The ELSAToolbox Package User Manual

IT Group, ELSA Lab

Department of Computer Science
National Tsing Hua University
Taiwan
June 23, 2021

Contents

Chapter 1

Quickstart

The `elsatoolbox` is a package that includes several commonly used packages, which make preparing LaTeX documents easier. In this chapter, we provide an overview of the usage of `elsatoolbox` as well as some guidelines on LaTeX writing. For the remainder of the article, we provide a number of examples and the usage of the included packages in the remainder of this article. The following chapters are organized as follows. Chapter 2 introduces the `algorithm2e` package for typesetting algorithms in pseudocode. Chapter 3 summarizes the usage of a collaborative annotation tool, FiXme. Chapter 4 demonstrates cross referencing with the `zref` package. Chapter 5 deals with figures and captions. Chapter 6 shows the usage of the `hyperref` package for enabling typesetting of hyperlinks. Chapter 7 presents the mathematics typesetting. Chapter 8 provides solutions to adjust the spacing between lines in a LaTeX document. Chapter 9 explains the typesetting of tables.

Usage

1. Create a new LaTeX project with template files from the targeted venue.
2. Copy the style file `elsatoolbox.sty` to the new project.
3. Use the `elsatoolbox` package by adding `\usepackage{elsatoolbox}`.

In practice, it is very likely that there are some warnings and errors. Most of them are due to package conflicts or incorrect commands. It is the users' responsibility to check and eliminate all warnings and error messages before proceeding. Finally, feel free to edit the style file after copying it to the new project.

Draft Mode

The `elsatoolbox` package provides an option for enabling draft mode. In order to enable it, add the option `draft` to `elsatoolbox` as the following:

```
\usepackage[draft]{elsatoolbox}.
```

After the draft mode is enabled, the following features are turned on:

1. "Draft Mode" showing at the top of each page
2. FiXme annotations
3. The list of corrections showing at the fist page
4. Colored hyperlinks

Choosing Packages

There are many packages that offer similar or identical functionality. It's recommended to select the package by the following order of preferences:

1. Choose more recently maintained packages
2. Choose packages with no compatibility issues
3. Choose simpler packages suited for users' needs

Dos

1. Read author guidelines, especially for strict limitations, such as paper length.
2. Use consistent words and code
3. Use proofreading tools, such as Grammarly
4. Follow KISS principle: Keep it simple, stupid (Write it simple)
5. Follow YAGNI principle: You aren't gonna need it (Include only necessary stuff)
6. Follow DRY principle: Don't repeat yourself (Avoid duplication)

Don'ts

1. Do not change the margins if users are preparing the manuscript of the targeted conference of journal
2. Do not adjust the font size of the main text
3. Do not use \\ to break a line (Instead, use a blank line)
4. Do not use any spacing commands `\vspace` or `\hspace` within the main contents

Better Do It

1. Output the image at least 300 dpi
2. Make sure the characters can be clearly identified in the images
3. Sharpen the images by Unsharp mask filter
4. Put the figures, tables, and algorithms in "figures", "tables", and "algorithms" folders, respectively
5. Include comments when the code can not explain what it does
6. Check the warnings from compilers

Better Avoid It

1. Should not adjust the spacing too much
2. Should not adjust the font size too much (of captions and references)
3. Should not use outdated or old-fashioned packages
4. Should not write complex code for trivial improvement

Support

The `elsatoolbox` is maintained by the IT group of ELSA Lab. Open an issue through our GitHub repository[1] if one has any questions.

[1] `https://github.com/elsa-lab/elsatoolbox`

Chapter 2

Algorithms

There are four common packages, `algorithmic`, `algorithm2e`, `algorithmicx`, and `program`, for typesetting algorithms in form of pseudocode. They provide stylistic enhancements over a uniform style (i.e., all in typewriter font) so that constructs such as loops or conditional statements are visually separated from other text. In this chapter, we introduce `algorithm2e`, which is loaded by `elsatoolbox`. For the other packages, please check out the Wikibooks of LaTeX[1].

The `algorithm2e` package provides a floating `algorithm` environment for wrapping pseudocode and provides commands for writing pseudocode. Please note that this package is not compatible with the following packages: `algorithm`, `algorithmic`, and `algpseudocode`. In order to disable `algorithm2e`, add the option `noalgo` to `elsatoolbox` as the following:

```
\usepackage[noalgo]{elsatoolbox}.
```

Usage

Typically, each statement of algorithms should be ended with `\;`. In the following paragraphs, we list the commonly used commands provided by `algorithm2e`. For more advanced usage, please check out the documentation on CTAN[2].

