LaTeX: A BRIEF INTRO

VIPUL NAIK

Abstract. This material is a basis for an introductory talk to LaTeX. I focus, not so much on the technical details and nitty-gritty, as on the entire experience of creating a complete document efficiently using LaTeX.

1. Why use LaTeX?

1.1. **Why write at all?** A valid question. Why bother to write mathematics/computer science/physics? Here are the two reasons:

- The *attempt to put one's thoughts into writing* forces a person to clarify those thoughts. Many of the shortcuts and bluffing tricks we use while speaking or explaining things on the board simply don't work in writing. For instance, in lectures, you'll hear people say "left action, or right action, or whatever" or "under suitable assumptions". When you put things in writing, sloppiness of ideas and expression is far more visible.
- To write down means to record for posterity. When you have written down the mathematics work, you can share it not just with your near and dear ones, but also with people on distant lands.

1.2. **Why write on a machine?** What is the use of writing on a machine when there is paper? After using a machine for a few years, you'll possibly ask the question the other way around: why bother to use paper when you have a machine?

Some reasons to use a machine:

- The final output looks much neater. Specially true if you are using LaTeX, but also true if you are using a word processor such as Microsoft Word or OpenOffice or whatever.
- Stuff written on the machine is *much, much easier* to edit. You can move around stuff, copy and paste, etc.

  Imagine this: Instead of using a computer with cut-and-paste and insert features, each time you needed to make an insert, you'd put in a (what do they call it?) and add some words on top, throw in some words in the margin, mark portions with blue pencil and so on. Who would want to do such editing? And then, you'd have to rewrite the whole stuff to present it to others.
- There is more natural organization in the material. It allows you to divide the material into chapters, sections, subsections, and so on.
- You can share the material with friends, not just those physically near you, but also those living in distant areas.

  Imagine if: Each time you wanted to send a mathematical write-up to a friend, you would have to write it all out again, or photocopy it, and then send it via post, and wait for your friend's reply letter.

1.3. **Why LaTeX?** Let me begin by saying that many, practically all of the things you'd want to do with LaTeX as a beginner *can* be done on Microsoft Word or Openoffice with reasonably satisfactory results. However, every document preparation system has its own *niche*, and structured mathematical documents is the niche of LaTeX. If you want to create a five line text document, a text editor is better than LaTeX. If you want to create a longer mathematical document, go in for LaTeX.

Specific advantages of LaTeX:

- The ease with which *mathematical symbols* can be used: Symbols like ∂, ∇, the hats and tildes, arrows above, below, on the left and on the right, are all easy to make with LaTeX. This is what attracts people to LaTeX to begin with.

- The ease with which *entire mathematical expressions* and derivations can be organized and displayed. For instance, manipulating one equation for a few lines to obtain another.
- The ease with which the *usual constructs of structured mathematical documents* can be accommodated. Apart from the division into chapter, section and subsection (which is common with other type of texts), the use of list environments (which is again common with other types of texts), there are the *theorem environments* and *proof environments* that allow us to indulge in the usual mathematical style of claims, lemma, corollaries and proofs.
- The ability to *define macros* both for shortcuts and for modularity.
- The pretty appearance of the final text.

Most of these functionalities *can* be extracted from word processors, but LaTeX allows it *much more easily*. So easily that I hope that after this one talk, you'll know how to do it.

2. What this talk is about

2.1. **Other resources.** Before launching off, I will list some "standard" introductions to LaTeX. Of course, you're likely to hit on many more through a search engine (type "introduction to LaTeX" in the search bar).

Online resources:

- "The Not-so-short introduction to LaTeX2e" is a rather famous (shall we say notorious?) introduction. Here's the link to one of the places you can find it:
  `http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf`
- An introduction to TeX and LaTeX by the Harvard Mathematics Department:
  `http://www.math.harvard.edu/texman`
- The Wikibooks book on LaTeX:
  `http://en.wikibooks.org/wiki/LaTeX`

Book resources:

- "LaTeX– users guide and reference manual" by Leslie Lamport, the guy who wrote LaTeX.
- "The LaTeX companion" by Goossens, Mittelbach, Samarin
- "LaTeX for everyone" by Jane Hahn

Important sites related to LaTeX are here:

- The **Comprehensive TeX Archive Network**: This has material to help people get started on LaTeX, and also has a number of extra packages that can be downloaded to make life easier for future TeXers. The webpage is `http://www.ctan.org`
- The **LaTeX Project page** contains general information about the LaTeX project. The webpage is `http://www.latex-project.org`
- The **TeX Users Group** page: The webpage is `http://www.tug.org`
- The MikTeX page. This has downloadable version of LaTeX for Windows (a 25 MB version as well as the more elaborate version). The webpage is `http://www.miktex.org`

2.2. **What I focus on in this talk.** In this talk, I want to give and tell something that is *not* easily available from a command list or quick reference. I want to give information (biased by my own opinions) that will help you create great documents quickly using LaTeX. I want that, at the end of this talk, people are able to convert thoughts *directly* into LaTeX files, rather than take the circuitous route of first writing them on paper or as plain text and then TeXing it out.

Some things I'll talk about:

- The overall *look and feel* of a LaTeX file. The way it works. The way to write it at first shot.
- More on mathematical symbols
- Global structural elements: chapters, sections, subsections
- Local structural elements: list environments, theorem environments, mathematical environments, tabular environments
- Indexing, cross references, and footnotes
- General points about mathematical writing
- The use of style files, commands, and macros

2.3. **The golden rule.** The golden rule of LaTeX use, that I'll reiterate in different *avatar*s, is: *use the logically correct and appropriate symbols* even if alternative symbols have similar appearance. Remember, it is not just the final output file that matters, the source LaTeX should also be easy to read and edit. Some examples:

- Use the proper symbols for summation and product rather than the mimicking Greek letters. Use the *name derived from the nature of the operation* wherever possible, rather than the *name derived from the physical appearance*.
- Tell LaTeX the title, author etc. of the document instead of using your own fonts to display the title and author.
- Use LaTeX sections and subsections rather than typing headings and subheadings in your own arbitrary format.
- Use standard LaTeX list environments.

