A Short Note on

Dilip Datta

Department of Mechanical Engineering, Tezpur University, Napaam, Tezpur - 784 028, India.
ddatta@tezu.ernet.in, datta_dilip@rediffmail.com
URL : http://www.tezu.ernet.in/dmech/people/ddatta.htm

Summary

LaTeX is a programming-based simple and easy approach for producing a document directly in the `dvi` or `pdf` format. LaTeX can be used for preparing letters, applications, articles, reports, publications, theses, books, or anything of that kind.

One of the major advantages of using LaTeX is that manual formatting of a document, as usually required in many word processors, can be automated in LaTeX. Therefore, the possibility of committing any mistake in formatting a document can be avoided, such as in numbering and referring items (sections, tables, figures, equations, or references), in choosing size and type of fonts for different sections and subsections, or in preparing bibliographic list. Further, LaTeX has the provision for automatically generating various lists of contents, index, and glossary.

The use of common word processors may be easier in preparing simple and small-size documents. But, the effort and time required in LaTeX for preparing complicated and big-size documents are quite less than those required in other word processors. One can become expert in LaTeX through a little practice. It can be realized that the preparation of only one academic dissertation would pay off all additional efforts required in learning LaTeX.

In spite of having such capabilities, a very limited number of books on LaTeX are available in market. Of course, a lot of resources on this subject can be obtained freely from the internet. However, most of the books emphasize on detailed documentation of LaTeX, while the internet-based resources are topic-specific. But people are either unable or not interested to spare time, during their busy schedules of research works, to understand and learn the detailed genotype of LaTeX covered in books, or to collect materials from different websites. Instead, they prefer to get direct and concise applications of various LaTeX syntax in a single window, which they can modify easily to get their works done in the least time and with the least effort.

Accordingly, this note is designed to present LaTeX in such a way that, even without having any prior working knowledge with LaTeX, one would understand easily, at least, how to prepare scientific research articles and reports, as well as academic dissertations. The main topics covered in this note include introduction to LaTeX, fonts selection, texts formatting, listing, tabbing, table preparation, figure insertion, equation writing, preparation of bibliographic list, etc. The note is concluded with article, thesis, and slide preparation in LaTeX.

Contents

1. Introduction to LaTeX

LaTeX is a macro-package used as a language-based approach for typesetting documents. Various LaTeX instructions are interspersed with the input file of a document, say `myfile.tex`, for obtaining the output as `myfile.dvi` or directly as `myfile.pdf`.

1.1. How to prepare a LaTeX input file?

The main structure of a LaTeX input file is divided into two parts (Fig. 1.1) – *preamble* and *body*.

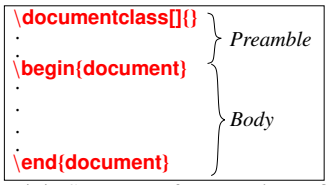


Fig. 1.1: Structure of a LaTeX input file.

The first part is the preamble that contains the global processing parameters for the entire document to be produced, such as the type of the document, page formatting, header and footer setting, inclusion of LaTeX packages for supporting additional instructions, and definitions of new instructions. The simplest preamble is `\documentclass{dtype}`, where `dtype` in `{}` is a mandatory argument as the class (or type) of the document, such as **letter**, **article**, **report**, or **book**. In the default setting, `\documentclass{}` prints a document on letter-size paper in 10 point fonts (1 point ≈ 0.0138 inch ≈ 0.3515 mm). Different user-defined formats for a document can be obtained through various options to `\documentclass{}`, in which case it takes the form of `\documentclass[fo1,fo2,...]{dtype}` with `fo1`, `fo2`, etc., in `[]` as the options (multiple options can be inserted in any order separating two options by a comma), e.g., `\documentclass[a4paper,11pt]{article}` for printing an article on A4 paper in 11 point fonts.

As shown in Fig. 1.1, the main body of a LaTeX input file starts with `\begin{document}` and ends with `\end{document}`. The entire contents to be printed in the output are inserted within the body, mixed with various LaTeX instructions (see §1.3 for detail). Any text entered after `\end{document}` is simply skipped.

A LaTeX input file is named with `tex` extension, say `myfile.tex`. It can be prepared in any operating system using any text editor that supports `tex` extension. There are also many open-source text editors developed specifically for preparing LaTeX input files, e.g., Kile (`kile.sourceforge.net`), Texmaker (`xm1math.net/texmaker`), WinEdt (`winedt.com`), etc.

A simple input file, say `myarticle.tex`, prepared under the document-class of **article** is shown in the left column of Table 1.1, along with its output in the right column. Surprisingly, the output is not the one as expected. The differences are shown underlined in the output file. This has happened due to the fact

Table 1.1: A simple LaTeX input file and its output.

LaTeX input	Output
\documentclass{article} \begin{document} **LaTeX** is a macro package for typesetting documents. It is a language-based approach, where LaTeX *instructions* are interspersed with the text file of a document, say `myfile.tex`, for obtaining the desired output as `myfile.dvi`. The `myfile.dvi` file can then be used to generate `myfile.pdf` file. \end{document}	LaTeX is a macro package for typesetting documents. It is a language-based approach, where LaTeX instructions are interspersed with the text file of a document, say myfile.tex, for obtaining the desired output as myfile.dvi. The myfile.dvi file can then be used to generate myfile.pdf file. 1

that many things in LaTeX can be obtained through some special instructions only as stated in §1.3.

