Typesetting Technical Reports

Rwitaban Goswami and Mihir Vahanwala

July 10, 2020

Contents

1 Introduction

You now know basic LaTeX, but what about all the fancy maths and equations stuff which is what LaTeX is pretty much known for the most outside the typesetting world? This week we bring to you the tools which will help you typeset technical reports, which includes, but is not limited to typesetting mathematics

2 Packages to load in the preamble

For the majority of purposes, when you need to typeset technical reports involving maths, your preamble will look something like this

```latex
\documentclass[12pt, letterpaper]{article}
\usepackage{amsmath} %see
\usepackage{amsthm}   %explanation
\usepackage{amssymb} %below

\usepackage{esint} %more integration symbols

\usepackage{graphicx} %for figures
\usepackage{float} %for formatting figures and tables
\usepackage{floatrow} %for formatting figures and
    tables
\usepackage{listings} %to highlight code
\usepackage{hyperref} %for referencing
\usepackage{url}

%We will explain this later
\newtheorem{theorem}{Theorem}
\newtheorem{lemma}{Lemma}
\newtheorem{corollary}[theorem]{Corollary}
\newtheorem{proposition}{Proposition}[section]

\theoremstyle{remark}
\newtheorem*{remark}{Remark}
```

The amsmath package makes typesetting formulae and equations much more

convenient. The amsthm package makes it a lot easier to define theorem-like environments, which are crucial to write any mathematical text. Finally, amssymb, as the name suggests, incorporates a few cool mathematical symbols into our arsenal. This TeX StackExchange thread gives a helpful technical summary. [8]

It's important to note that amsthm must be loaded *after* amsmath. Although we will deal with the package briefly, we strongly recommend reading the documentation[5] for quick pointers.

Alright, we're good to go!

3 Equations and formulae

3.1 Math mode

Whenever you need to typeset mathematics and equations, you will need to be in a special mode that LaTeX calls maths mode. There are two types of math modes:

Inline maths The typesetted maths sits inline, and is a part of a line which in general is not typesetted in math environment, generally used for simple statements such as $a = b$ to be embedded inside a line

Math environment The typesetted maths sits in its own space, generally used for large or important equations or a number of equations

$$\vec{E} = \frac{1}{4\pi\epsilon_0} \frac{q\left(\vec{r} - \vec{r}_0\right)}{|\vec{r} - \vec{r}_0|^3}$$

Generally you would not write text in math mode, as it will be rendered *likethis*, and no spaces are rendered in math mode

3.2 Inline Maths

If you'd like mathematical symbols in the midst of your text, place them between two \$ signs. For example,

```
1  If a function $f$ is continuous, then for every $\
      varepsilon > 0$, there is a $\delta$, such that $||\
      mathbf{x}-\mathbf{x_0}|| < \delta$ implies $|f(\
      mathbf{x}) - f(\mathbf{x_0})| < \varepsilon$
```

typesets

> If a function f is continuous, then for every $\varepsilon > 0$, there is a $\delta > 0$, such that $||\mathbf{x} - \mathbf{x_0}|| < \delta$ implies $|f(\mathbf{x}) - f(\mathbf{x_0})| < \varepsilon$.

Alternate delimiters for inline maths include:

```
1  This is the \[ \varepsilon-\delta\] definition. \\
2  This is the $$\epsilon-\delta$$ defintion.
```

This typesets:

> This is the
> $$\varepsilon - \delta$$
> definition.
> This is the
> $$\epsilon - \delta$$
> defintion.

Note the difference between `\varepsilon` and `\epsilon`.

From one simple line, we have observed how to typeset the Greek alphabet [3]: here's an example, it works for all letters.

```
$\Theta$ for upper case, $\theta$ for lowercase.
```

typesets

> Θ for upper case, θ for lowercase.

