

Mathematical Report Typesetting

Prakhar Mittal

July 14, 2020

Contents

1	Equations and Formulae	2
1.1	Math Mode	2
1.2	Inline Math	2
1.3	The Equation Environment	3
1.4	Matrices	4
1.5	Macros: Custom Commands	5
2	Theorems	6
3	Custom Environments	7
4	Figures and Tables	8
4.1	Figures	8

1 Equations and Formulae

1.1 Math Mode

To type maths in \LaTeX , one needs to switch to math mode, a special mode in \LaTeX . There are 2 types of modes:

Inline Math The typesetted math sits in the same line as the running text and is not typesetted in a math environment, generally used for simple statements such as $a = b$ to be embedded inside a line.

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

Generally we don't write text in math environment as it would be rendered *likethis* without any spacing

1.2 Inline Math

For typesetting mathematical expressions in the midst of your text, place the expression between two $\$$ signs. For example :

If a function f is continuous, then for every $\varepsilon > 0$ there exists a δ such that $\|\mathbf{x} - \mathbf{x}_0\| < \delta$ implies $\|f(\mathbf{x}) - f(\mathbf{x}_0)\| < \varepsilon$.¹

Other ways to typeset inline math is to wrap the expression in:

- $\backslash[$ 'expression here without the quotes' $\backslash]$
- $\mathbb{\$}$ 'expression here without the quotes' $\mathbb{\$}$

For small greek letters like θ we use $\backslash\theta$. To capitalize it (Θ) use $\backslash\Theta$.

To bold in math we use $\backslash\mathbf{}$. However, this does not work for greek symbols. For greek symbols, we use $\backslash\boldsymbol{\symbol}$. For example: $\boldsymbol{\omega}$

For using subscripts/superscripts, we use underscore/carat sign before the text in subscript/superscript. If the text does not include any spaces, we can directly use them, eg. x_0^1 , however, if the text does include spaces, then we have to wrap it in curly braces, eg. $p_{ij}^{\alpha+\beta}$.

Now consider $\$t \in \mathbb{R}\$, this will typeset as$

$$t \in \mathbb{R}$$

See how many other math stuff can be easily typesetted?

In some cases, we wish to use symbols that otherwise hold a meaning in syntax handling such as $\$$ or $\{\}$. To escape their special meaning and typeset them as it is, we use backslash followed by the symbols (in most cases), like $\backslash\$$.

¹ $\backslash\varepsilon \rightarrow \varepsilon$ whereas $\backslashepsilon \rightarrow \epsilon$

Tilde creates a space between symbols in inline math mode. For eg.

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

(Notice the distance between vertical line and x)

There are several symbols which can be typesetted using L^AT_EX (including this one). A comprehensive list can be found [here](#).

Some other math expressions which are commonly used are :

Combinations $\binom{n}{k}$ also written as $\frac{n!}{(n-k)!}$

Continued Product $n! = \prod_{i=1}^n i$

Integration An extension of factorial function is the Gamma function. For positive integers n ,

$$(n-1)! = \Gamma(n) = \int_0^\infty x^{n-1} e^{-x} dx$$

Continued Sum It follows from the Binomial Theorem that $\sum_{k=1}^n \binom{n}{k} = 2^n$.

Root The Binomial Theorem is incredibly powerful and can be used to approximate $\sqrt{1+x}$, or even $\sqrt[n]{1+x}$.

Sometimes we want regular text while writing in math mode, for that we use `\text`. For eg.

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

1.3 The Equation Environment

Equations are often pivotal to a report, and they need to stand out. This is where the equation environment comes into play. Here's an example:

The Ideal Gas Law states that :

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

Notice the label (in the source code :P). This helps to refer the equation in later sections. Hence, it is a good practice to label the equations.

Sometimes when you are absolutely sure that an equation would not come into use later (like an intermediate step of a derivation), then you would like to do away with the side numbering. This can be done in two ways, 1. using modified equation environment (`equation*`), or by using `\nonumber`.

Eg:

Vanderwall Proposed the real gas equation as a more accurate model.