2.4. **Typographical conventions.** In the coming sections, I shall follow some conventions:

- While naming the LaTeX command in a text description, I shall use a `typewriter font`. Also, I will *not* put the preceding backslash in the text description of the command. However, to run the command effectively, please do put the backslash before the command name, and put the parameters in curly or square braces, depending on whether or not they are optional.
- In some cases, I show both the original LaTeX source and the compiled text. The original LaTeX source is printed using the verbatim mode.

3. The look and feel of a LaTeX file

3.1. **Preparing documents using LaTeX.** These are the typical steps in document preparation using LaTeX:

(1) Type out the LaTeX source in a conducive editor. Typical editors are `emacs` (both Windows and Linux), `vi` (Linux), and `winedt` (Windows). There are other preparation tools as well, specialized for LaTeX. A popular tool is `kile`.

(2) Compile the LaTeX source, either from within the editor, or from the command prompt. Depending on what you want to convert the LaTeX file to, you can choose the appropriate compiler.

Final type	Command
DVI	`latex`
PDF	`pdflatex`
RTF	`latex2rtf`
HTML	`latex2html`
XML (addon)	`ttm`

Each time you invoke the command, it does *one pass* of compilation, and produces the corresponding output file. In the first pass, table of contents, index, footnotes etc. may not all be put, because these require information available only in the second pass. The information needed is stored in other files: `.toc` for the table of contents, `.idx` for the index entries, `.aux` for auxiliaries, `.log` for the compilation log.

If compilation gives an error, figure out the error and quit (type X to quit). Correct in the original document and recompile. If compilation says something about "rerun to get references right" then do so.

(3) To make the index, you need to run the command `makeindex` on the `.idx` file. This sorts the entries and produces a `.ind` file. When you next compile the original document, the `.ind` file will get included at places where you have asked for the index to be printed. Note that you may *again need to recompile* for the table of contents to contain the correct link to the index.

(4) After you have compiled enough times and got a stable document, view it. See if the output document is what you had hoped for.

3.2. **What goes on top.** Here's how a typical LaTeX file looks:

```
\documentclass[12pt]{article}
\usepackage{fullpage,hyperref}

\input{/ag/vipul/math/ownworks/seriousart.sty}

\title{How to win friends and influence people: a review}
\author{Vipul Naik}

\makeindex
```

```
\begin{document}
\maketitle
\begin{abstract}
  I am ....
\end{abstract}

\section{}

....

\printindex
\end{document}
```

Here's the analysis line by line.

(1) The first line describes the *type* or *class* of the document. The standard document classes are: `article, book, slides, report, letter`. The American Mathematical Society has brought out variants of each of these, and we thus have: `amsart` (for `article`), `amsbook` (for `book`), and `amsreport` (for `report`).

The document classes differ in global structural elements: `article` has only `section, subsection` and `subsubsection`, whereas `book` has `part, chapter, section, subsection`.

Look carefully at the line:

```
\documentclass[12pt]{article}
```

The first word (documentclass) is the *command name* and the thing in curly braces (article) is the *parameter*. The things in square braces (12pt) are *optional parameters*. This is a general LaTeX rule: command begins with a back slash, compulsory arguments in curly braces, optional arguments in square braces (if any). The optional arguments in this case give further information to the paragraph-making and page-making programs so that they can produce the correct effect.

(2) The second line gives a list of *packages* to use in the document. LaTeX has a diverse collection of packages, each specializing for a different purpose. The `fullpage` package fits in more content into the page (by reducing margin sizes) while the `hyperref` package makes the final PDF file with clickable links. The kind of package you want to use depends on the kind of document you want to finally produce.

(3) The third line is my own style file– essentially a list of macros, commands and other conveniences I frequently use throughout my LaTeX writing.

(4) The fourth line tells the LaTeX processor the *title* of the document. Note that this is *not equivalent* to actually printing the title on screen. Rather, it is like abstractly *informing* LaTeX about the title.

LaTeX can do a lot with the title apart from printing it on top: it uses the title for headers, footers and other things depending on the document class and other options.

(5) The fifth line tells the LaTeX processor the name of the *author* of the document. Again, this is *not equivalent* to printing the author on screen. As with the title, LaTeX uses the author for headers, footers and other purposes.

(6) The line saying `makeindex` essentially tells LaTeX to be willing to index items as and when indicated.

(7) The line `\begin{document}` is the official indication that the document has begun. It is only *after this line* that one can start typing plain text and expect it to appear in the final document. From `begin{document}` to `end{document}` is the content of the document.

(8) The `maketitle` command tells LaTeX to print the title (along with author and other relevant information). The way it prints depends on the document class, but that can be changed using various options.

(9) The `abstract` is something where you briefly describe the purpose of the document. It isn't necessary to write an abstract, but it helps people decide whether to read what's below, and it helps you remember what you wrote!

(10) The document then begins, with sections and subsections and so on.

3.3. **Basic environment rules.**

3.3.1. *Outside the "document".* There are six basic modes in LaTeX (don't ask me to list them right now). One of them, incidentally, is the mode *before* the `begin{document}` instruction. The commands given here are *general information about the document* such as: what style files to include and use, what new commands and macros are being defined, what is the title, author, whether or not to make the index and so on. Things given there are *not directly printed on the final document.*

3.3.2. *Text mode within the document.* Text mode within a document is almost like ordinary text – it goes word for word into the final document. However, there are some rules:

- Multiple spaces are treated as a single space. Thus the sentences: "I am Vipul" and "I am Vipul" are treated equivalently.
- A single carriage return (or enter) is also treated as a single space. This allows lot of flexibility in the LaTeX file (you can keep short lines) which does not affect the final look of the document.
- Two consecutive carriage returns indicate to LaTeX the beginning of a new paragraph. However, LaTeX does not itself leave a new line for a new paragraph. Use double backslash to force a new line in the final document, or to force an empty line.

3.3.3. *Display environments within text.* These function largely like the text mode itself, except that the indentation and spacing may be a little different.

- The `quote` environment.
- The `verse` environment.
- The `center` environment causes whatever is inside to get center-aligned.