Customization

```
\DontPrintSemicolon  \SetAlgoCaptionSeparator[s]{<sep>}
```

Input, Output, Basic Keywords

```
\KwIn{<input>}     \KwData{<input>}      \KwTo
\KwOut{<output>}   \KwResult{<output>}   \KwRet{<value>}
```

[1] https://en.wikibooks.org/wiki/LaTeX/Algorithms
[2] https://www.ctan.org/pkg/algorithm2e

Control Flow

`\If{<condition>}{<block>}`	`\For{<condition>}{<loop>}`
`\ElseIf{<condition>}{<block>}`	`\While{<condition>}{<loop>}`
`\Else{<block>}`	`\ForEach{<condition>}{<loop>}`
`\Switch{<condition>}{<block>}`	`\ForAll{<condition>}{<loop>}`
`\Case{<case>}{<block>}`	`\Repeat{<condition>}{<loop>}`

Example

The following example creates Algorithm 1.

```
\begin{algorithm}[H]
    \KwData{this text}
    \KwResult{how to write algorithm with \LaTeX2e }
    initialization\;
    \While{not at end of this document}{
        read current\;
        \If{understand}{
            go to next section\;
            current section becomes this one\;
        }
        \Else{
            go back to the beginning of current section\;
        }
    }
\caption{How to write algorithms}
\label{algo:howto}
\end{algorithm}
```

Algorithm 1: How to write algorithms

Data: this text
Result: how to write algorithm with LaTeX2e

```
initialization;
while not at end of this document do
    read current;
    if understand then
        go to next section;
        current section becomes this one;
    end
    else
        go back to the beginning of current section;
    end
end
```

Chapter 3

Annotations

Annotating a document here refers to inserting notes that do not belong to the document for development or reviewing purposes. Such notes may involve different importance levels, ranging from simple "fix the spelling" flags to critical "this paragraph is a lie" mentions. Annotations like this should be visible during the development or reviewing phase, but should disappear in the final version of the document.

3.1 Basic Usage

FiXme provides four different levels for inserting notes: `note`, `warning`, `error`, and `fatal`. Users may insert notes through the command:

`\fx<level>[<option>]{<note>}`.

The most commonly used option is `author=<name>`, which tags the author of the inserted note. For example, one may want to leave a note to the other collaborators, he/she can use the command under the draft mode:

`\fxnote[author=someone]{note to be inserted}`.

Additionally, we list the example commands for inserting different levels of annotations as follows.

Commands	Annotations
`\fxnote[author=Anna]{This is a note.}`	Anna: This is a note.
`\fxwarning[author=Belle]{This is a warning.}`	Belle: This is a warning.
`\fxerror[author=Cindy]{This is an error.}`	Cindy: This is an error.
`\fxfatal[author=Dora]{This is a fatal.}`	Dora: This is a fatal.

3.2 Highlighting Text

Sometimes, users might want to insert a note and highlight the relevant part of the text to which it applies. FiXme provides starred versions of its annotation commands to do that. For example, the following phrase contains a typo: the fature representation. One can highlight the typo with the command:

```
\fxerror*[author=Elsa]{This is a typo}{fature}.
```

3.3 Registering Authors

FiXme offers a command to registers a new author:

```
\FXRegisterAuthor{<cmdprefix>}{<envprefix>}{<author>}.
```

It takes three arguments, where the last argument `<author>` is simply the name of the author to be registered. For the former two arguments, `<cmdprefix>` and `<envprefix>` stand for the prefix of commands and environments[1] created by FiXme later, respectively. Suppose that we have registered Fiona like this:

```
\FXRegisterAuthor{fon}{afon}{Fiona}.
```

After that, Fiona can use the commands `\fonnote`, `\fonwarning` etc., along with their starred versions. For the same example: the fature representation. Fiona can highlight the typo with the command:

```
\fonerror*{This is a typo}{fature}.
```

Warning `<cmdprefix>` and `<envprefix>` need to be different. The technical reason is that in LaTeX, an environment named `foo` is defined in terms of two commands: `\foo` and `\endfoo` (the first one should be `\beginfoo`). Consequently, if one uses the same prefix, he/she will get a name clash between the annotation commands and environments.