1.2. How to compile a LaTeX input file?

A LaTeX input file can be compiled in many LaTeX editors mentioned in §1.1. Besides, many open-source LaTeX compilers are also available, e.g., MiKTeX (`miktex.org`) or TeXLive (`tug.org/texlive`). In a GUI-based compiler, like MiKTex or Kile, a LaTeX file can be compiled just by a mouse-click. In other command-line compilers, a LaTeX file is to be compiled through the pdflatex or latex command, followed by the name of the input file with or without its `tex` extension. For example, `myarticle.tex` of Table 1.1 can be compiled as:

```
    $ pdflatex myarticle.tex
Or, $ latex myarticle
```

The pdflatex command above will produce `myarticle.pdf` directly, while the latex command will produce 3 files, namely `myarticle.aux`, `myarticle.log` and `myarticle.dvi`. Out of these 3 files, `myarticle.dvi` can be viewed directly in a document viewer. It can also be used for producing `myarticle.pdf` as the output of `myarticle.tex` as follows:

```
$ dvipdf myarticle.dvi
```

1.3. LaTeX syntax

LaTeX syntax consists of commands and environments, which are kinds of instructions interspersed with the texts of a document for performing some specific jobs. Such instructions are defined in different packages.

1.3.1. Commands

A LaTeX command is an instruction used either for producing something new or to change the form of an existing item, e.g., producing the symbol α or printing `italic` as *italic*.

- ▷ A command usually starts with a \ (backslash), followed by one or more alphabets without any gap, e.g., \LaTeX and \copyright for producing LaTeX and © respectively.
- ▷ Many commands require some mandatory arguments, each in a separate pair of {}, e.g., \textcolor{blue}{`blue colored texts`} (detail is in §2.3) for printing the second argument in blue color.
- ▷ Many commands have the provision for some optional instructions also, written in [] separating two options by a comma, e.g., \documentclass[a4paper,11pt,twoside]{article}. A command with the provision for optional arguments must have at least one mandatory argument.
- ▷ A command without having any argument ignores trailing blank spaces. Hence, if followed by a word or a number, such a command should be ended by \␣ (the ␣ symbol indicates a blank space). For example, \copyright␣2007 will produce ©2007, while \copyright\␣2007 will produce © 2007. However, if such a command is followed by any punctuation, it needs not to be ended by \␣ as a punctuation is not to be preceded by any blank space.

1.3.2. Environments

A LaTeX environment is a structure composed of two complementary commands, within which some particular job can be performed, e.g., writing an equation or inserting a figure.

- ▷ The pair of complementary commands creating an environment are \begin{`ename`} and \end{`ename`}, where `ename` is the name of the environment, e.g., \begin{document} and \end{document} as shown in Fig. 1.1 creates the **document** environment (or the body) in a LaTeX input file.
- ▷ It is possible to use a command inside an environment, or to nest two or more environments, e.g., within the **document** environment, the \LaTeX command for printing LaTeX or the **figure** environment for inserting a figure.
- ▷ Many environments require some mandatory arguments placed after \begin{}, e.g., \begin{tabularx}{10cm}{XXX} for creating a table of three equal-width columns over 10 cm length through the **tabularx** environment.
- ▷ Like a command, many environments also have the provision for some optional instructions written in a pair of [], e.g., \begin{table}[t] preferring through the option **t** in the **table** environment to place a table at the top of a page.

1.3.3. Packages

The class (or type) of a document, incorporated through the \documentclass{} command, includes only basic features of LaTeX. Many additional commands and environments are defined in separate files, known as packages.

- ▷ A package is loaded in the preamble, in between the \documentclass{} and \begin{document} commands, through the \usepackage{pname} command, where pname is the name of the package, e.g., \usepackage{amssymb} for producing AMS type mathematical symbols.
- ▷ Like commands and environments, many packages also have the provision for some optional instructions in [], e.g., \usepackage[tight]{subfigure} preferring through the option tight to reduce extra space between figures.
- ▷ Unlike an option to \documentclass[]{}, which is global to the entire document, an option to \usepackage[]{} is local only to the features defined in the package(s) loaded through the \usepackage[]{} command.

1.4. Keyboard characters in LaTeX

Not all, but only the following characters of an English keyboard can be printed directly in a LaTeX document: alphabets (a b c d e f g h i j k l m n o p q r s t u v w x y z) both in uppercase and lowercase, digits (0 1 2 3 4 5 6 7 8 9), parentheses (( )), brackets ([ ]), quotations (` ' "), punctuations (, ; : ! . ?), math operators (+ - * / =), rate (@).

All other characters of an English keyboard need to be produced in a LaTeX document through some commands, which are (with their producing commands in parentheses) $ (\$), % (\%), { } (\{ \}), _ (_), ^ (\^\), & (\&), # (\#), \ ($\backslash$), ~ ($\sim$), | ($|$), < ($<$), and > ($>$). Note that the amssymb package may be required for the commands in $ $.

2. Fonts Selection

There are three modes for processing texts in LaTeX – *paragraph*-mode, *LR*-mode and *math*-mode. The paragraph-mode is for producing normal texts with automatic word-splitting, and line and page breaking to fit the texts within the specified area. In contrast, the LR-mode processes texts from left-to-right without any word-splitting and line breaking, such as \mbox{} or \fbox{} command whose arguments may span even beyond the specified width of a page. On the other hand, the math-mode is for writing mathematical expressions, like equations. In this note, the paragraph-mode and LR-mode will occasionally be addressed by a single name, known as the *text*-mode.

2.1. Text-mode fonts

The default font type of a LaTeX document is medium series serif family in upright shape and 10 pt size. The sizes of fonts in different parts of a document, say in headings and in paragraphs, are calculated proportionately. The default font setting can be altered globally through various options to the \documentclass[]{} command, e.g., \documentclass[12pt]{article} for producing an article in 12 pt fonts. The type of fonts in a particular segment can also be set manually.

Types of fonts in LaTeX are classified into four categories – *family*, *series*, *shape* and *size*. The detail of each category is given in Table 2.1, where atext is the piece of texts to be produced in the specified form.

Table 2.1: Different types of text-mode fonts used in LaTeX.