3.3.4. *Math mode within text.* For mathematical symbols within plain text, enclose the mathematical text between $ signs. Remember that *mathematical text* doesn't just mean complicated mathematical symbols. Even if using an English letter for a variable, enclose the English letter between $ signs so that it gets the right appearance. For instance, the text:

```
Let me explain the use of elementary algebra in word problems.
Suppose we are told that a fruit basket contains twice as many apples
as oranges, and that the number of bananas
is one more than the number of apples.
If the total number of apples,
oranges and bananas is $26$, then
what is the number of apples?

Here's how the solution runs: let $x$ be the number of oranges. Then, from
the given information, the number of apples is $2x$, and the number of
bananas
is $2x + 1$. Hence the total number of fruits is
$x + (2x) + (2x + 1)$
making it $5x + 1$. We then have the equation $5x + 1 = 26$, giving
the solution $x = 5$. Because the number of apples is $2x$, we obtain
that
there are $10$ apples.
```

This is how the same text appears when compiled:

Let me explain the use of elementary algebra in word problems. Suppose we are told that a fruit basket contains twice as many apples as oranges, and that the number of bananas is one more than the number of apples. If the total number of apples, oranges and bananas is 26, then what is the number of apples?

Here's how the solution runs: let x be the number of oranges. Then, from the given information, the number of apples is $2x$, and the number of bananas is $2x + 1$. Hence the total number of fruits is $x + (2x) + (2x + 1)$ making it $5x + 1$. We then have the equation $5x + 1 = 26$, giving the solution $x = 5$. Because the number of apples is $2x$, we obtain that there are 10 apples.

3.3.5. *Display math mode.* In display math mode, whole lines are devoted to displaying mathematical text. The symbols are thus wider, better spaced out, and are prettier to look at on the whole. Be careful about distinguishing between inline math and display math.

A formula that looks congested in inline math: $\binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1}$ looks nice in display math:

$$\binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1}$$

Display math can be activated in any of these ways:

- Simply use a double dollar to open and a double dollar to close.
- Use an equation environment or equation array environment. Equation array environments are used for listing one or more equations, aligned at the equality symbols. Moreover, these equations automatically get numbered. More on this later.

Just as there is inline math in text mode, there is inline text in display math mode using the `text` command. For instance:

```
\begin{equation*}
\overline{F} = m\overline{a} \text{ (Newton's second law)}
\end{equation*}
```

appears as:

$$\overline{F} = m\overline{a} \text{ (Newton's second law)}$$

3.3.6. *Verbatim mode.* Verbatim mode is one of the least LaTeX like: the text appears on screen *exactly as you type it.* No need to hit double enters for a single enter. Spaces are reproduced faithfully on screen. Constant width font is used. Typically, verbatim mode is used to reproduce pieces of program code.

3.4. **Font options for text.** There are six font options and some variants:

- Roman or `rm`. To print Roman font, use \rm. This is anyway the default font.
- Italic or `it`, Slant or `sl`, and Emphasis or `em`: They have almost the same effect. An example of the use of emphasis:

  ```
  There is practically {\em not even one} shop open in the street on Sunday.
  ```

  Some people use `emph` instead of `em`. `emph` works by *taking in a parameter*, so the above becomes:

  ```
  There is practically \emph{not even one} shop open in the street on Sunday.
  ```

  This appears as:
  There is practically *not even one* shop open in the street on Sunday.
- Bold face or `bf`: Gives a thick bold face appearance. For instance:

  ```
  A {\bf charge} is a unit of electrical accumulation.
  ```

  Some people use `textbf` instead of `bf`. `textbf` works by *taking in a parameter*, so the above becomes:

  ```
  A \textbf{charge} is a unit of electrical accumulation.
  ```

  Both of them appear as:
  A **charge** is a unit of electrical accumulation.
- Sans serif or sf, typewriter or tt (gives Courier New) and Small Caps. My name in each of them: Vipul Naik, `Vipul Naik` and Vipul Naik.

3.5. **Comments.** Comments in a LaTeX file begin with a % symbol and proceed right till the next carriage return. There is no support for nested comments.

4. More on mathematical symbols

4.1. **Sources for complete lists of symbols.** CTAN has a comprehensive listing:

`http://www.ctan.org/tex-archive/info/symbols/comprehensive/`

4.2. **Augmentations.** Important points about subscripting and superscripting:

- **Subscript**: To put stuff in the subscript, use the underscore or _ followed by the content in the subscript. If the content is only 1 letter, or a single command, there is no need to put curly braces. Otherwise, curly braces are necessary.

  It is not possible to have two subscripts one after the other in LaTeX. However, we can have *nested subscripting* – one subscript within the other.
- **Superscript**: To put stuff in the superscript, use the carat symbol (the one above 6 on QWERTY keyboards).
- **Head clothing**: To put symbols on top, we specify the top symbol, and provide to it as parameter the thing on top of which it is being put. Important commands:

Command	Parameter	Result
`hat`	x_i	$\hat{x_i}$
tilde	m	$\tilde{m}$
overline	α	$\overline{\alpha}$

- **Foot clothing**:

Command	Parameter	Result
`underline`	f	$\underline{f}$
underbrace	$abcd$	$\underbrace{abcd}$

4.3. **Typical symbols used in formulae.** Symbols used in formulae include:

- **Letters to denote variables**: When using English letters, always enclose within math mode as illustrated in section 3.3.4. When using Greek letters, prefix the Greek letter with a backslash. All small Greek letters work, but for capital Greek letters, commands exist only for those capital Greek letters that differ from their English equivalents.