[1] For the environments provided by FiXme, please check out the documentation on CTAN: `https://www.ctan.org/pkg/fixme`.

Chapter 4

Cross Referencing

The `elsatoolbox` package allows cross-referencing across two independent documents, `main.tex` and `supp.tex`. In other words, users can have a reference in `main.tex` to something labelled in `supp.tex`, without including `supp.tex` in `main.tex`. For example, `\zref{supp:fig:example}` can easily reference Fig. S1 of `supp.tex`. There are two steps required to enable cross-referencing:

1. Create a new file named `latexmkrc` and add the following code to it.
2. Clear cached files and recompile.

```
# Preserve .aux files for cross referencing
# Reference: https://www.overleaf.com/learn/how-to/
    Cross_referencing_with_the_xr_package_in_Overleaf

add_cus_dep( 'tex', 'aux', 0, 'makeexternaldocument' );

sub makeexternaldocument {
    if ($root_filename eq $_[0])
    {
        # FOR PDFLATEX
        system( "latexmk -pdf \"$_[0]\"" );

        # FOR LATEX+DVIPDF
        # system( "latexmk \"$_[0]\"" );

        # FOR XELATEX
        # system( "latexmk -xelatex \"$_[0]\"" );

        # FOR LUALATEX
        # system( "latexmk -lualatex \"$_[0]\"" );
   }
}
```

Chapter 5

Floats

Floats are containers for things in a document that cannot be broken over a page. The LaTeX typesetting system by default recognizes "table" and "figure" floats. In this chapter, we first explain floating figures in detail. We then briefly introduce the adjustment of the captions of floats. Note that floating tables are mainly covered in Chapter 9.

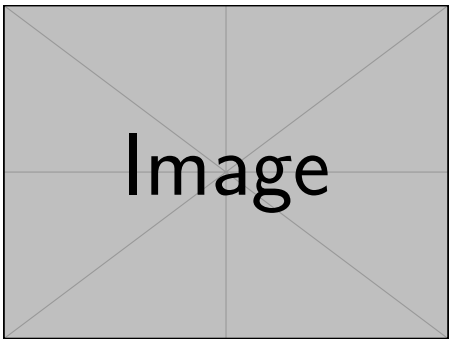
5.1 Figures

To import external graphics, one can insert the external graphic with the command:

`\includegraphics[<options>]{<file>}`.

The most commonly used option is `width=<value>`, which scales the graphic to this value. For example, the following command imports an example image, and LaTeX itself treats the image like normal text:

```
\includegraphics[width=.4\linewidth]{example-image}
```



To create a figure that floats, use the `figure` environment.

```
1 \begin{figure}[placement specifier]
2 ... figure contents ...
3 \end{figure}
```

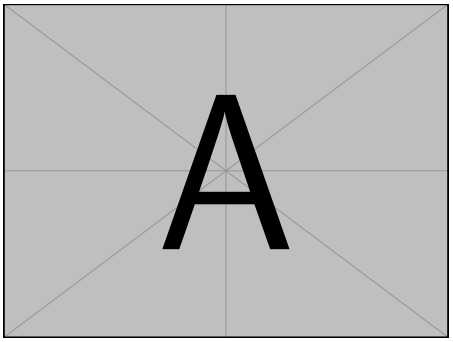


Figure 5.1: An example image with an A.

The `placement specifier` gives a greater degree of control over where certain floats are placed. We list the commonly used specifiers as follows.

Specifier	Permission
h	Place the float here, i.e., approximately at the same point it occurs in the source text (however, not exactly at the spot)
t	Position at the top of the page.
b	Position at the bottom of the page.

The following example creates Fig. 5.1 that placed at the the top of the page:

```
\begin{figure}[t]
    \centering
    \includegraphics[width=.4\linewidth]{example-image-a}
    \caption{An example image with an A.}
    \label{fig:example-a}
\end{figure}
```

Instead of placing floats on the current page, one may consider moving floats to the next page. To achieve this, surround the float with `\afterpage{<float>}` to make it appear after the current page. For example, the following commands create Fig. 5.2 that placed at the top of the following page:

```
\afterpage{
\begin{figure}[t]
    \centering
    \includegraphics[width=.4\linewidth]{example-image-b}
    \caption{An example image with a B.}
    \label{fig:example-b}
\end{figure}
}
```

