professional scientific reports.

Remember, LaTeX is a powerful tool; the more you use it, the more confident and efficient you will become. Don't hesitate to experiment with different packages, templates, and styles as you grow more comfortable with the language.

Whether you're writing lab reports, thesis chapters, or scientific articles, LaTeX will support your work with precision, consistency, and elegance.

References

- **Overleaf Documentation**. A complete online LaTeX editor with real-time collaboration and version control.
`https://www.overleaf.com/learn`

- **Wikibooks – LaTeX**. An open-content textbook covering most aspects of LaTeX.
`https://en.wikibooks.org/wiki/LaTeX`

- **LearnLaTeX.org**. A beginner-friendly interactive platform to learn LaTeX step by step.
`https://www.learnlatex.org/`

- **YouTube – LaTeX Tutorials by Michelle Krummel**. A clear and concise playlist that covers LaTeX basics and document structuring.
`https://www.youtube.com/playlist?list=PLNFI79D7MYqPlOvLY1TqRKnZWCGKMIIGw`

- **The PGFPlots Package Manual**. Official documentation for creating 2D/3D plots in LaTeX.
`https://mirrors.ctan.org/graphics/pgf/contrib/pgfplots/doc/pgfplots.pdf`