  For some small Greek letters, there are variants with slightly different appearance. Namely, epsilon looks like ϵ while varepsilon looks like ε. Similarly, sigma looks like σ while varsigma looks like ς.
- **Comparison and connection symbols**: List of important ones:

Symbol	meaning	appearance
`sim`	similar	$\sim$
simeq	similar or equal	$\simeq$
approx	approximately	$\approx$
le, leq	less than or equal	$\leq$
ge, geq	greater than or equal	$\geq$
equiv	equivalent, congruent	$\equiv$
cong	congruent, isomorphic	$\cong$
ne	not equal to	$\neq$
implies	implies	$\implies$
iff	if and only if	$\iff$
to	to	$\to$
mapsto	maps to	$\mapsto$
models	models (logic)	$\models$
parallel	parallel	$\parallel$
perp	perpendicular	$\perp$
propto	proportional to	$\propto$

- **Quantifier symbols**: Typically, LaTeX leaves little space before and after the quantifier symbols, so you may want to force spaces.

Symbol name	appearance
`exists`	$\exists$
forall	$\forall$

- **Operations with parameters**: The parameters need to be provided in curly braces.

Symbol	Parameters	appearance
`frac`	*num*, *den*	$\frac{num}{den}$
sqrt	*num*	$\sqrt{num}$
binom	*up*, *down*	$\binom{up}{down}$

- **Inline binary operators**:

Symbol	meaning	appearance
`times`	product	$\times$
oplus	direct sum	$\oplus$
odot	?	$\odot$
otimes	tensor product	$\otimes$
cup	union	$\cup$
cap	intersection	$\cap$
sqcup	disjoint union	$\sqcup$
wedge	wedge product	$\wedge$
vee	?	$\vee$
lor	logical or	$\vee$
land	logical and	$\wedge$
in	member	$\in$
ni	contains	$\ni$
notin	not a member of	$\notin$
subset	proper subset of	$\subset$
subseteq	subset (or equal)	$\subseteq$

- **Arrows of different kinds**: Remember that when the *logically correct* command exists, do not use the arrow symbol. Thus, use `to` instead of `rightarrow` when you mean to say `to`. Use the implies sign rather than a double right arrow. Remember, the LaTeX version of the text should be humanly readable and editable.

symbol	appearance
`leftarrow`	$\leftarrow$
rightarrow	$\rightarrow$
longleftarrow	$\longleftarrow$
longrightarrow	$\longrightarrow$
Leftarrow	$\Leftarrow$
Rightarrow	$\Rightarrow$
nearrow	$\nearrow$
searrow	$\searrow$
nwarrow	$\nwarrow$
swarrow	$\swarrow$

- **Summation types**: For each operator, there is a subscript, a superscript, and what appears after the operator. The syntax is: operator name (underscore symbol) subscript (carat symbol) superscript followed by whatever comes after it. Also note that these types appear better in display math mode than in the inline math mode used in this table.

  Remember the *golden rule for summations*: Always use the sum command, and *not* Sigma, for writing down summations. Always use the prod command, and not Pi, for writing products. While speaking as well, say "summation" and *not* "Sigma".

Symbol	Parameters	appearance
`sum`	$i = 1$, n, $f(i)$	$\sum_{i=1}^{n} f(i)$
prod	$i \in S$, blank and i^2	$\prod_{i \in S} i^2$
int	0, 1, $f(x)\,dx$	$\int_0^1 f(x)\,dx$
max	$i \in [0,1]$, blank and $i^2 - i + 3$	$\max_{i \in [0,1]} i^2 - i + 3$
min	$i \in [0,1]$, blank and $i^2 - i + 3$	$\max_{i \in [0,1]} i^2 - i + 3$
bigcup		$\bigcup$
bigcap		$\bigcap$
bigwedge		$\bigwedge$
bigvee		$\bigvee$
bigsqcup		$\bigsqcup$
bigoplus		$\bigoplus$

- **Trigonometric and logarithmic functions**: These appear in straight font even within math mode. Important symbols include: `cos, sin, exp, cot, tan, log, sinh, tanh`, and so on.
- **Miscellaneous symbols**:

Symbol	meaning	appearance
`infty`	infinity	∞
partial	partial derivative	∂
angle	angle	$\angle$
surd	root symbol	$\surd$
bottom	falsehood	$\bot$
top	tautology	$\top$

4.4. **More fancy letters.** Mathematicians probably know more scripts than people in the other sciences – think of an archaic symbol from an archaic script, and the likelihood is that it is used in mathematics. Some typical things used in mathematics:

- The **blackboard bold math** symbols, given as `mathbb`. As we know, making bold symbols on the blackboard is rather difficult, so to distinguish certain letters as bold, mathematicians used double lines for those letters. For instance, $\mathbb{R}$ for the reals, $\mathbb{C}$ for the complex numbers, $\mathbb{Q}$ for the rationals, $\mathbb{Z}$ for the integers, $\mathbb{N}$ for the natural numbers, $\mathbb{H}$ for the Hamiltonians, and so on. LaTeX allows us to write these symbols via `mathbb`, thus, $\mathbb{R}$ is expressed as `\mathbb{R}`. Not all people use blackboard bold, but it is a reasonably common convention.
- The **German letters** and **calligraphic letters**: `mathfrak` (for German letters) and `mathcal` (for calligraphic letters). Here's how a few letters look in `mathfrak`: g becomes $\mathfrak{g}$, h becomes $\mathfrak{h}$. Under `mathcal`, A goes to $\mathcal{A}$ and B goes to $\mathcal{B}$.
- **Accents and inflections**: Check these out yourself!

4.5. **Making an equation array.** Here's a sample proof in mathematics, using an equation array and all:

```
\begin{proof}
  Let $b$ and $c$ be two inverses of $a$. Then we have :
  \begin{eqnarray*}
    b * (a * c) & = & (b * a) * c \text{ by associativity}\\
    \implies b * e & = & e * c \text{ by inverse property}\\
    \implies b & = & c \text{ by neutral element property}
  \end{eqnarray*}
  Thus, every element has a unique inverse.
\end{proof}
```

Note that the ampersand symbols are used for *alignment* here: the equations are aligned about the equality symbols.

Aligning stuff can be a pain at times, and i don't really know much about it.