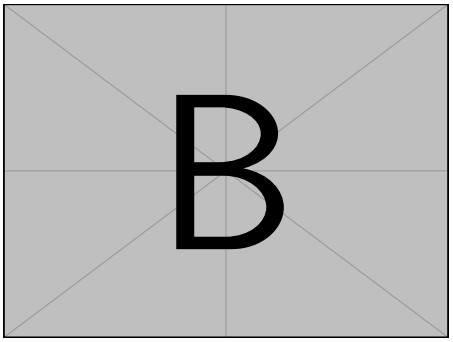


Figure 5.2: An example image with a B.

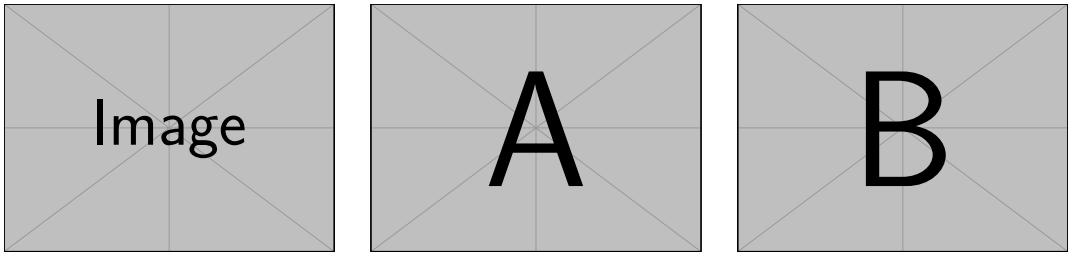
5.2 Subfloats

In this section, we introduce the `subcaption` package that allows users to have subfloats within floats. Please note that `elsatoolbox` loads `subcaption` that may be incompatible with some template files. To solve this issue, the `subfig` package is a useful alternative to `subcaption`. Before using `subfig`, users should disable `subcaption` by adding the option `nosubcap` to `elsatoolbox` as the following:

`\usepackage[nosubcap]{elsatoolbox}`.

The following example creates Fig. 5.3.

```
\begin{figure}[t]
\centering
\begin{subfigure}[t]{.33\linewidth}
    \centering
    \includegraphics[width=.9\linewidth]{example-image}
    \subcaption{\texttt{example-image}.}
    \label{fig:exmples-a-b:example-image}
\end{subfigure}% <-this % stops a space
\begin{subfigure}[t]{.33\linewidth}
    \centering
    \includegraphics[width=.9\linewidth]{example-image-a}
    \subcaption{\texttt{example-image-a}.}
    \label{fig:exmples-a-b:example-image-a}
\end{subfigure}% <-this % stops a space
\begin{subfigure}[t]{.33\linewidth}
    \centering
    \includegraphics[width=.9\linewidth]{example-image-b}
    \subcaption{\texttt{example-image-b}.}
    \label{fig:exmples-a-b:example-image-b}
\end{subfigure}
\caption{Example subfigures using \texttt{subcaption}.}
\label{fig:exmples-a-b}
\end{figure}
```



(a) `example-image`. (b) `example-image-a`. (c) `example-image-b`.

Figure 5.3: Example subfigures using `subcaption`.

5.3 Captions

Although adjusting the vertical spacing between floats and captions to compact/relax the main text is not recommended, it is wiser to use smaller floats or reduce words. We still provide ways to customize captions using `subcaption` as following:

```
% Customize figure environment
\captionsetup[figure]{
    font=small, % Font size option
    skip=1ex    % Vertical space between float and caption
}

% Customize table environment
\captionsetup[table]{
    font=small, % Font size option
    skip=1ex    % Vertical space between float and caption
}

% Customize subfigure environment
\captionsetup[sub]{
    font=small, % Font size option
    skip=1ex    % Vertical space between subfloat and subcaption
}
```

Chapter 6

Hyperlinks

The `elsatoolbox` package loads the `hyperref` package, which allows users to produce hypertext links. There are three common commands: `\url{<URL>}`, `\href{<URL>}{<text>}`, and `\hyperref[<label>]{<text>}`. For example,

Commands	Outputs
`\href{https://google.com}{Google}`	Google
`\url{https://google.com}`	`https://google.com`
`\hyperref[content:quickstart]{Quickstart}`	Quickstart

Chapter 7

Mathematics

Typesetting mathematics is one of LaTeX's greatest strengths. It is also a large topic due to the existence of so much mathematical notation. In this chapter, we are not going to go through the typesetting in detail. Instead, we only mention the principle of it in the following sections. For those who desire to learn from scratch, the Wikibooks of LaTeX[1] contains extensive examples and explanations.