Type	Variety	Command
Family	Serif (default)	\textrm{atext} or {\rm atext}
	Sans serif	\textsf{atext} or {\sf atext}
	Typewriter	\texttt{atext} or {\tt atext}
Series	Medium (default)	\textmd{atext}
	Boldface	\textbf{atext} or {\bf atext}
Shape	Upright (default)	\textup{atext}
	Italic	\textit{atext} or {\it atext}
	Slanted	\textsl{atext} or {\sl atext}
	Caps & small caps	\textsc{atext} or {\sc atext}
	Emphasized	\emph{atext} or {\em atext}
Size	Tiny	{\tiny atext}
	Script	{\scriptsize atext}
	Foot note	{\footnotesize atext}
	Small	{\small atext}
	Normal (default)	–
	Large	{\large atext}
	Larger	{\Large atext}
	Largest	{\LARGE atext}
	Huge	{\huge atext}
	Hugest	{\Huge atext}

Different combinations of font family, series, shape and size in a logical way are allowed for producing a wide variety of fonts, e.g., \emph{\textbf{emphasized boldface fonts}} for producing '***emphasized boldface fonts***'. Also note that, in order to maintain a proper posterior vertical spacing, the arguments of the {\it }, {\em } and {\sl } commands may be followed by \/. For example, '{\it red\/} line' for producing '*red* line'.

2.2. Math-mode fonts

Like in text-mode, different types of fonts can be used in math-mode also as shown in Table 2.2.

Table 2.2: Different types of math-mode fonts used in LaTeX.

Font type	Command	Package required	Output
Serif	\mathrm{ABC abc}	—	ABCabc
Italic	\mathit{ABC abc}	—	*ABCabc*
Sans serif	\mathsf{ABC abc}	—	ABCabc
Typewriter	\mathtt{ABC abc}	—	ABCabc
Boldface	\mathbf{ABC abc}	—	**ABCabc**
	\boldmath{ABC abc}	amssymb	***ABCabc***
Normal	\mathnormal{ABC abc}	—	*ABCabc*
Calligraphic	\mathcal{A B C}	—	$\mathcal{ABC}$
Open	\Bbb{A B C}	amsfonts/ amssymb	$\mathbb{ABC}$
Open	\mathbb{A B C}	amsfonts/ amssymb	$\mathbb{ABC}$

1. If used in text-mode, the commands of Table 2.2 (except \boldmath{}) are to be written within a pair of $ symbol, e.g., $\mathbf{abc}$ for printing **abc**. In the case of the \boldmath{} command, the argument is to be enclosed in a pair of $ symbol, e.g., \boldmath{abc} for printing ***abc***.
2. The \mathcal{}, \mathbb{} and \Bbb{} commands do not accept lower-case letters.

3. Any blank space in the arguments of the commands of Table 2.2 is omitted. In such a case, most of the text-mode commands having the forms of \text..{} (e.g., \textbf{} or \textit{}) and \emph{}, as shown in Table 2.1, may be used for writing normal texts in math-mode preserving the space provided between two letters or words.

2.3. Colored fonts

Colored texts in LaTeX are supported by the color package. There are basically three types of color combinations – black and white (gray), additive primaries (rgb) and subtractive primaries (cmyk), under which black, white, red, green, blue, cyan, magenta and yellow are predefined colors. Apart from those, various new colors can be defined, through the \definecolor{}{}{} command, by setting different values to gray and each of the letters of rgb and cmyk as follows (where cname is the name of the user-defined new color):

\definecolor{cname}{gray}{w} ; $w \in [0, 1]$
\definecolor{cname}{rgb}{w, x, y} ; $w, x, y \in [0, 1]$
\definecolor{cname}{cmyk}{w, x, y, z} ; $w, x, y, z \in [0, 1]$

Once different colors are defined as above (if required), colored texts can be produced through the \textcolor{cname}{atext} command, where atext is the piece of texts to be colored by cname color. For example, \textcolor{blue}{this is in blue} will print 'this is in blue', while \textcolor{urgb}{this is in rgb = \{0,0.7,0.3\}} will print 'this is in rgb = {0,0.7,0.3}', where urgb is a new color defined as \definecolor{urgb}{rgb}{0,0.7,0.3}.

3. Texts Formatting

Many people format a document manually and hence commit many mistakes, such as type and size of fonts for headings of sectional units, numbering and referring sectional units, line and paragraph breaking, horizontal and vertical spacing, etc. LaTeX has numerous predefined macros for automatic and uniform formatting of a document without any mistake.

3.1. Sectional units

Various sectional units, like chapters and sections, are generated using the \chapter{}, \section{}, \subsection{}, \subsubsection{}, \paragraph{} and \subparagraph{} commands, whose argument is the heading or title of a sectional unit, e.g., the current section of this note is written as \section{Sectional units}. The sectional unit commands work in order and hence they should be nested properly, i.e., a \subsection{} command should follow a \section{} command or a \subparagraph{} command should follow a \paragraph{} command. As shown in Fig. 3.1, LaTeX assigns three-tier serial numbers to chapters, sections, subsections and subsubsections (paragraphs and subparagraphs are not numbered).

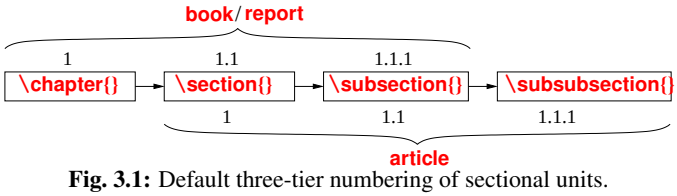


Fig. 3.1: Default three-tier numbering of sectional units.

Note that the numbering of a sectional unit can be omitted by using the starred form of the sectional command, such as \chapter*{}, \section*{}, \subsection*{} and \subsubsection*{}.

3.2. Labeling and referring numbered items

Like to sectional units addressed in §3.1, LaTeX assigns serial numbers to many environments or elements of an environment (e.g., table, figure, equation, or \item, which are discussed later). This default numbering system eliminates the possibility of committing any mistake as may happen in manual numbering. Moreover, LaTeX allows to label a numbered item by a unique *reference key*, which can be used to refer the item in any part within the same document (un-numbered items cannot be referred in this way). As illustrated in Table 3.1, the labeling and referring of an item are performed through \label{rkey} and \ref{rkey} respectively, where rkey is the assigned unique reference key of the item.