4.6. **Lots of symbols all at once.** Here's a long list of equations:

$$
\begin{aligned}
\overline{F} &= m\overline{a} \text{ (Newton's law)} \\
\frac{\partial^2 f}{\partial x \partial y} &= \frac{\partial^2 f}{\partial y \partial x} \text{ (Fubini's theorem)} \\
e^{i\theta} &= \cos\theta + i\sin\theta \text{ (Euler's formula)} \\
\cosh^2 x - \sinh^2 x &= 1 \text{ (miscellaneous)} \\
(a+b)^n &= \sum_{p+q=n} \binom{n}{p,q} a^p b^q \\
\alpha, \beta &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \text{ (Quadratic roots)} \\
|A| + |B| &= |A \cap B| + |A \cup B| \text{ (inclusion-exclusion)} \\
\min_{x \in \mathbb{R}} |x| &= 0 \text{ (Exercise)} \\
\frac{d}{dx} \int_0^x f(x)\, dx &= f(x) \text{ (Fundamental theorem of calculus)} \\
\sum_{n=1}^{\infty} n^{-3} &\leq \pi^2/6 \\
x^n - 1 &= \prod_{d|n} \Phi_d(x) \\
|G \times H| &= |G||H|
\end{aligned}
$$

The code for these is given below:

```
\begin{eqnarray*}
  \overline{F} & = & m \overline{a} \text{ (Newton's law)}\\
  \frac{\partial^2 f}{\partial x \partial y} & = & \frac{\partial^2 f}{\partial y \partial x} \text{
  e^{i\theta} & = & \cos \theta + i \sin \theta \text{ (Euler's formula)}\\
  \cosh^2 x - \sinh^2 x & = & 1 \text{ (miscellaneous)}\\
  (a + b)^n & = & \sum_{p+q=n} \binom{n}{p,q} a^pb^q\\
  \alpha, \beta & = & \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \text{ (Quadratic roots)}\\
  |A| + |B| & = & |A \cap B| + |A \cup B| \text{ (inclusion-exclusion)}\\
  \min_{x \in \mathbb{R}} |x| & = & 0 \text{ (Exercise)}\\
  \frac{d}{dx} \int_0^x f(x) \, dx & = & f(x) \text{ (Fundamental theorem of calculus)}\\
  \sum_{n=1}^{\infty} n^{-3} & \leq & \pi^2/6 \\
  x^n - 1 & = & \prod_{d|n} \Phi_d(x)\\
  |G \times H| & = & |G||H|\\
\end{eqnarray*}
```

5. Global structural elements

5.1. **The golden rule.** The golden rule for global structural elements, is: *use the standard stuff out there.* Even if you don't *quite like the way* the section and subsection headings appear on-screen, follow the standard way.

As you get more and more comfortable, you can create your own style files that will control the display and show stuff *exactly the way* you want. Till then, be satisfied with less.

5.2. **Sections and subsections.** The typical `article` document class allows sections, subsections, and subsubsections. Section headings appear centered, in small caps, and are number 1,2,3 and so on. Subsections headings appear after section headings, separated by a period (something like 1.2 to indicate the second subsection of the first section). Subsubsection headings appear after the subsection headings separated by yet another period (something like 1.2.1). Subsubsection headings do not make it to the table of contents or bookmarks page.

5.3. **Table of contents.** To print the table of contents, the command is `\tableofcontents`. Typically, this is given right after the `maketitle` command.

If the package `hyperref` is being used, a table of contents automatically appears under `Bookmarks`, showing section headings as *top-level bookmarks* and subsection headings as *second-level bookmarks*. These bookmarks are not available in a printout.

It is possible to give a separate table of contents title to a section. This can be done by giving the table of contents title as an *optional argument* in square brackets.

5.4. **Section numbering.** The general convention for *not numbering* something in LaTeX is to put a $*$ after the command. Thus the command `section*` creates an unnumbered section, the command `subsection*` creates an unnumbered subsection, and the command `subsubsection*` creates an unnumbered subsubsection. Unnumbered sections, subsections, and subsubsections have *no effect* on the section number counts for the numbered sections, subsections and subsubsections.

So far, I haven't felt the need to interfere with the section numbering algorithms of LaTeX. However, interference is possible. At any point in the document, there is a section count (that is, which section are we in right now), a subsection count, and so on. There are commands that:

- Increment or decrement the counter by a fixed value.
- Reset the counter to a fixed value.

5.5. **Printing a table of contents.** Remember the points made about repeated compilation to have the table of contents printed properly.

5.6. **Appendices.** A LaTeX document can have an appendix, with the appendix appearing after the rest of the document is over, but *before* the command ending the document. The appendix is begun by the `\appendix` command, and it can have sections and subsections. Sections of the appendix are numbered using capital English letters.

6. Local structural elements

6.1. **The golden rule.** The golden rule for local structural elements, as for global structural elements, is: *use the standard stuff out there.* Even if the document you finally get doesn't look *exactly the way* you wanted it. Do not use man-made lists, use the standard lists. Do not try to manually simulate a theorem environment to make it look your own way, use the standard theorem environments. This not only saves you effort, it also helps others who may have to edit your text.

As you get more and more comfortable, you can create your own style files that will control the display and show stuff *exactly the way* you want. Till then, be satisfied with less.

6.2. **List environments.** LaTeX views a list as a sequence of items. Each list starts off with a \ begin{list environment} command, each item begins with a \ item command, and the list ends with a \ endlist environment command.

The three list environments differ in the way they view the items:

- The `enumerate` environment gives numbers to the items. This is the equivalent of *ordered list* or `ol` in HTML.
- The `itemize` environment gives bullet points to the items. This is the equivalent of *unordered list* or `ul` in HTML.

  For instance, the LaTeX for this list you're reading is of the `itemize` type.
- The `description` environment is somewhat different. Read about it yourself.

The above list's LaTeX source:

```
\begin{itemize}

\item The {\tt enumerate} environment gives numbers to the items.
  This is the equivalent of {\em ordered list} or {\tt ol} in HTML.

\item The {\tt itemize} environment gives bullet points to the items.
  This is the equivalent of {\em unordered list} or {\tt ul} in HTML.

  For instance, the \LaTeX \ for this list you're reading is
  of the {\tt itemize} type.
```

```
\item The {\tt description} environment is somewhat different. Read about
  it yourself.

\end{itemize}
```

Explore for yourself to find how LaTeX handles nested lists.
The golden rule: *whenever making lists, use the list environments.*

6.3. **Theorem environments.** You've probably often seen stuff like this in mathematical texts:

Theorem 1 (Beals' conjecture)**.** The equation

$$x^m + y^n = z^l$$

has no solutions for x, y and z pairwise relatively prime positive integers, and m, n and l all positive integers greater than 2.