We have observed how to typeset bold symbols with `\mathbf`. However, for Greek letters, `\mathbf{omega}` doesn't typeset $\boldsymbol{\omega}$, you have to use `\boldsymbol{\omega}`

We have also observed how to typeset subscripts. However, be careful when a subscript or superscript has multiple characters! An example:

```
We have thus proven that $p_{ij}^{\alpha+\beta}$ is
    algebraic. Now consider arbitrary $t \in \mathbb{R}$
    ...
```

typesets:

> We have thus proven that $p_{ij}^{\alpha+\beta} \in \mathbb{C}$ is algebraic. Now consider arbitrary $t \in \mathbb{R}$...

Woah! Did you notice that? We just saw how to typeset a few more fancy symbols!

When you write the representation of sets, make sure you escape the special meaning of the curly braces with a backslash.

```
$S := \{ x | ~x \equiv 1 \pmod {9} \}$. $T := \{x | ~x
    \equiv 1 \pmod {3}\}$. $S \subset T$.
```

typesets:

> $S := \{x|\ x \equiv 1 \pmod 9\}$. $T := \{x|\ x \equiv 1 \pmod 3\}$. $S \subset T$.

Note the ~ (tilde) to ensure a space between symbols in math mode.

There are several symbols you'd want to typeset with LaTeX (including this one! Try it, type \LaTeX), and you'd find this list [1] comprehensive.

Here's a short passage that covers some features the list does not include:

> A well known expression of $\binom{n}{k}$ is $\frac{n!}{k!(n-k)!}$. $n!$ denotes the factorial function for non-negative integers, and is recursively defined as $0! := 1$, $n! := n \cdot (n-1)!$
> $n! = \prod_{j=1}^{n} j$. A common extension of the factorial function is the Gamma function. For positive integers n,
> $(n-1)! = \Gamma(n) = \int_0^\infty x^{n-1}e^{-x}dx$
> It follows from the binomial theorem that $\sum_{k=0}^{n} \binom{n}{k} = 2^n$. The binomial theorem is incredibly powerful, and can be used to approximate $\sqrt{1+x}$, or even $\sqrt[71]{1+x}$.

Solution:

```
A well known expression of $\binom{n}{k}$ is $\frac{n
    !}{k!(n-k)!}$.
$n!$ denotes the factorial function for non-negative
    integers, and is recursively defined as \\
$0! := 1$, $n! := n \cdot (n-1)!$

$n! = \prod_{j=1}^{n}j$. A common extension of the
    factorial function is the Gamma function. For
    positive integers $n$,
```

```
6
7   $(n-1)! = \Gamma(n) = \int_{0}^{\infty}x^{n-1}e^{-x}dx$
8
9   It follows from the binomial theorem that $\sum_{k=0}^{
    n}\binom{n}{k} = 2^n$. The binomial theorem is
    incredibly powerful, and can be used to approximate
    $\sqrt{1+x}$, or even $\sqrt[71]{1+x}$.
```

Sometimes, you'd want regular text in math mode; say, while writing

$P \subseteq NP$. However, we know that testing the positivity of the term residue(n) is decidable in coNP$^{\text{RP}}$.

Here's how we do it:

```
1  $\text{P} \subseteq \text{NP}$. However, we know that
      testing the positivity of the term $\text{residue}(n
      )$ is decidable in $\text{coNP}^{\text{RP}}$.
```

3.3   The Equation environment

Equations are often pivotal to the report, and you want them to stand out. Here's where you use the equation environment. Here's a sample for demonstrating the effect it creates.

Charles' Law, Boyle's Law, Gay-Lussac's Law and Avogadro's law were empirical rules relating the pressure, volume, temperature and amount of gas. These theories culminated in the elegant Ideal Gas Equation.

$$pV = nRT \tag{1}$$

How do you do it?

```
1  \begin{equation}
2  pV = nRT
3  \label{eq:idealgas}
4  \end{equation}
```

It's that simple. Notice the label; it helps if you want to refer to the equation in the later part of the document.