7.1 Mathematics Environments

The following table summarizes special environments for typesetting math notation.

Type	Inline	Displayed	Numbered and displayed
Environment	`math`	`displaymath`	`equation`[2]
LaTeX shorthand	`\(...\)`	`\[...\]`	
TeX shorthand	`$...$`	`$$...$$`	

Suggestion Using the `$$...$$` should be avoided, as it may cause problems, particularly with the AMS-LaTeX macros. Furthermore, should a problem occur, the error messages may not be helpful.

In order for some operators, such as lim or $\sum$, to be displayed correctly inside some math environments (read `$...$`), it might be convenient to write the `\displaystyle` class inside the environment. Doing so might cause the line to be taller, but will cause exponents and indices to be displayed correctly for some math operators. For example, $\displaystyle\sum_{i=0}^\infty$ (`$\displaystyle\sum_{i=0}^\infty$`) is preferable to $\sum_{i=0}^\infty$ (`$\sum_{i=0}^\infty$`).

[1] `https://en.wikibooks.org/wiki/LaTeX/Mathematics`

[2] The starred version `equation*` suppresses numbering.

7.2 Brackets, Braces, and Delimiters

Mathematical features will differ in size frequently, in which case the delimiters surrounding the expression should vary accordingly. This can be done automatically using the `\left`, `\right`, and `\middle` commands. For examples,

```
\left(\frac{x^2}{y^3}\right)
```
$$\left(\frac{x^2}{y^3}\right)$$

```
P\left(A=2\middle\vert\frac{A^2}{B}>4\right)
```
$$P\left(A=2\middle|\frac{A^2}{B}>4\right).$$

Curly braces are defined differently by using `\left\{` and `\right\}`,

```
\left\{\frac{x^2}{y^3}\right\}
```
$$\left\{\frac{x^2}{y^3}\right\}$$

If a delimiter on only one side of an expression is required, then an invisible delimiter on the other side may be denoted using a period (`.`).

```
\left.\frac{x^3}{3}\right\vert_0^1
```
$$\left.\frac{x^3}{3}\right|_0^1$$

7.3 Horizontal Spacing

Suppose one is trying to display the following equation:

$$\int y\,\mathrm{d}x,$$

he/she may write "`\int y \mathrm{d}x`". However, this results in the equation below instead of the one above.

$$\int y\mathrm{d}x$$

In this situation, a `\quad` would clearly be overkill. What is needed are some small spaces to be utilized in this type of instance, and that's what LaTeX provides:

Commands	Description	Commands	Description
`\,`	small space	`\;`	large space
`\:`	medium space	`\!`	negative space

By taking advantage of these horizontal spacing commands, he/she is able to rectify the above problem using "`\int y \, \mathrm{d}x`".

Chapter 8

Spacing

The `elsatoolbox` package loads the `setspace` package, which allows users to adjust the spacing between lines in a document. However, as mentioned in Section 5.3, adjusting the spacing is not recommended. It is wiser to reduce words. In this chapter, we first introduce the `spacing` environment provided by `setspace` in the first section. We then show several example commands for adjusting various types of spacing in the following sections. Please note that `setspace` is incompatible with the `beamer` LaTeX class. To disable `setspace`, add the option `nosetspace` to `elsatoolbox` as the following:

```
\usepackage[nosetspace]{elsatoolbox}
```

8.1 The `spacing` Environment

The `spacing` environment allows users to adjust the spacing between lines in a local fashion.

```
1 \begin{spacing}{<stretch>}
2 ... contents ..
3 \end{spacing}
```

The default `<stretch>` is set to 1. Smaller `<stretch>` will have less vertical spacing. For example, the following commands reduce the vertical spacing between equations as shown beside.

```
1 \begin{spacing}{0.5}
2     \[ x=y \]
3     \[ x=y \]
4 \end{spacing}%
```

$$x = y$$
$$x = y$$

8.2 Fixed-length Spaces

The following example commands change the vertical spacing of `\smallskip`, `\medskip`, and `\largeskip`.

```
\setlength{\smallskipamount}{3.0pt plus 1.0pt minus 1.0pt}
\setlength{\medskipamount}{6.0pt plus 2.0pt minus 2.0pt}
\setlength{\bigskipamount}{12.0pt plus 4.0pt minus 4.0pt}
```

8.3 Floats and Text

There are two common types of layout for typesetting LaTeX documents, the single column and the double column. Therefore, commands are different when using different layout types. For each layout type, the first example command adjusts the vertical spacing between two floats, and the second example command is for changing the vertical spacing between the last floats and the first textline.