Corollary 1 (Fermat's Last Theorem: Wiles '94)**.** The equation

$$x^n + y^n = x^n$$

has no solutions for $x, y, z \in \mathbb{N}$ with $n \geq 3$.

Proof. The only observation needed is that if the equation $x^n + y^n = z^n$ has a solution, then the gcd of any two of x, y and z divides the third. Hence, dividing by this common gcd, we get a solution where the entries are pairwise relatively prime. This contradicts Theorem 1 and we are done. □

This is the code dump of the stuff:

```
\begin{theorem}[Beals' conjecture]\rm\label{beals}
  The equation $$x^m + y^n = z^l$$ has no solutions for $x$, $y$ and $z$
  pairwise relatively prime positive integers,
  and $m$, $n$ and $l$ all positive integers greater than $2$.
\end{theorem}

\begin{corollary}[Fermat's Last Theorem: Wiles '94]\rm
  The equation $$x^n + y^n = x^n$$ has no solutions for $x,y,z \in \N$
  with $n \geq 3$.
\end{corollary}

\begin{proof}
  The only observation needed is that if the equation $x^n + y^n = z^n$
  has a solution, then the gcd of any two of $x$, $y$ and $z$ divides the third. Hence, dividing by
  are pairwise relatively prime. This contradicts Theorem \ref{beals}
  and we are done.
\end{proof}
```

Here's how theorem environments work:

- We first need to *inform* LaTeX of our new type of *theorem*. Possible things could be: definition, fact, claim, lemma, conjecture, open problem and so on. While informing, we write commands like these:

  ```
  \newtheorem{theorem}{Theorem}
  \newtheorem{crl}{Corollary}
  \newtheorem*{claim}{Claim}
  \newtheorem{propn}[theorem]{Proposition}
  ```

  The *first compulsory argument* is the *environment name* as we write it in the LaTeX source code. The *second compulsory argument* is the *screen name* of the environment. By default, each theorem environment has is own numbering, starting from 1, and going up by 1 each time.

  However, writing `newtheorem*` makes the theorem environment unnnumbered.

The *optional argument* can ask the theorem environment to number as per some other theorem environment. Thus in the fourth line, putting [theorem] tells LaTeX that the counter for propositions is to be shared with that of theorems. For this optional argument, it is also possible to put [section], indicating that each theorem is labelled by its section.

- Let's look more closely at the way we had written out Beals' conjecture:

```
\begin{theorem}[Beals' conjecture]\rm\label{beals}
  The equation $$x^m + y^n = z^l$$ has no solutions for $x$, $y$ and $z$
  pairwise relatively prime positive integers,
  and $m$, $n$ and $l$ all positive integers greater than $2$.
\end{theorem}
```

The word after `begin` is the environment name (not the screen name). There is then an *optional argument*, which gives the *name* of the theorem. The `rm` command after that is to make the text following it come in Roman text rather than emphasized (which is the default for theorem environments). The `label`, as we shall see section 7.2, is so that the theorem can be referred to later in the text.

6.4. **Mathematical environments.** A mathematical environment is one which operates in display math mode. Here are the important mathematical environments:

- `equation` and `equation*`: `equation` is the numbered variant and `equation*` is the unnumbered one. Equation often includes inequations as well ... as long as they're *facts*.
- `eqnarray` and `eqnarray*`: We already saw this in section 4.5. Remember that in the `eqnarray` environment, all the equation receive numbers, while in the `eqnarray*` environment, the equations are unnumbered.

6.5. **Tabular environments.** If you're looking for LaTeX to type in resumes and CVs, you'll need to learn the tabular environments. Here's how one of the tables we saw earlier on looks:

```
\begin{tabular}{|l|l|}
  \hline
  Final type & Command\\
  \hline
  DVI & {\tt latex}\\
  PDF & {\tt pdflatex}\\
  RTF & {\tt latex2rtf}\\
  HTML & {\tt latex2html}\\
  XML (addon) & {\tt ttm}\\
  \hline
\end{tabular}
```

A quick explanation of the tabular environment:

- The first line tells LaTeX to begin the tabular environment. It also needs to specify the *number of columns*, the *alignment of each column*, and whether to have a *separating vertical line* between columns. This saying `|l|l|` means: make a leftmost vertical line, then have a left-aligned column, make a vertical line, then have another left-aligned column, and then make another vertical line. Simply saying `lll|r` would mean three left-aligned columns, then a vertical line, and then a right-aligned column.
- The `hline` command makes a horizontal line. I chose to make a horizontal line on top, after the header row, and at the end of the table.
- For each row, the column entries are separated by & signs. The end of each row is indicated by the double backslash (same as the new line symbol in ordinary text mode).

7. Embellishments

7.1. **Indexing.** Here's the golden rule of indexing: *index your document as you write it.* Going through the document a second time in order to find stuff to index is a *real and unnecessary pain.* Moreover, indexing while writing forces one to write more carefully.

Here's an example of indexing in practice:

```
A subgroup of a group is termed {\em normal}\index{subgroup!normal}
if it is invariant under all inner automorphisms of the group.
```

Here's how indexing works:

- At a certain point in the LaTeX document, the index request is made. The index request uses the `index` command followed by the entry as it should appear in the index. The entry could be a plain word or number, or it could be a nested entry. For instance, `subgroup!normal` means the entry `normal` nested under the entry `subgroup`. The index also has other feature, such as redirecting from one index entry to another.
- During the first pass of compilation, a `.idx` file is created (with the same base name) storing all the index entries, along with the relevant page. Each time a compilation is done, this file is updated.
- When `makeindex` is run on the `.idx` file, the index entries are sorted alphabetically, and the new sorted index is stored in the `.ind` file with the same base name.
- When the original LaTeX document is compiled *after* the `.ind` file has been created, the `.ind` file is printed out wherever the document contains the `printindex` command.