Sometimes, when you're sure you won't need to reference the equation, you might decide to do away with the number for that particular equation. Here are two ways of doing that:

```
1  Vanderwaal proposed the real gas equation as a more
       accurate model.
2  \begin{equation*}
3  \left(p + \frac{an^2}{V^2}\right)(V - nb) = nRT
4  \end{equation*}
5  If you introduce a variable compressibility factor $Z$,
       this can be expressed as
6  \begin{equation}
7  pV = ZnRT \nonumber
8  \end{equation}
```

This typesets to:

Vanderwaal proposed the real gas equation as a more accurate model.

$$\left(p + \frac{an^2}{V^2}\right)(V - nb) = nRT$$

If you introduce a variable compressibility factor Z, this can be expressed as

$$pV = ZnRT$$

Often, it so happens that you want to write several equations one after the other. We strongly recommend the align environment, that is part of the amsmath package. Do not use eqnarray, which is simply base LaTeX. [7]

Use the ampersand (&) right before the symbol you want to align. Use double backslash (\\) to signify a new line. Here's an example:

```
1  \begin{align*}
2  \boldsymbol{\nabla} \cdot \mathbf{E} &= \frac{\rho}{\
       varepsilon_0} \\
3  \boldsymbol{\nabla} \cdot \mathbf{B} &= 0 \\
4  \boldsymbol{\nabla} \times \mathbf{E} &= -\frac{\
       partial \mathbf{B}}{\partial t} \\
5  \boldsymbol{\nabla} \times \mathbf{B} &= \mu_0\mathbf{j
       } + \frac{1}{c^2}\frac{\partial \mathbf{E}}{\partial
```

```
     t }
6  \end{ align ∗}
```

$$
\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}
$$

$$
\nabla \cdot \mathbf{B} = 0
$$

$$
\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}
$$

$$
\nabla \times \mathbf{B} = \mu_0 \mathbf{j} + \frac{1}{c^2}\frac{\partial \mathbf{E}}{\partial t}
$$

3.4   Matrices

Typesetting matrices is straightforward; very similar to tables. Here's an example:

```
1  The  companion  matrix  $M$  is  given  as :
2  $$\ begin { bmatrix }
3          0 & 1 & 0 & \dots & 0 \\
4          0 & 0 & 1 & \dots & 0 \\
5          \vdots & \vdots & \vdots & \ddots & \vdots \\
6          0 & 0 & 0 & \dots & 1 \\
7          a_0 & a_1 & a_2 & \dots & a_{\kappa−1}
8  \end{ bmatrix }$$
9  $M$  is  invertible  if  and  only  if  $a_0 \neq 0$
```

The companion matrix M is given as:

$$
\begin{bmatrix}
0 & 1 & 0 & \dots & 0 \\
0 & 0 & 1 & \dots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \dots & 1 \\
a_0 & a_1 & a_2 & \dots & a_{\kappa-1}
\end{bmatrix}
$$

M is invertible if and only if $a_0 \neq 0$

Explore the different styles of matrices the amsmath package allows you to typeset! [4]

3.5   Macros: Custom commands

Very often, you'll find yourself typing a cumbersome combination of commands, over and over again. You can create custom commands to save yourself the boredom. These are called macros. Macros can take multiple arguments.

Suppose you are writing a report that is based on Linear Algebra, and uses the notion of inner product heavily. Obviously, an expression like $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ is going to show up left and right. This what you'd do, if you had to brute force every time:

```
1 $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$
2 $\langle \boldsymbol{\theta}, \boldsymbol{\zeta} \
     rangle$
```

ad nauseum.

What you could do instead, is observe that the inner product requires two arguments, and write a macro. The macro is to be defined in the preamble. (before you declare `\begin{document}`)

```
1 \newcommand{\innerproduct}[2]{\langle \boldsymbol{ #1 }
     , \boldsymbol{ #2 } \rangle}
```

`\innerproduct` is how the custom command will be invoked. The [2] is the number of arguments it will take. Specifying this is optional. If this is not specified, the custom command doesn't take any arguments. #1 and #2 specify the first and second arguments that will be supplied to our custom command. So, to typeset $\langle \boldsymbol{u} + \boldsymbol{w}, \boldsymbol{v} \rangle$, simply write

```
1 $\innerproduct{u+w}{v}$
```

Suppose you're writing about a super efficient algorithm, and every second sentence contains the expression $\mathcal{O}(1)$. If you're particularly lazy, you could make a macro and save yourself the effort of typing out a few extra characters every time.

```
1 \newcommand{\Oone}{\mathcal{O}(1)}
```

should do the job. Please note that macro names can only contain alphabetic characters. [2] Also, this is an example of a macro that doesn't take any arguments.