Table 3.1: Labeling and referring numbered items.

LaTeX input	Output
\section{Centre of gravity}\label{sec:cg} A point though which the resultant of the gravitational forces of all elemental weights of a body acts. % \section{Centre of mass}\label{sec-ex} The definition of the centre of gravity is given in Section~\ref{sec:cg} ...	**3.2 Centre of gravity** A point though which the resultant of the gravitational forces of all elemental weights of a body acts. **3.3 Centre of mass** The definition of the centre of gravity is given in Section 3.2 ...

3.3. Quoted texts

For quoting texts within quotation marks, (`) may be used as the left quote and (') as the right quote (each twice for double quotation), e.g., `single-quote' will produce 'single-quote', while ``double-quote'' will produce "double-quote".

For quoting an existing statement in a narrowed width without any change, the quote or quotation environment may be used (quote for a short display, while quotation for more than one paragraph).

3.4. New lines and paragraphs

LaTeX does not respond to a new line or paragraph set manually by pressing the enter button of the keyboard. Unless specified commands are used, LaTeX considers everything in a single line and single paragraph.

The command for creating a new line is \newline. A new line can also be created by using a line break command (\linebreak, \\, \\\\, or one or more blank lines) at the end of the previous line. Some extra vertical space above the next new line can also be specified in [] after the \\ command, e.g., \\[5mm] will create an extra vertical space of 5 mm above the next line.

Though a new paragraph can be started manually by creating a new line as above, the direct command for the same is \par. The \paragraph{} and \subparagraph{} commands can also be used for creating new paragraphs with the arguments of the commands as the headings of the paragraphs.

3.5. Creating and filling blank space

Excess blank spaces, created by pressing the `spacebar` or `tab` button of the keyboard, are just ignored in LaTeX, i.e., a sequence of blank spaces is treated as a single one only. LaTeX provides its own commands for creating a blank space of a specified size, both in horizontal and vertical directions, which are given in Tables 3.2 and 3.3. The need of a blank space after a

Table 3.2: Creating blank spaces.

Command	Package	Application	
\quad	—	x\quad␣y	x y
\qquad	—	x\qquad␣y	x y
\, or \thinspace	—	x\,y	x y
\: or \medspace	amsmath	x\:y	x y
\; or \thickspace	amsmath	x\;y	x y
\!	amsmath	x\!y	xy
\!\!	amsmath	x\!\!y	xy
\!\!\!	amsmath	x\!\!\!y	xy

command, ended by an alphabet and followed by another alphabet, can be avoided by writing the following alphabet or word in {}, e.g., 'x\quad{y}' to produce the same output as that by 'x\quad␣y'.

Table 3.3: Applications of some blank space creating commands.

LaTeX input	Output
\begin{center} \LaTeX\ in 24 Hours\bigskip\\ A Practical Guide for Writing \end{center}	LaTeX in 24 Hours A Practical Guide for Writing
\begin{center} \LaTeX\ in 24 Hours \vskip 5mm A Practical Guide for Writing \end{center}	LaTeX in 24 Hours A Practical Guide for Writing
\begin{center} \LaTeX\ in 24 Hours \vspace{5mm}\\ A Practical Guide for Writing \end{center}	LaTeX in 24 Hours A Practical Guide for Writing
Language: \hspace{8mm} English	Language: English
Marks: 100 \hfill Time: 3 Hrs	Marks: 100 Time: 3 Hrs

Units of the lengths in the \vskip, \vspace{} and \hspace{} commands can be any one of mm (millimeter), cm (centimeter), in (inch), pt (point), em (width of M) and ex (width of x). Apart from these units, a length can also be taken as a fraction of \textheight (height of texts on a page), \textwidth (width of texts in a page) or \linewidth (width of a column), e.g., 0.2\textheight for a vertical space of 20% of \textheight or 0.3\linewidth for a horizontal space of 30% of \linewidth.

3.6. Producing dashes within texts

LaTeX provides three types of dashes: -, – and —, which are produced by -, -- and ---, respectively. Out of these dashes, the shortest one is used between inter-related words, the medium one to indicate a range, while the longest one to show the extension of an expression as illustrated in Table 3.4.

Table 3.4: Dashes of different lengths.

LaTeX input	Output
Inter-related	Inter-related
May--August	May–August
Weather --- like clear sky	Weather — like clear sky

3.7. Foot notes

LaTeX provides the \footnote{} command for printing its argument as a foot note. As shown in Table 3.5, the command is to be inserted just after the word or phrase against which a foot note is to be generated. In the output, foot notes are numbered

Table 3.5: Foot notes generated through the \footnote{} command.

LaTeX input	Output
Both Rubi and Lila\footnote{They are sisters.} study in class I, while Ravi and Joy\footnote{They are friends.\label{fn:friends}} study in class II.	Both Rubi and Lila[1] study in class I, while Ravi and Joy[2] study in class II. [1]They are sisters. [2]They are friends.

in Arabic numerals and hence they can be labeled and referred using the \label{} and \ref{} commands as discussed in §3.2.

4. Listing Texts

Important matters are usually listed point-wise, either for concise presentation or for making them prominent. There are three listing environments, namely enumerate, itemize and description, where an item is written through an \item command.

4.1. Numbered listing through the enumerate environment

The enumerate environment produces a numbered list of items, where the items are numbered by Arabic numerals as shown in Table 4.1.

A maximum of four enumerate environments can be nested one inside another for producing a hierarchy of items, where an inner environment belongs to an \item of the previous environment. Table 4.2 illustrates three nested enumerate environments,

Table 4.1: Numbered listing through the enumerate environment.

LaTeX input	Output

`Some states of India:`
`\begin{enumerate}`
`\item Assam`
`\item Punjab`
`\item Rajasthan.`
`\end{enumerate}`
	Some states of India:
	1. Assam
	2. Punjab
	3. Rajasthan.