7.2. **Labels and references.** A typical mathematical document refers back frequently to its own results and sections. The section number of a section is determined *at run time*, and hence it *should not* be put in the LaTeX source. The way LaTeX handles this is:

- Insert a label with a label name for any section, theorem or other object you think you may want to refer to. This works via the label command.
- At any place where you want to refer back to it, use the ref command, quoting the label name used.

If you use the package `hyperref`, an internal clickable link is created in the PDF file.

Here are things to which labels can be put:

- Chapters, sections, and subsections
- Equations (numbered)
- Theorems or other things in a theorem environment

However, the *set of label names* is common, so the same label name cannot be used for two different objects even if they are of different types.

For instance, in subsection 6.3, we had Beals' conjecture (Theorem 1). Check out the use of the label command there and the way it links up to this.

The above paragraph looks thus in verbatim:

```
For instance, in subsection \ref{theoremenv}, we had
Beals' conjecture (Theorem \ref{beals}). Check out the use
of the label command there and the way it links up to this.
```

The golden rule for labels and references are: *create the label for an object at the time you create the object, rather than when you need to refer back to it.*

7.3. **Footnotes.** At the point where you want the footnote inserted, simply throw in the command `footnote` with the content of the footnote[1] in curly braces. LaTeX will automatically figure out the layout and appearance so that the footnote appears on the same page as the text, and so that, if the footnote is too long, it gets split across pages. Here's how the above looked:

```
At the point where you want the footnote inserted, simply throw in the
command {\tt footnote} with the content of the footnote\footnote{Like
  this} in curly braces. \LaTeX \ will automatically figure out the
layout and appearance so that the footnote appears on the same page as
the text, and so that, if the footnote is too long, it gets split
across pages. Here's how the above looked:
```

7.4. **Bibliography.** LaTeX also offers in-built support for the creation of bibliographies. A bibliography is a list of references. In the article environment, the bibliography is printed at the end, under the header References, whereas in the report environment, it is printed under the header Bibliography.

Bibliographies are very similar to list environments, and a typical bibliography looks like:

```
\begin{thebibliography}{9}
\bibitem{Vipul}
Vipul Naik,
```

[1] Like this

```
Introduction to LaTeX,
LaTeX Books, 2007.

\bibitem{Cinderella revisited}
Vipul Naik,
Cinderella in the Internet age,
Fairy Tales, 2008.
\end{thebibliography}
```

A couple of points:

- The number 9 occurring there is called a widest-label parameter, and is typically chosen as a number higher than the total number of bibliography items.
- Each bibliography item begins with `\bibitem`, followed by the label of the item in curly braces. The two items here are labeled `vipul` and `Cinderella revisited`. These labels can be invoked elsewhere in the document, using the `cite` command. So, I can say:

  ```
  According to \cite{vipul}, LaTeX is easy to learn.
  ```

  This will print the reference number for the bibliography item `vipul`, in square braces.
- The text of the bibliography item following the label, is what gets printed for that item.

As with other forms of cross-referencing, multiple compilations are needed to get all the references correct.

8. Other miscellanea

8.1. **Diagrams in LaTeX.** LaTeX does offer a lot of diagramming support, but since I've never used it directly, I cannot comment much on it.

LaTeX can also import diagrams/figures of other formats. The package `graphicx` is typically used to aid in this importing. Imports can be from files of type `eps` (ecapsulated postscript) whne compiling to DVI and using DVIPS after that. There is some issue of *drivers* for `graphicx` that you can read more on at the "Not-so-short guide to LaTeX2e" (section 2.1).

Depending on the kind of diagrams you are keen on drawing, you may find suitable style files for those kinds of diagrams. Given the ongoing fascination of mathematicians with categories and morphisms, many packages exist that help ease the drawing of diagrams with lots of arrows and morphisms. Those who write a lot about finite state automata take the help of packages designed for easy construction of automata.

8.2. **Presentations.** LaTeX has an in-built document class called `slides`, which can be used for rudiemntary slide making. However, much better and more advanced slide preparation can be done using things like `beamer` and `prosper`. Here's how a typical slide in `beamer` looks:

```
\begin{frame}
  \frametitle{Existence of undecidable problems}
  \pause
  \begin{itemize}
  \item Turing Machine descriptions are of finite length \pause
  \item Countably many turing machines only \pause
  \item Only countably many computable functions!\pause
  \item Boolean functions correspond to subsets of strings, uncountable. \pause
  \item Clearly shows existance of undecidable problems\pause
  \end{itemize}

  Doh...a mathematician's proof, is there a constructive proof?
\end{frame}
```

The top stuff is as follows:

```
\documentclass{beamer}
\mode<presentation>
{
\usetheme{Marburg}
}
```

```
\usepackage{graphicx}
\usepackage{amsfonts}


\title{Which game is harder?}
\author[Ramprasad Saptharishi]{Ramprasad Saptharishi
}
\institute{
  Chennai Mathematical Institute\\Third Year B.Sc Mathematics
}
\date{September 6, 2006}



\AtBeginSection[]{
\begin{frame}<beamer>
\frametitle{Outline}
\tableofcontents[currentsection]
\end{frame}
}
```

9. Writing mathematics in LaTeX

9.1. **Think directly in LaTeX.** Think: what is the difference between native language speakers and non-native speakers? Native language speakers tend to *formulate their thoughts* in the language, so the words and sentences come out the correct way. Non-native speakers often need to think in another language, then perform a literal translation, and speak out the translated text.

Ideally, you want to directly convert your mathematical thoughts into a pretty-looking LaTeX file. It is important that you do so with *no gap between thought and result*, with *no barrier to putting your thoughts down*. And this is possible, it is not difficult, all it requires is a bit of practice at *directly puouring your thoughts into LaTeX* rather than first putting them in another format and then TeXing it out.

9.2. **Use words and not symbols.** Do *not* use symbols where words convey the meaning better. In particular, in a *verbal* explanation, avoid using symbols. Your goal should be that an English editor should be able to work directly with your LaTeX source. Take pride in keeping the content readable and editable.