11

If your custom command has the same name as one that is already provided in a package you've imported, it is a redefinition. The same syntax is followed, except `\newcommand` is replaced by `\renewcommand`

4   Theorems

Recall, in the preamble, we have imported the `amsthm` package. The following lines in the preamble made it very convenient to define theorem like environments.

```
1  \newtheorem{theorem}{Theorem}
2  \newtheorem{lemma}{Lemma}
3  \newtheorem{corollary}[theorem]{Corollary}
4  \newtheorem{proposition}{Proposition}[section]
5
6  \theoremstyle{remark}
7  \newtheorem*{remark}{Remark}
```

Theorem 1 (Optional clarification)**.** *This is our first theorem.*

Lemma 1. *This is our first lemma. LATEX maintains a separate counter for theorems and lemmas.*

Corollary 2. *This is a corollary. Because we supplied "theorem" as an optional argument in square brackets when we defined the environment, Corollaries and Theorems share a common counter.*

Theorem 3. *Guess this proves our point about the counter.*

Lemma 2. *We hope you're convinced.*

Remark. Because of that asterisk, LATEX does not maintain a counter for remarks. Also, we changed the style from plain(default) to remark.

Proof. Behold the power of the `amsthm` package; it provides a proof environment as well. Socrates is a man. All men are mortal. Therefore, Socrates is mortal. □

Proposition 4.1. *Because of the optional `section` argument we supplied while defining the proposition environment, the counter resets every time there's a new section.*

Typesetting this is as easy as invoking environments.

```
1
2  \begin{theorem}[Optional clarification]
3  \label{mytheorem}
4  This is our first theorem.
5  \end{theorem}
6
7  \begin{lemma}
8  \label{mylemma}
9  This is our first lemma. \LaTeX~ maintains a separate
       counter for theorems and lemmas.
10 \end{lemma}
11
12 \begin{corollary}
13 \label{mycorollary}
```

```
14   This is a corollary. Because we supplied "theorem" as
         an optional argument in square brackets when we
         defined the environment, Corollaries and Theorems
         share a common counter.
15   \end{corollary}

16
17   \begin{theorem}
18   \label{mytheoremagain}
19   Guess this proves our point about the counter.
20   \end{theorem}

21
22   \begin{lemma}
23   \label{mylemmaagain}
24   We hope you're convinced.
25   \end{lemma}

26
27   \begin{remark}
28   Because of that asterisk, \LaTeX~ does not maintain a
         counter for remarks. Also, we changed the style from
          plain(default) to remark.
29   \end{remark}

30
31   \begin{proof}
32   Behold the power of the \verb!amsthm! package; it
         provides a proof environment as well. Socrates is a
         man. All men are mortal. Therefore, Socrates is
         mortal.
33   \end{proof}

34
35   \vspace{1em}

36
37   \begin{proposition}
38   \label{myfirstproposition}
39   Because of the optional \verb!section! argument we
         supplied while defining the proposition environment,
          the counter resets every time there's a new section
          .
40   \end{proposition}
```

5   Custom environments

Proposition 5.1. *You can create environments like these, and make them more elaborate!*

Claim 1
Bertrand Russell

If $2 + 2 = 5$, then I am the Pope. You can prove any statement under the sun if you start with a false premise.

Claim 2
Gambler's Fallacy

If a fair coin shows heads 20 times in a row, it is more likely to show tails the next time.

Claim 3
Twin Prime Conjecture

There are infinitely many pairs of prime numbers that differ by exactly 2.