Table 4.2: Nested listing through the enumerate environment.

LaTeX input	Output

`\begin{enumerate}`
`\item India\label{item:Ind}`
`\begin{enumerate}`
`\item Assam\label{item:Ass}`
`\begin{enumerate}`
`\item Nagaon\label{item:Nag}`
`\item Kamrup`
`\item Cachar`
`\end{enumerate}`
`\item Bihar`
`\item Punjab`
`\end{enumerate}`
`\item Sri Lanka`
`\end{enumerate}`
`District~\ref{item:Nag} belongs to state~\ref{item:Ass} of country~\ref{item:Ind}.`
	1. India
	(a) Assam
	i. Nagaon
	ii. Kamrup
	iii. Cachar
	(b) Bihar
	(c) Punjab
	2. Sri Lanka
	
	District 1(a)i belongs to state 1a of country 1.

which also shows how their items can be labeled and referred through the \label{} and \ref{} commands respectively (blank spaces preceding inner lines in the LaTeX input are kept only for easy understanding of a loop, otherwise they do not have any sense in LaTeX). Notice in Table 4.2 the numbering and referring styles of items in the nested enumerate environments.

4.2. Unnumbered listing through the itemize environment

Unnumbered lists are produced through the itemize environment. Like the enumerate environment, a maximum of four itemize environments can be nested one inside another. Table 4.3

Table 4.3: Nested listing through the itemize environment.

LaTeX input	Output

`\begin{itemize}`
`\item India`
`\begin{itemize}`
`\item Assam`
`\begin{itemize}`
`\item Nagaon`
`\item Kamrup`
`\item Cachar`
`\end{itemize}`
`\item Bihar`
`\item Punjab`
`\end{itemize}`
`\item Sri Lanka`
`\end{itemize}`
	• India
	– Assam
	∗ Nagaon
	∗ Kamrup
	∗ Cachar
	– Bihar
	– Punjab
	• Sri Lanka

illustrates three nested itemize environments (items of the itemize environment cannot be referred as they are not numbered). Notice in Table 4.3 the marking styles of items in the nested itemize environments.

4.3. Listing with user-defined labels through the description environment

The description environment facilities to prepare a list of items with user-defined labels. Like in the itemize environment, the items of the description environment also cannot be referred by any serial number. An item in the description environment is labeled through an optional argument to the \item command, which can be anything, like `(a)`, `(b)`, `(i)`, `(ii)`, or `Rule`, `Action`, etc. which is printed in boldface fonts, e.g., `\item[(a)]` will label its item by **(a)**. Such an example is given in Table 4.4. Like the enumerate and itemize environments, the description environments can also be nested one inside another.

Table 4.4: Listing with user-defined labels through the description environment.

LaTeX input	Output

`\begin{description}`
`\item[(a)] Assam`
`\item[(b)] Bihar`
`\item[(c)] Punjab`
`\item[(d)] Rajasthan.`
`\end{description}`
	(a) Assam
	(b) Bihar
	(c) Punjab
	(d) Rajasthan.

4.4. Nesting different listing environments

Nesting of different listing environments is also possible for producing a hierarchy of items as illustrated in Table 4.5.

Table 4.5: Nested different listing environments.

LaTeX input	Output

`\begin{enumerate}`
`\item SI System`
`\begin{enumerate}`
`\item Metre`
`\item Newton`
`\item Second`
`\end{enumerate}`
`\item MKS System`
`\begin{itemize}`
`\item Metre`
`\item Kilogram`
`\item Second`
`\end{itemize}`
`\item FPS System`
`\begin{description}`
`\item [(i)] Foot`
`\item [(ii)] Pound`
`\item [(iii)] Second`
`\end{description}`
`\end{enumerate}`
	1. SI System
	(a) Metre
	(b) Newton
	(c) Second
	2. MKS System
	• Metre
	• Kilogram
	• Second
	3. FPS System
	(i) Foot
	(ii) Pound
	(iii) Second

5. Tabbing Texts

The tabbing environment is used for aligning texts in different columns. The \= command is used, usually in the first row, to generate a new column by ending the current column. The \> command moves the control to the next column in the subsequent rows. Each row is terminated by a line brake command

\\ to go to the next row (the last row is not required to be terminated by \\). Table 5.1 shows a simple two-column example of tabbing through the tabbing environment.

Table 5.1: Tabbing texts through the tabbing environment.

LaTeX input	Output
\begin{tabbing} Potato \= 12.00\\ Rice \> 20.00\\ Oil \> 60.00\\ Sugar \> 23.00 \end{tabbing}	Potato 12.00 Rice 20.00 Oil 60.00 Sugar 23.00

In the tabbing environment, columns of required widths and number can be generated using the \kill command. In that case, all the columns are generated in the first row itself, where the entry of a column is the widest entry which appears later in that column. Finally, the first row is ended by the \kill command, instead of the line breaking command \\, instructing not to print the row but just to generate the columns. Such an example is shown in Table 5.2.

Table 5.2: Adjusting tabbing column width in the tabbing environment using the \kill command.

LaTeX input	Output
\begin{tabbing} Base area (A) \= = bdh \= = 24\kill Breadth (b) \> = 3\\ Depth (d) \> = 2\\ Height (h) \> = 4\\ Volume (V) \> = bdh \> = 24\\ Base Area (A) \> = bd \> = 6\\ \end{tabbing}	Breadth (b) = 3 Depth (d) = 2 Height (h) = 4 Volume (V) = bdh = 24 Base Area (A) = bd = 6

6. Table Preparation

A table is used for presenting data or items row and column-wise in a concise form. In LaTeX, tables are prepared through the tabular or tabularx environment. However, tables produced by these environments are not assigned any serial number or title, which are generally required for identifying a table. For this purpose, the tabular and tabularx environments are usually put inside the table environment.