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

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

$$pV = ZnRT$$

Many a times, it is required to write several equations one after the other. In such cases, it looks nice if they are properly aligned. It is strongly recommended to use the `align` environment for this, which is a part of the `amsmath` package. Do not use `eqnarray`, which is simply base L^AT_EX.

Use `&` right before the symbol you want alignment about. Use double backslash (`\\`) for changing line. Example:

$$\begin{aligned}\nabla \cdot \mathbf{E} &= \frac{\rho}{\varepsilon_0} \\ \nabla \cdot \mathbf{B} &= 0 \\ \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\ \nabla \times \mathbf{B} &= \mu_0 \mathbf{j} + \frac{1}{c^2} \frac{\partial \mathbf{B}}{\partial t}\end{aligned}$$

See how the 4 equations are neatly aligned such that all the ‘=’ sign lie on the same line.

1.4 Matrices

Typesetting matrices is straightforward, very similar to tables. Here’s an example:

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 **iff** $a_0 \neq 0$.

There are various styles of matrices that the `amsmath` package allows you to typeset.

²`\left(\right)` ensure that the brackets are appropriately sized

1.5 Macros: Custom Commands

Many a times we need a certain typing style used again and again. If it requires multiple commands, then this work becomes quite in-efficient. Luckily, you can define custom commands in \LaTeX for your ease. Custom commands are known as macros in \LaTeX lingo. Macros can take multiple arguments.

Suppose you are writing a report based on **Linear Algebra**, which heavily uses the notation of inner product. Obviously, an expression like $\langle \mathbf{u}, \mathbf{v} \rangle$ is gonna come up left and right. If you were to hardcode it everytime, it would be:

```
1 $\langle \text{\textcolor{blue}{langle}} \text{\texttt{\textbf{u,v}}} \text{\textcolor{blue}{rangle}}$  
2 $\langle \text{\textcolor{blue}{langle}} \text{\texttt{\boldsymbol{Theta}}} \text{\textcolor{blue}{rangle}} \text{\texttt{\boldsymbol{zeta}}} \text{\textcolor{blue}{rangle}}$
```

ad nauseum.

What you could do instead is write a macro which takes in 2 arguments and returns them in this format. Macros are defined in the preamble.

```
1 % In the preamble  
2 \newcommand{\innerproduct}[2]{\text{\textcolor{blue}{langle}} \text{\texttt{\boldsymbol{#1}}} \text{\textcolor{blue}{rangle}} \text{\texttt{\boldsymbol{#2}}} \text{\textcolor{blue}{rangle}}}
```

The new command will be invoked by using `\innerproduct`. The [2] shows that it will take in 2 arguments. Specifying this is optional, if not specified, the custom command takes in no arguments. #1 and #2 are the placeholder for each of the arguments. Our command can be invoked as follows:

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

Which will typeset into $\langle \mathbf{u} + \mathbf{w}, \mathbf{v} \rangle$, hence saving us from excessive labour.

Suppose if you're writing a report on an extremely efficient algorithm, and have to use the expression $\mathcal{O}(1)$ again and again. If you wish to be a bit lazy, you can just define a macro and save yourself from typing a few extra characters every time.

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

should do the job. Note that the macro names can only contain alphabetic characters.

If you wish to make a custom command having the same name as an already existing command from the packages you have imported, it is a redefinition. To do so, the same syntax is followed, except the `\newcommand` is replaced by `\renewcommand`.