Claim 4
Default Colour

Observe the code. The colour that is passed to the environment is an optional argument. It's default value is `black`. **Optional arguments come first. They must be specified in square brackets.**

This is how the environment was defined in the preamble:

```
1  \usepackage{xcolor}
2  \newcounter{claim}
3  %if you want this counter to reset every section,
4  %append [section] as an optional argument:
5  %\newcounter{claim}[section]
6
```

```latex
%\newenvironment{<name of environment>}[<number of
    arguments>]
%{<code at the beginning>}
%{<ending code>}

%Of course, you can define unordered environments!
\newenvironment{goodclaim}[2][black]
{
    \color{#1!50!black}
    \refstepcounter{claim}
    %this increments the value of the counter by one

    \begin{center}
        \textbf{\large{Claim \theclaim}} \\
        %this prints the value of the counter
        \textbf{#2}
    \end{center}
}
{
    \vspace{1.5em}
}
%For ordered environments like these, which are
    equipped with a counter, you can make labels and
    refer to them later, just like theorems.
```

Thus, a custom environment called `goodclaim` is defined. It takes two arguments: first, an optional colour, in square brackets (if this is not specified, the default is black), and then, the compulsory title, in curly brackets. Here's an example.

```latex
\begin{goodclaim}[orange]{Twin Prime Conjecture}
\label{milliondollars}
There are infinitely many pairs of prime numbers that
    differ by exactly $2$.
\end{goodclaim}
```

If you'd like to redefine an existing environment, simply use `\renewenvironment` instead of `\newenvironment`.

6 Figures and Tables

Often, in a technical setting, you'll find the need to paste pictures, and to draw tables. LaTeX has environments to do just that.

6.1 Figures

For illustration purposes, we will use the very first image we showed you in the introductory tutorial.

```
1  \begin{figure}[H] %[H] is from the floatrow package, it
       forces the picture to be typeset RIGHT HERE.
2      \centering
3      %make sure your picture is in the same working
           directory/folder, or specify the graphics path
4      \includegraphics[width=0.6\linewidth]{ease-graph.
           png} %note the optional argument to adjust its
           size. Do it manually!
5      \caption{Simple.}
6      \label{fig:vanilla}
7  \end{figure}
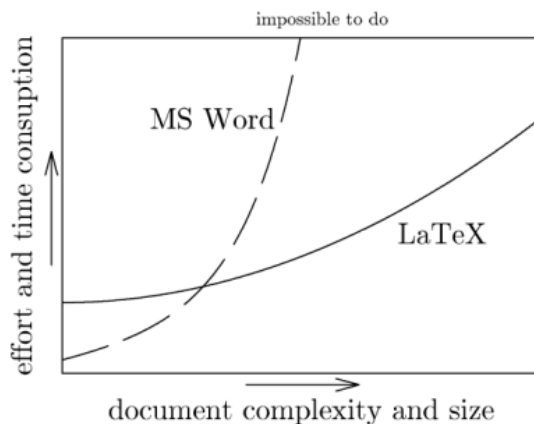```

And voila!



Figure 1: Simple.

You could use floatrow to insert multiple pictures together:

```
1  \begin{figure}[H]
2      \centering
3      \begin{floatrow}
4   %Figure boxes in a floatrow will be placed side by
           side.
5   %It's wise to keep a gap while specifying
           dimensions.
6           \ffigbox[0.4\textwidth]{\caption{A small graph
               .}}{\includegraphics[width=0.4\textwidth]{
               ease-graph.png}}
7           %Make sure the width of the picture doesn't
               exceed that of the box.
8           \ffigbox[0.4\textwidth]{\caption{The same graph
               .}}{\includegraphics[width=0.4\textwidth]{
               ease-graph.png}}
9       \end{floatrow}
10      \begin{floatrow}
11          \ffigbox[0.6\textwidth]{\caption{You need more
               pictures man.}}{\includegraphics[width=0.6\
               textwidth]{ease-graph.png}}
12      \end{floatrow}
13  \end{figure}
```

See below. The subfigure method is an alternative to floatrow; it may work on your machine, but sometimes it throws an inexplicable error about the environment not being defined.

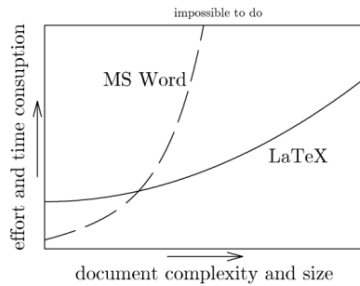For further reading, we encourage you to go through [9, Chapter 5]
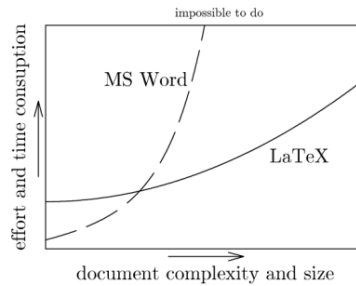
Figure 2: A small graph.
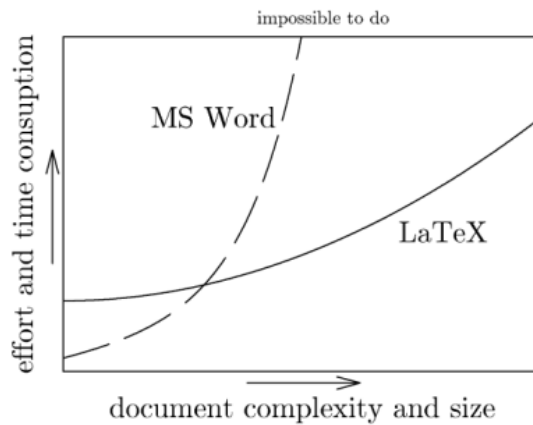

Figure 3: The same graph.

Figure 4: You need more pictures man.

6.2 Tables

The `table` environment encapsulates the LaTeX table. It takes care of the placement of the table in the document, caption, etc. Nested in the `table` environment is the `tabular` environment. The latter is where we enter the contents of our table.

Note that `\\` starts a new row, and the ampersand `&` acts as the column separator.

Here we build our first simple table.

```
1  \begin{table}[H]
2      \centering
3      \begin{tabular}{l|c|r} %note the alignment
```

```latex
 4            \hline
 5             \textbf{Left} & \textbf{Centre} & \textbf{
                 Right} \\
 6             \hline
 7            Liberal & Moderate & Conservative \\
 8            Blue & vs & Red \\
 9            Biden & vs & Trump \\
10             \hline
11        \end{tabular}
12        \caption{A Simple Table. USA votes.}
13        \label{tab:my_label}
14  \end{table}
```

Left	Centre	Right
Liberal	Moderate	Conservative
Blue	vs	Red
Biden	vs	Trump

Table 1: A Simple Table. USA votes.

Include the `multirow` and `multicol` packages in the preamble.

With the `\multirow{NUMBER_OF_ROWS}{WIDTH}{CONTENT}` command, we can have a cell span multiple rows. (Using * as the width lets LaTeX automatically determine it):

```latex
 1  \begin{table}[H]
 2      \centering
 3      \begin{tabular}{l|c|r} %note the alignment
 4          \hline
 5           \textbf{Left} & \textbf{Centre} & \textbf{
                 Right} \\
 6           \hline
 7          Liberal & Moderate & Conservative \\
 8           \hline
 9          Blue & \multirow{3}{*}{vs} & Red \\ %both rows
                 with vs are combined
10          Biden &  & Trump \\ %content of the second
                 column omitted
```

```
11            Democrats &  & Republicans \\
12            \hline
13      \end{tabular}
14      \caption{Multirow Table. USA votes.}
15      \label{multirow}
16 \end{table}
```

Left	Centre	Right
Liberal	Moderate	Conservative
Blue		Red
Biden	vs	Trump
Democrats		Republicans

Table 2: Multirow Table. USA votes.

With the `\multicolumn{NO_of_columns}{alignment}{content}` command, we can have a cell span several columns.