6.1. Table through the tabular environment

In the tabular environment, the columns of a table are generated through the mandatory argument of the environment, e.g., \begin{tabular}{|l|c|c|c|} in Table 6.1 generates a five-column table. A column is generated through one of the three letters of l, r and c, each of which represents a column as well as the alignment of the entries in that column (l for left alignment, r for right alignment and c for center alignment). The | symbol in the argument of \begin{tabular}{} is used either to mark a boundary or to separate two columns by a vertical line in the specified location, covering the full height of the table. Following the \begin{tabular}{} command, the column-wise entries of a row are inserted, separating two entries by an & and ending the row by a line break command \\. Further, the \hline command is used either to mark a boundary or to separate two rows by a horizontal line in the specified location, covering the full width of the table. Finally, the tabular environment is ended by the \end{tabular} command.

The table environment, inside which the tabular environment is put in Table 6.1, is started through the \begin{table}[!hbt] command (the optional argument !hbt is for the preferred vertical positioning of the table, which is explained in §6.3). The next command is \centering, which instructs for width-wise center alignment of the table (other commands could be \flushleft for left alignment or \flushright for right alignment). The \caption{attl} command used in the table environment (but outside the tabular environment) assigns a serial number to the table along with its argument attl as the title (caption) of the table (since the title usually comes on the top of a table, the \caption{} command is used before the tabular environment). Following the \caption{} command, the \label{} command is inserted with a unique reference key, which as shown in Table 6.1 can be used in the \ref{} command for referring the table anywhere in the document. Note that \label{} is always used after \caption{}. Moreover, \label{} does not have any effect without \caption{}, in which case the table is not numbered. Also note that %, used in Table 6.1 after \end{tabular}, is a commented line (any texts in a line preceded by % is ignored by LaTeX compilers).

6.2. Table through the tabularx environment

The width of a column generated by one of the options of l, c and r, as discussed in §6.1, is made equal to the length of the longest entry in that column. This may extend a table even beyond the width of a page if the table has some very long entries.

The tabularx package provides the tabularx environment, which can calculate automatically the width of a column so as to restrict a table within a pre-specified horizontal width irrespective of the lengths of the entries in the table. The tabularx environment takes two mandatory arguments, i.e., \begin{tabularx}{awidth}{acols}, where awidth is the horizontal width of the table and acols is its columns. The columns in the tabularx environment are generated in the same way as in the tabular environment. A fixed-width column is generated through l, c or r, while a flexible-width column (i.e., a column whose width is to be calculated automatically) is generated through a X. All the flexible-width columns of a table are of equal width, which is calculated internally based on the values of the total width (awidth) of the table and total width of the fixed-width columns. Entries in a flexible-width column are made full aligned. Other alignments can be obtained using either >{\raggedright\arraybackslash}, >{\centering\arraybackslash} or >{\raggedleft\arraybackslash} before X, which make the entries left, center and right aligned respectively. Table 6.2 shows an application of the tabularx environment for generating a three-column table of a total width of 80% of the page width, i.e., 0.8\linewidth (a fixed value, say 10cm or 6in, can also be used). Since the middle column is generated by the option c, its width is fixed by the longest entry in that column. The ex-

Table 6.1: A simple table through the tabular environment.

LaTeX input

```
\begin{table}[!hbt]
\centering
\caption{Obtained marks.}
\label{tab-marks}
\begin{tabular}{|l|c|c|c|c|}
\hline Name   & Math & Phy & Chem & English\\
\hline Robin  & 80   & 68  & 60   & 57\\
\hline Julie  & 72   & 62  & 66   & 63\\
\hline Robert & 75   & 70  & 71   & 69\\
\hline
\end{tabular}
\end{table}
%
Table~\ref{tab-marks} shows the ...
```

Output

Table 1: Obtained marks.

Name	Math	Phy	Chem	English
Robin	80	68	60	57
Julie	72	62	66	63
Robert	75	70	71	69

Table 1 shows the marks obtained by three students in the final examination.

Table 6.2: A simple table through the tabularx environment.

LaTeX input

```
\begin{table}[!hbt]
\centering
\caption{Scored points.}
\begin{tabularx}{0.8\linewidth}{|X|c|>{\raggedleft\arraybackslash}X|}
 \hline {\bf Name} & {\bf Sex} & {\bf Points}\\
 \hline Milan      &     M     &        1,500\\
        Julie      &     F     &        1,325\\
        Sekhar     &     M     &          922\\
        Dipen      &     M     &          598\\
        Rubi       &     F     &           99\\
 \hline
\end{tabularx}
\end{table}
```

Output

Table 2: Scored points.

Name	**Sex**	**Points**
Milan	M	1,500
Julie	F	1,325
Sekhar	M	922
Dipen	M	598
Rubi	F	99

treme two columns are generated by the option X, for which their widths are equal and calculated internally to accommodate all the three columns in the pre-specified width of the table. Moreover, the last column is made right aligned by generating it through >{\raggedleft\arraybackslash}X, instead of just through X. All other matters of Table 6.2 are same with those of Table 6.1.

6.3. Vertical positioning of tables

As shown in Tables 6.1 and 6.2, the preferred vertical position of a table on a page can be specified as an optional argument to the table environment, i.e., \begin{table}[avp], where avp is the specifier for vertical positioning of the table. The commonly used specifiers are h, b and t, which stand for here, bottom of the page, and top of the page, respectively. These specifiers can be used individually or in a combination of two or three in any order. Moreover, for placing the table in the specified position even if enough space is not available on the current page, the specifier or the combination of the specifiers may be preceded by a ! symbol, like !h, !b or !hbt.

Besides h, b and t, the float package provides the specifier H, which instructs to put a table *here* only. If the blank space on the current page is not sufficient to hold the table, it is taken to the top of the next page along with the texts that follow the table, by leaving the current page incomplete. The specifier H is used alone, i.e., H is not combined with ! or any of h, b and t.