Here's a passage with symbols overused:

```
A group is a set $G$ with a binary operation $*$ such that $\forall
g,h,k \in G$, we have $g * (h * k) = (g * h) * k$ (associativity) and
$\exists e$ such that $\forall g \ g* e = e* g = g$ and $\forall g exists h =
g^{-1}$ such that $g * h = h * g = e$. Associativity $\implies$
inverse is unique.
```

A better version:

```
A group is a set $G$ with a binary operation $*$ satisfying the
conditions of associativity, identity element, and existence of an
inverse. Associativity means that the order of parenthesization is
unimportant whne computing products (viz $g * (h * k) = (g * h) * k$
for all $g,h,k \in G$).  An identity element is an element $e$ such
that left (or right) multiplication with $e$ is the identity map (viz
$g * e = e * g = g$ for all $g \in G$) An inverse to $g$ is another
element $h$ such that both $g * h$ and $h * g$ are equal to $e$.

\begin{eqnarray*}
  \text{Associativity}: g * (h * k) & = & (g * h) * k \ \forall \ g,h,k \in G\\
  \text{Identity element}: \exists \ e \in G \text{ such that } g * e & = & e * g = g \forall g \in
  \text{Inverse}: \forall g \in G, \exists h \in G \text{ such that } g * h & = & h * g = e\\
\end{eqnarray*}
```

In particular, avoid putting quantifier and logic symbols within text.

9.3. **Local concerns.** Keep thinking locally of the following

- Am I giving a list of things? Which list environment is best suited?
- Am I providing data that is best expressed tabularly? How should the table look?
- Am I providing a new theorem, definition ,fact, result, conjecture etc.? What theorem environment should I use? What name should I give it?
- Is the section/theorem I am stating worth referring back to? What label should I give it?
- Is the term I am introducing/refering to worth indexing? What should I index it by?

9.4. **Global concerns.** Think of these:

- What is the purpose of my writing this document?
- What is the overall organization of the document? What is the top-level breakup into sections and subsections?
- What should the abstract contain?
- How will a reader locate stuff he/she is looking for within the document, easily?

9.5. **What all you can use LaTeX for.**

(1) **Scribing** or **preparing class notes**: When doing this as a collective effort, use a scribe style file and follow a uniform pattern in terms of both layout and content.
(2) **Jotting down ideas in rough**: Type in a quick abstract to indicate the main ideas you have. Give section names for the rough idea structures, and expand the ones you are clear upon.
(3) **Writing articles for others to view**: Similar to the previous one, except that you should be more careful about global structure. If the others happens to be a journal, be very clear of the style requirements of the journal as well as their submission instructions.
(4) **Creating presentations**: Not really covered in this talk.

10. New stuff and style files

10.1. **New commands.** Three important commands to create new commands are:

(1) `newcommand`: Here's how `newcommand` works. The first parameter is the *command name*. The command name *must* begin with a backslash, and must not conflict with existing LaTeX commands. The second parameter is the *command action*. LaTeX simply replaced the command name with the command action. The preplacement is done iteratively.

It is possible to create commands that accept *one or more* parameters. In this case the `newcommand` command itself must be given an *optional* parameter that counts the number of parameters.

The command action refers to the i^{th} parameter as $\#i$.

Some of the new commands that I often use:

```
\newcommand{\C}{\mathbb{C}}
\newcommand{\R}{\mathbb{R}}
\newcommand{\defined}[1]{{\bf #1}\tiny(defined)\normalsize}
\newcommand{\definedind}[1]{\defined{#1}\index{#1}}
\newcommand{\definedproperty}[3]{\defined{#1}\index{#3 #2}\index{#2!#3}}
\newcommand{\sdefinedproperty}[2]{\definedproperty{#2 #1}{#1}{#2}}
```

Let's understand these one by one. The first two lines simply create a shortcut for printing the blackboard bold complex numbers and real numbers. The third line is a little more complicated. It takes in an argument, and prints out that argument in bold face, followed by a tiny little thing saying (defined). Essentially, whenever I'm defining a term, I use the `defined` command.

The fourth command `definedind`, in addition to indicating that the new term has been defined, also automatically indexes the same entry.

Interpret the fifth and sixth commands yourself!

(2) `newtheorem` creates a new theorem environment. This was already discussed in section 6.3.
(3) `newenvironment` creates a new environment. Evey environment has a `\begin{environmentname}` followed by stuff within the environment, followed by a `\end{environment}` command. To create a new environment, we must specify the *action at the beginning* and the *action at the end*. Thus, the first argument is the *environment name*, the second argument is the *beginning action*, and the third argument is the *ending action*.

Some examples of environment definitions:

```
\newenvironment{pointstoponder}
       {\begin{center}{\sc Points to Ponder}\end{center}\begin{itemize}}
       {\end{itemize}}
\newenvironment{concepttester}
       {\begin{center}{\sc Concept Testers}\end{center}\begin{enumerate}}
       {\end{enumerate}}
\newenvironment{pauseandrecall}
       {\begin{center}{\sc Pause and Recollect}\end{center}\begin{enumerate}}
       {\end{enumerate}}
```

The first line of these creates a new environment titled `pointstoponder`. When such an environment is begun, the words POINTS TO PONDER are printed in the middle of the line. Then a list environment is begun. At the end of the environment, the list environment is also closed.

Other new environments can be similarly analyzed.

10.2. **Creating a style file.** A style file is just your (or somebody else's) *personal style* of LaTeXing things out. As you start writing more and more mathematical documents, you'll frequently find yourself needing a standard set of new commands, theorems and environments. You might also develop tastes for how the title, author and other details should be displayed, what should come in the headers and footers, and so on. All this needs to go into your style file.

In some cases, other people may impose their style files on you. This may be necessary if notes for a lecture series are being scribed by different people, or if papers have to be submitted for a journal with a particular format. Externally imposed style rules include:

- What environments to use for theorems, claims, propositions, lemmas, corollaries.
- How to create references or bibliography.
- Whether to have an index, table of contents, abstract, footnotes, or other embellishments.

Index