```
1  \begin{table}[H]
2      \centering
3      \begin{tabular}{|c|c|c|c|}
4      \hline
5            \textbf{Day} & Session 1 & Session 2 & Session
               3 \\
6            \hline
7            1 & \multicolumn{2}{c|}{Australia} & India \\
8            \hline
9            2 & \multicolumn{3}{c|}{Australia}\\
10           \hline
11           3 & \multicolumn{2}{c|}{India} & Australia \\
12           \hline
13           4 & \multicolumn{3}{c|}{India}\\
14           \hline
15           5 & \multicolumn{2}{c|}{Shared} & India \\
16           \hline
17      \end{tabular}
18      \caption{Who dominated which session: India vs
            Australia, Kolkata 2001}
```

```
19        \label{edengardens}
20    \end{table}
```

Day	Session 1	Session 2	Session 3
1	Australia	Australia	India
2	Australia	Australia	Australia
3	India	India	Australia
4	India	India	India
5	Shared	Shared	India

Table 3: Who dominated which session: India vs Australia, Kolkata 2001

We can also use `multirow` and `multicolumn` in tandem. Observe the code carefully to see how it's done.

a^2	ab	.
ab	b^2	.
.	.	.	.

Table 4: Middle School Algebra

```
1  \begin{table}[H]
2      \centering
3      \begin{tabular}{|c|c|c|c|}
4          \hline
5          \multicolumn{2}{|c|}{\multirow{2}{*}{$a^2$}} &
                  \multirow{2}{*}{$ab$} &   . \\
6          %first entry: 2*2 cell; second entry: 2*1 cell
7          \multicolumn{2}{|c|}{} &  & . \\
8          %the empty content is placeholder, content
                  will come from the row above
9          \hline
10         \multicolumn{2}{|c|}{$ab$} & $b^2$&.  \\
11         \hline
12         .  &. &. &. \\
13          \hline
```

22

```
14      \end{tabular}
15      \caption{Middle School Algebra}
16      \label{multirowcolumn}
17  \end{table}
```

We have introduced the basics. This tutorial [9, Chapter 9] is comprehensive. Read it to learn a few cool tricks you can do with LaTeX tables!

6.3   Floats

Figures and Tables in LaTeX are actually examples of Floats, containers for things in a document that cannot be broken over a page. LaTeX does not put the floats into wherever you write them in your code, but it decides the best place to put it. Sometimes your image or table can even end up in a different page than the content where you wrote the code to insert the table.

You can tell LaTeX where you would prefer it to put your float, by passing in a location specifier inside []. For example, if you notice our Table/Figure codes, you'll see something like `\begin{table}[H]`, here the H tells LaTeX that you want to place the float precisely at the postion in your LaTeX code. While this may always seem desirable, it does not always give desirable results. Here are the specifiers you can put in

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.

p Put on a special page for floats only.

! Override internal parameters LaTeX uses for determining "good" float positions. You would use it in conjunction with another specifier, like `\begin{table}[h!]`. This is kind of like forcing LaTeX to put it wherever you are telling it to

H As you know, places the float at precisely the location in the LaTeX code.

7 Highlighting Code

Programming now permeates in almost every field of research, and hence, code is a necessity in most technical reports. To get started, we recommend the easily accessible `listings` package. `minted` does an even more elaborate job, however, it requires some effort to get it running.

Set your preferences in the preamble like this:

```
1  \lstset{
2      numbers=left ,
3      %language = <set default language >,
4      breaklines=true , %automatic line breaks only at
           whitespace
5      keywordstyle=\color{blue}\bfseries ,
6      numberstyle=\tiny\color{gray},
7      commentstyle=\color{green!30!black},
8      stringstyle = \color{violet}
9  }
```

And then when you want to adorn your document with code, use the `lstlisting` environment as follows:

```
1  \begin{lstlisting}[language=<the programming language >]
2  <your code goes here!>
3  \end{lstlisting}
```

And then, it's magic.

7.1   Example: C++ code snippet

`[language=C++]`

```cpp
#include <iostream>
//preprocessor directive
using namespace std;

int main(int argc, char* argv[])
{
    string s = "Hello World";
    cout<<s<<endl;
}
```

7.2   Example: Python code snippet

`[language=python]`

```python
from numpy import *
#insanely powerful library.

print("First five odd numbers")
for x in range(10):
    if not x%2 == 0:
        print(x)
```

7.3   Example: Java code snippet

`[language=java]`