6.4. Merging rows and columns of tables

When presenting different types of information in a table, some cells are often required to be merged into a single one. The multirow package provides the \multicolumn{}{}{} and \multirow{}{}{} commands for merging two or more columns and rows respectively, which are illustrated in Table 6.3.

In \multicolumn{n_c}{calign}{centry}, n_c is the number of columns to be merged, calign is the alignment of the merged column and centry is the entry of that merged cell. Since 4 columns in the first row in Table 6.3 are merged into a single cell, the number of entries in that row is reduced from 6 to 3. The permitted calign in the tabular environment is l (for left alignment), r (for right alignment) or c (for center alignment).

Similarly, in \multirow{n_r}{cwidth}{centry}, n_r is the number of rows to be merged, cwidth is the width of the merged cell and centry is the entry of that merged cell. The value of cwidth can be set manually (e.g., 25mm or 1.0in), or can be obtained an auto-adjusted one using an * only. When some rows in a column are merged, \multirow{}{}{} is used in the first row to be merged and the column in each of the remaining merged rows is left blank (i.e., the column is ended simply by a & or \\) as shown in the first and last columns in the second row of Table 6.3.

Further, the \cline{m-n} command is used in Table 6.3 for drawing a horizontal line covering columns m to n only.

Table 6.3: Merging two or more cells of a table into a single one.

LaTeX input

```
\begin{tabular}{|l|*{5}{c|}}
\hline \multirow{2}{*}{Name}&  \multicolumn{4}{c|}{Subjects}& \multirow{2}{*}{Total}\\
 \cline{2-5}          & Math & Phy & Chem & English &    \\
 \hline      Robin    &  80  &  68 &  60  &    57   & 265\\
 \hline      Julie    &  72  &  62 &  66  &    63   & 263\\
 \hline      Robert   &  75  &  70 &  71  &    69   & 285\\
 \hline
\end{tabular}
```

Output

Name	Subjects				Total
	Math	Phy	Chem	English	
Robin	80	68	60	57	265
Julie	72	62	66	63	263
Robert	75	70	71	69	285

6.5. Tables in multi-column documents

In a multi-column document, where texts are printed column-wise, a table is also placed in a single column. However, if the width of the column is not large enough to accommodate a table in it, the **table*** environment may be used for producing the table over the entire width of the page. In that case, the **\begin{table}** and **\end{table}** commands are to be replaced by the **\begin{table*}** and **\end{table*}** commands respectively.

6.6. Tables at the end of a document

Some publishers want the tables and figures of an article are to be grouped at the end of the article. Just the inclusion of the **endfloat** package in a normal document automatically performs this job, regardless of the actual positions of the tables and figures in the LaTeX input file (the **endfloat** package produces two auxiliary files with `fff` and `ttt` extensions for writing information about the figures and tables respectively). Not only the tables and figures are grouped at the end of the document, notes are also produced in their actual positions, like `[Table 3 about here.]` or `[Figure 7 about here.]`. Moreover, the tables and figures are preceded by two lists, namely '`List of Tables`' and '`List of Figures`' respectively, containing their contents. The produced lists of tables and figures can be turned off by using the **\notablist** and **\nofiglist** commands in the preamble. Note that the use of the **endfloat** package may require an additional `latex` run to move the tables and figures at the end of the document (§9.3 discuss about the `latex` run).

7. Figure Insertion

It is stated in §1.2 that a LaTeX file can be compiled using either the latex or pdflatex command. When a LaTeX file involves figures from external files, either of the compilation commands is to be used based on the format of the figures. Note that the file formats of all the inserted figures must be supported by a single compilation command, either latex or pdflatex. Figure formats supported by the latex command are `eps` (encapsulated postscript) and `ps` (postscript), while those supported by the pdflatex command are `pdf` (portable document format), `jpg` (joint photographic expert group), `tiff` (tag index file format) and `png` (portable network graphic).

7.1. Commands and environment for inserting figures

An `eps` format figure can be inserted using the **\epsfig{file=**`fname`**}** command defined in the **epsfig** package, where `fname` is the name of the figure file with or without the `eps` extension. Additionally, the size of a figure can also be specified in **\epsfig{}** through two optional fields, **width** and **height**, one separated from another by a comma. Without any of the **width** and **height**, a figure is printed in its original size. If one of them is specified, the other one is automatically taken in proportion. On the other hand, the presence of both **width** and **height** prints a figure in the specified fixed size (in this case, the figure may get distorted if their values are not set properly). Further, a figure can also be rotated through the option **angle=**`theta`, where a positive value of `theta` (in degree) will rotate the figure in counter-clockwise direction and a negative value in clockwise direction.

The more general command for inserting a figure is **\includegraphics[**`aopt`**]{**`fname`**}** defined in the **graphicx** package, where `fname` is the name of the figure file without its extension, and `aopt` is(are) the option(s) like **width**, **height** and **angle**. The advantage of using **\includegraphics[]{}** is that a figure in any format can be inserted without making any change in the input file.

Similar to nesting the **tabular** or **tabularx** environment in the **table** environment as discussed in §6.1, the **\epsfig{}** and **\includegraphics[]{}** commands can be used in the **figure** environment, so that a figure can be assigned a serial number and a caption through the **\caption{}** command, as well as a reference key through the **\label{}** command for the purpose of referring it anywhere within a document. Further, similar to the **table** environment, the **figure** environment can also be created as **\begin{figure}[]** with optional preferences in **[]** for vertical positioning of a figure. The standard preferences for vertical positioning are **H**, and any or combination of **h**, **b** and **t** along with **!** (refer §6.3 for detail of **[H]** and **[!hbt]**).

7.2. Inserting a simple figure

Two examples of inserting a figure are shown in Table 7.1, where the first command in the **figure** environment is **\centering** that instructs for width-wise center alignment of its figure (other commands could be **\flushleft** for left alignment or **\flushright** for right alignment). The **\caption{}** command is used for assigning a serial number to the figure and for printing its argument as the title of the figure (since the title usually comes at the bottom of a figure, **\caption{}** is used after **\epsfig{}**). The **\caption{}**

Table 7.1: Figure insertion through the \epsfig{} command.

