

## 1 Figures

Making figures in L<sup>A</sup>T<sub>E</sub>X is somewhat different from other word processors. Rather than placing the figure ourselves, the best practice is usually to let L<sup>A</sup>T<sub>E</sub>X chose the placement. For that reason, we also tend to refer to figures by their label rather than writing something like “...as seen in the figure below”. The code for including a figure looks something like this:

```
\begin{figure}
\centering
Figure Content
\caption{My figure caption.}
\label{fig:myfigure}
\end{figure}
```

The `\centering` simply centers the content (as is usually desired). This piece of code is typically placed after the paragraph where we first reference the figure. In the above code, the figure content is just text. To include an image, we first need to load the package `graphicx` in the preamble with the command `\usepackage{graphicx}`. We can now replace “Figure Content” with

```
\includegraphics[width=0.6\textwidth]{image.png}
```

The optional `width` parameter decides the width of the image. In this case it is set to 60% of the page width. As seen earlier, we could also use centimetres or millimetres as our unit of choice. In this case the image `image.png` is placed in the same directory as our `.tex` file. Usually, you would prefer to have all of your images in a subfolder. If `image.png` is placed in the subfolder `img`, the code would look like this:

```
\includegraphics[width=0.6\textwidth]{img/image.png}
```

Try adding a figure (with an image) to your document and refer to its given label by using the `ref` command as in lesson one. The `includegraphics` command supports a variety of file types, so most images will work.

## 2 Tables

Tables function very similar to Figures. The main difference is that table captions are usually placed above the actual table (rather than below the image when using figures). To create a table write:

```
\begin{table}
\centering
\caption{My table caption.}
\label{tab:mytable}
Table Content
\end{table}
```

To create the *actual* table to be placed at “Table Content”, we use the *tabular* environment. As the code to type it in manually can be a little cumbersome, we will make use of the Texmaker wizard. In the menubar of Texmaker, go to “Wizard” and then “Quick Tabular”. This gives you an interface to design the content and look of your table. If you are using a different editor from Texmaker, I recommend going to <http://www.tablesgenerator.com/> and using the interface there to create your code. The latter also supports the `booktabs` package, which improves the aesthetics of your tables.

Try creating a table using one of the above methods and add it to your document. Remember to reference it somewhere! While the details of the *tabular* environment has not been explained here, it should be somewhat apparent after using one of the wizards. To convert Excel sheets to L<sup>A</sup>T<sub>E</sub>X tables you can use the plugin found at <https://www.ctan.org/pkg/excel2latex>.

### 3 Mathematics

Mathematics in L<sup>A</sup>T<sub>E</sub>X can be used both inline, such as  $E = mc^2$ , or on a line with an equation number:

$$\sum_{i=1}^n i = \frac{(n+1)n}{2} \quad (1)$$

or as a number of aligned equations:

$$\begin{aligned} \frac{dx}{dt} &= -A\omega \sin(\omega t + \theta) \\ \frac{d^2x}{dt^2} &= -A\omega^2 \cos(\omega t + \theta) \end{aligned}$$

Before we begin, load the package `amsmath` by adding `\usepackage{amsmath}` to your preamble. Inline mathematics is written as a part of your text by surrounding the math code with the `$` character:

`... such as $E = mc^2$, or on a ...`

This also provides us with a method to easily reference variables, such as the mass  $m$  (by writing `$m$`), in our text. As the example shows, the ‘`^`’ character is used to give variables exponents. Mathematics is generally done by using various pieces of code to represent the equations that we want. I have supplied a few examples in the following table:

---

$\frac{a}{b}$	<code>\frac{a}{b}</code>
$a \cdot b$	<code>a \cdot b</code>
$a^b$	<code>a^{b}</code>
$\sqrt{a}$	<code>\sqrt{a}</code>
$\int_a^b x dx$	<code>\int_{a}^{b} x dx</code>
$\sum_{i=1}^N i$	<code>\sum_{i=1}^N i</code>

---

Texmaker has a “Math” menu that can help you out as well. As you may have noticed, there is a lot of things you can do with mathematics in  $\text{\LaTeX}$ , so I encourage you to explore on your own. Google can be very helpful! To have a labelled equation on its own line, use the following piece of code:

```
\begin{equation}\label{eq:myequation}
\sum_{i=1}^n i = \frac{(n+1)n}{2}
\end{equation}
```

To reference the equation use `\eqref{eq:myequation}`. To align more than one equation, the following code creates two equations aligned at ‘=’.

```
\begin{align*}
\frac{dx}{dt} &= - A \omega \sin(\omega t + \theta) \\
\frac{d^2 x}{dt^2} &= - A \omega^2 \cos(\omega t + \theta)
\end{align*}
```

Notice how ‘&’ dictates where to align the two equations and how `\\` makes sure the two equations are on separate lines. Finally, to create a matrix use the following code:

```
\[ A = \begin{pmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{pmatrix}
```

to create

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

Play around with making different formulas and equations – maybe even something you need for a project. It can be a bit daunting at first, but try to work through it! To test your skills, try creating the following set of equations in your document:

$$\Psi(x, 0) = \sum_{n=1}^{\infty} c_n \psi_n(x)$$

$$c_n = \sqrt{\frac{2}{a}} \int_0^a \sin\left(\frac{n\pi}{a}x\right) \Psi(x, 0) dx$$

The sine function is displayed using ‘`\sin`’. This changes the font so we don’t confuse it with a product of three variables. To make sure the parenthesis are stretched vertically, use ‘`\left(`’ and ‘`\right)`’ rather than just ‘`(`’ and ‘`)`’. The Greek letters in the above equation are ‘`\pi`’, ‘`\psi`’ and ‘`\Psi`’.

## 4 Units

When writing projects in the natural sciences, it is often necessary to write physical units in a variety of context. To make this process a little more manageable in L<sup>A</sup>T<sub>E</sub>X, we can use the `siunitx` package. After loading the package in your preamble with `\usepackage{siunitx}`, we can represent units using the `\si{}` command. As an example, to write:

$$g = 9.8 \text{ m s}^{-2}$$

use the code

```
\[ g = 9.8 \si{\metre\per\square\second} \]
```

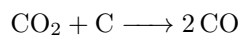
or

```
\[ g = 9.8 \si{ms^{(-2)}} \]
```

The `siunitx` package has a ton of options for typesetting physical units and numbers in general. The full documentation can be found at <https://www.ctan.org/pkg/siunitx>

## 5 Chemical formulae

To typeset chemical molecular formulae and equations, we can make use of the `mhchem` package. First, load it in your preamble using `\usepackage[version=4]{mhchem}`. Now we can typeset chemical equations using the `\ce{}` command. To write H<sub>2</sub>O, simply write `\ce{H2O}`. For a reaction, use `->` as the “arrow”. For example, to output



simply write `\ce{CO2 + C -> 2 CO}`. Similar to the `siunitx` package, the `mhchem` package is quite extensive, and can typeset a wide variety of chemical formulas. See the documentation at <https://www.ctan.org/pkg/mhchem> for further details.