```java
//yet another Hello World
public class welcome
{
    public static void main(String[] args)
    {
        System.out.println("Welcome to the world of Java");
    }
}
```

8   Cross References and Citations

You first need to import the package in the preamble: `\usepackage{hyperref}`.
Now, at various points in the document, we've made labels. More specifically,
if you want to make a reference to an instance of an **ordered environment
or section**, like how we remind you of equation 1 or Theorem 1, simply type
`\ref{<label of the point>}`. For instance `\ref{mycorollary}` makes a
reference to Corollary 2. We can also refer to Claim 3, a custom claim.

For reasons related to the nature of the typesetting algorithm, that are be-
yond the scope of this tutorial, you may need to compile your TEX file twice
for the cross references to work.

You've probably noticed that there were a few hyperlinks in this document.
It's pretty simple.
`\href{https://www.youtube.com/watch?v=MKk1u5RMTn4}{An awesome song}`
typesets: An awesome song. It's neat.

Sometimes, you care to show the URL itself.
In such cases, use `\url{www.google.co.in}`. Makes sense in a context like,
"This search engine gives all the answers. Check it out: `www.google.co.in`".
Make sure to `\usepackage{url}` in the preamble.

For formal citations, we use BibTeX. Here's a sample entry from the .bib
file

```
@misc{greek,
title = {The Greek Alphabet},
url = {https://web.mit.edu/jmorzins/www/greek-alphabet.html},
publisher = {MIT},
author = {Jacob Morzinski}
}
```

Here is a comprehensive guide to BibTeX citations. [6]. We strongly rec-
ommend you read it yourself, and refer to it whenever you write technical
reports.

If our main source code is `foo.tex`, the bib file must be named `foo.bib`

The word `greek` is the key. In order to cite this source, we type in :
`\cite[<optional info>]{greek}` in our TEX source code.

In order to make the url show in the citations, we must load the following in our preamble:

```
1  \usepackage{hyperref}
2  \usepackage{url}
```

To make a References section that supports URLs,

```
1  \bibliographystyle{plainurl}
2  \bibliography{foo} %do not write the .bib extension
```

In order for citations to work as intended, we need the following compilation order

- Compile the tex file
- Compile the bib file
- Compile the tex file
- Compile the tex file (yes, again!)

If you're using the VSCode plugin LaTeX workshop, it does this automatically for you

Remark. When you write technical articles with the intention of publishing them in reputed journals or conferences, your citations are well recognised. It shouldn't be hard to look them up and find a BibTeX entry online.

9   Conclusion

With this, you're equipped with quite a few commands to typeset a mathematical report yourself. We've tried to make the examples as diverse as possible, drawing from our own experience with LaTeX. Although this tutorial is far from exhaustive, we've tried our best to provide resources, tips and tricks that frequently prove themselves useful. Good luck! :)

References

[1] OEIS community. List of latex mathematical symbols. URL: `https://oeis.org/wiki/List_of_LaTeX_mathematical_symbols`.

[2] TeX FAQ. Non-letters in macro names. URL: `https://texfaq.org/FAQ-linmacnames`.

[3] Jacob Morzinski. The greek alphabet. URL: `https://web.mit.edu/jmorzins/www/greek-alphabet.html`.

[4] Overleaf. Matrices. URL: `https://www.overleaf.com/learn/latex/Matrices`.

[5] American Mathematical Society. Using the `amsthm` package. Version 2.20.3, September 2017. URL: `http://www.ams.org/arc/tex/amscls/amsthdoc.pdf`.

[6] RSI 2012 Staff. Bibtex templates, 2012. URL: `http://web.mit.edu/rsi/www/pdfs/bibtex-format.pdf`.

[7] Community Thread. eqnarray vs align, 2010. URL: `https://tex.stackexchange.com/questions/196/eqnarray-vs-align`.

[8] Community Thread. What does each ams package do?, 2011. URL: `https://tex.stackexchange.com/questions/32100/what-does-each-ams-package-do`.

[9] Claudio Vellage. A simple guide to latex - step by step. URL: `https://www.latex-tutorial.com/tutorials/`.