2 Theorems

Theorems, lemmas, propositions, proof, remarks, and corollaries, any mathematical report feels empty without these. The \LaTeX package `amsmath` provides us with separate environments to easily typeset them and be able to showcase all these easily. Recall, in the preamble, we imported `amsmath` package. The following lines 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}[theorem]{Proposition}
5
6 \theoremstyle{remark}
7 \newtheorem*{remark}{Remark}
```

The `\newtheorem` takes in 2 arguments. The first argument states the environment that is defined, and the second one is the word that will be printed, in boldface font, at the beginning of the environment. Once this new environment is defined it can be used normally within the document, delimited it with the marks `\begin{theorem}` and `\end{theorem}`.³

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. Since we provided an optional argument [theorem] when defining the environment, Corollaries and Theorems share the same counter.*

Theorem 3. *This proves the point about the counter.*

Lemma 2. *Hope you're satisfied by now.*

Remark. Because of the asterisk, \LaTeX does not use a counter for the remarks. Also, we changed the style of Remarks from plain(default boldface un-italicized) to remark(italicized). `\theoremstyle` changed the way the starting word shows.

Proof. Behold the power of `amsmath` package; it provides a separate environment for proofs as well. Notice the square at the end of the proof as well, this is another feature of the proof environment.³ \square

Proposition 2.1. *Because of the optional [section] argument we provided in the last, the proposition counter resets after every new section. Each proposition would be numbered as [sectionnumber][propositionnumber].*

³Refer Overleaf tutorial

3 Custom Environments

L^AT_EX also gives us the ability to define new environments. The definition of the new environment is given in the preamble. Here is an example creating an environment called “claim”:

```
1 \usepackage{xcolor} % Just for the colors (not necessary for
   environment creation)
2 \newcounter{claim} % To create a new counter for numbering the
   environments (like theorem)
3 %If you want the counter to reset after every section, add a [
   section] argument after \newcolor{claim}
4 %\newcounter{claim}[section]
5 %\newenvironment{nameofenvironment}[no. of arguments]
6 %<code at the beginning>
7 %<ending code>
8
9 \newenvironment{claim}[2][black] %The [black] makes the first
   argument optional with a default value of black
10 % The second argument remains required
11
12 { % This cell executes when the \begin{env.} is used
13   \color{#1!50!black}
14   \refstepcounter{claim} %increments the value of counter by 1
15
16   \begin{center}
17     \textbf{\large{Claim \theclaim}} \\ % prints value of
       counter
18     \textbf{#2}
19   \end{center}
20 }
21
22 { % This cell executes when the \end{env.} is used
23   \vspace{1.5em}
24 }
25 % For ordered environments like these, you can use labels for
   later referencing them.
```

The new environment would look like this:

Claim 1
First Custom Environment

Hello There

Claim 2

Optional and Necessary Arguments

The optional argument is wrapped in `[]` and the necessary arguments in `{}`.

Claim 3

Default Color

The color specifications are to be given in `[]` brackets, if none is given, then the environment assumes the default colour, specified in it's definition.

If you'd like to redesign an existing environment, use `\renewenvironment` in place of `\newenvironment`.

4 Figures and Tables

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

4.1 Figures

The `floatrow` package will come into use here.

```
1
2 \begin{figure}[H] % [H] s from floatrow package which forces the
   picture to be typeset RIGHT HERE.
3   \centering
4   % make sure your picture is in the same working directory ,
   or else , provide a relative path to it
5   \includegraphics[width=0.6\linewidth]{<imagename.extension
   here>} %note the optional argument provided to adjust the
   size of the image.
6   \caption{Anything} % This would display below the picture
   like ‘‘Figure 1: <caption>’’
7   \label{fig:img}
8 \end{figure}
```


And, voila!

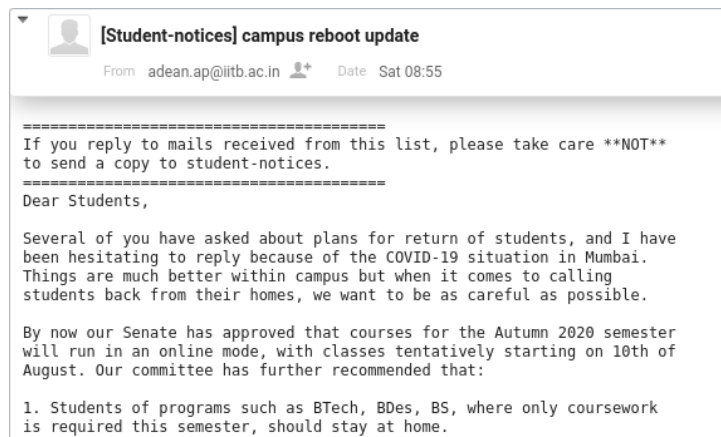


Figure 1: Semester will be online, starting 10th August

You can use `floatrow` for inserting multiple images together.