Single Column	**Double Column**
`\setlength{\floatsep}{2ex}`	`\setlength{\dblfloatsep}{2ex}`
`\setlength{\textfloatsep}{1ex}`	`\setlength{\dbltextfloatsep}{1ex}`

8.4 Paragraph Formatting

There are two different spacing related to paragraphs, horizontal spacing before the first line of a paragraph and vertical spacing between two paragraphs. The command `\parindent` controls the former horizontal spacing, and the command `\parskip` controls the latter vertical one. One can use the following example commands to change these spacing:

```
\setlength{\parindent}{1em}  \setlength{\parskip}{0.0pt plus 1.0pt}
```

8.5 Display Style Formula

There are four commands to control the amount of vertical space before, and after, a displayed equation. Users may adjust these spacing via the following example commands:

```
\setlength{\abovedisplayskip}{10.0pt plus 2.0pt minus 5.0pt}
\setlength{\belowdisplayskip}{10.0pt plus 2.0pt minus 5.0pt}
\setlength{\abovedisplayshortskip}{0.0pt plus 3.0pt}
\setlength{\belowdisplayshortskip}{6.0pt plus 3.0pt minus 3.0pt}
```

Chapter 9

Tables

The LaTeX has built-in support to typeset tables and provides two environments, `tabular` and `table`. To typeset material in rows and columns, the `tabular` environment is needed. In this chapter, we briefly go through the `tabular` environment and the `table` environment. For the advance usage of `tabular` and the usage of the other packages loaded by `elsatoolbox` (i.e., `tabu`, `tabularx`, and `tabulary`), please check out the Wikibooks of LaTeX[1].

9.1 The `tabular` Environment

To typeset tables with optional horizontal and vertical lines, one can use the `tabular` environment. The width of each column of tables is determined automatically.

```
\begin{tabular}[pos]{table spec}
... table contents ...
\end{tabular}
```

The `table spec` argument tells LaTeX the alignment to be used in each column and the vertical lines to insert. The number of columns does not need to be specified as it is inferred by looking at the number of arguments provided.

Table Spec	Description
`l`	left-justified column
`c`	centered column
`r`	right-justified column
`p{<width>}`	paragraph column with text vertically aligned at the top
`m{<width>}`	paragraph column with text vertically aligned in the middle
`b{<width>}`	paragraph column with text vertically aligned at the bottom
`|`	vertical line between columns

[1] `https://en.wikibooks.org/wiki/LaTeX/Tables`

By default, if the text in a column is too wide for the page, LaTeX won't automatically wrap it. Using `p{<width>}` you can define a special type of column which will wrap-around the text as in a normal paragraph. You can pass the width using any unit supported by LaTeX, such as 'pt' and 'cm', or command lengths, such as `\textwidth`.

The optional parameter `pos` can be used to specify the vertical position of the table relative to the baseline of the surrounding text. In most cases, you will not need this option.

Specifier	Permission
b	bottom
c	center (default)
t	top

This example shows how to create a simple table, which is a 3×3 table.

```
\begin{tabular}{ l | c | r }
    \toprule
    1 & 2 & 3 \\ \hline
    4 & 5 & 6 \\ \hline
    7 & 8 & 9 \\
    \bottomrule
\end{tabular}
```

1	2	3
4	5	6
7	8	9

9.2 The `table` Environment

The optional `table` environment is a container for floating material similar to `figure`, into which a `tabular` environment may be included. The following example shows how to create a floating tables.

```
\begin{table}[h]
\centering
\begin{tabular}{ l | c | r }
    \toprule
    1 & 2 & 3 \\ \hline
    4 & 5 & 6 \\ \hline
    7 & 8 & 9 \\
    \bottomrule
\end{tabular}
\caption{An example table.}
\label{tab:example}
\end{table}
```