LaTeX input	Output
\begin{figure}[!hbt] \centering \epsfig{file=girl.eps, width=2.0cm} \caption{A girl.} \label{girl1} \end{figure}	 Figure 1: A girl.
\begin{figure}[!hbt] \centering \epsfig{file=girl, width=2cm, angle=30} \caption{A girl.} \label{girl2} \end{figure}	 Figure 2: A girl.

command is followed by the \label{} command for assigning a unique reference key, which can be used for referring the figure through the \ref{} command.

7.3. Sub-numbering a group of figures

A group of figures can be sub-numbered under a main number, e.g., 3(a) or 5(e). Within the figure environment, the \subfigure[atitle]{afig} command defined in the subfigure package (or the new \subfloat[atitle]{afig} command defined in the subfig package) may be used for inserting a figure with a sub-numbering, where optional atitle is the title of the figure, and mandatory afig is the insertion of the figure either through the \epsfig{} or \includegraphics[]{} command. For the purpose of referring, a sub-figure can be assigned a reference key through the \label{} command inside the mandatory argument of \subfigure[]{}. Moreover, the group of sub-figures can be captioned and labeled as a whole using respectively the \caption{} and \label{} commands inside the figure environment. Table 7.2 shows such an example, which also shows that the sub-figures can be inserted in a single row or even in multiple rows (for inserting a sub-figure in the next row, a line break command \\ is to be used at the end of the previous \subfigure[]{} command).

7.4. Figures in multi-column documents

In a multi-column document, where texts are printed column-wise, a figure is also placed in a single column. However, if the width of the column is not large enough to accommodate a figure in it, the figure* environment may be used for inserting the figure on the entire width of the page. In that case, the \begin{figure} and \end{figure} commands are to be replaced by the \begin{figure*} and \end{figure*} commands respectively.

7.5. Figures at the end of a document

Refer §6.6 for detail.

8. Equation Writing

Mathematical expressions or equations in LaTeX are written in math-mode environments, such as equation or eqnarray. The math-mode environments are defined in the amsmath package, while mathematical symbols are in the amssymb package.

8.1. Basic mathematical notations and delimiters

Since various mathematical notations are basic tools for writing mathematical expressions, LaTeX commands for some frequently used notations are listed in Table 8.1 as a quick reference. Also, Greek letters used as symbols are given in Table 8.2.

Some basic delimiters, a pair of which acts like parentheses to enclose an expression, are also given in Table 8.3. To fit automatically around the height of a mathematical expression, the \left and \right commands may be used before the opening and closing delimiters (\left and \right are used as a complementary pair). The two delimiters enclosing an expression need not to be similar. For example, \left(and \right] can be used to enclose an expression in ( ]. If no delimiter is required in one side of an expression, the \left. or \right. command, as applicable, may be used. On the other hand, for fixed big-size delimiters, the \big, \Big, \bigg and \Bigg commands may be used by appending l and r for producing opening and closing delimiters respectively, e.g., \biggl\{ and \biggr\} will produce a pair of big-sized curly braces. Note that none of the commands of the forms of \big, \Big, \bigg and \Bigg is required to appear in a complementary pair, i.e., either the opening or closing delimiter can also be used alone as shown in the right column of Table 8.3.

8.2. Mathematical operators

A mathematical expression is formed by connecting various terms through some operators, which are classified as binary operators and relation operators. Such operators along with their LaTeX commands are given in Tables 8.4 and 8.5.

8.3. Mathematical expressions in text-mode

A mathematical expression, say amath, can be inserted in running texts of a paragraph as $amath$, \(amath\) or \begin{math}amath\end{math}, where '$$', '\(\)' or the math environment create math-modes in running texts. A single notation is usually inserted in $$, while an expression is inserted in \(\) or in the math environment (however, all three are applicable in either case). Consider the following example:

> The equation of an origin-centered circle is $x^2 + y^2 = r^2$, where x and y are the coordinates of a point on the circumference of the circle, and r is its radius.

Table 7.2: Sub-numbering a group of figures using the \subfigure[]{} command.

LaTeX input

```
\begin{figure}[!htb]
\centering
\subfigure[A girl.]
{   \includegraphics[width=2.0cm]{girl}
    \label{girl}
}\hfill
\subfigure[A flower.]
{   \includegraphics[width=2.0cm]{flower}
    \label{flower}
}\\
\subfigure[A finger work.]
{   \includegraphics[width=4.0cm]{finger}
    \label{finger-work}
}
\caption{Girl, flower and finger work.}
\label{girl_flower_finger}
\end{figure}
%
In Figure~\ref{girl_flower_finger}, \ref{girl} and \ref{flower}
display a girl and a flower, while \ref{finger} displays a
beautiful finger work.
```

Output

(a) A girl.

(b) A flower.

(c) A finger work

Figure 3: Girl, flower and finger work.

In Figure 3, 3(a) and 3(b) display a girl and a flower, while 3(c) displays a beautiful finger work.

Table 8.1: Frequently used mathematical notations (math-mode).

Function	Command with application	Output
Prime	`p'`	p'
Dots	`\dot{x}`, `\ddot{x}`	$\dot{x}$, $\ddot{x}$
	`\dddot{x}`, `\ddddot{x}`	$\dddot{x}$, $\ddddot{x}$
Single sub-/super-script	`x_i`, `x^2`	x_i, x^2
Multiple sub-/super-scripts	`x_{ij}`, `x^{2k}`	x_{ij}, x^{2k}
Subscript and super-script	`x^{2k}_{ij}` or `x_{ij}^{2k}`	x^{2k}_{ij}
Summation	`\sum`, `\sum_{i=1}^{20}`	$\sum$, $\sum_{i=1}^{20}$
Product	`\prod`, `\prod_{i=1}^{i=20}`	$\prod$, $\prod_{i=1}^{20}$
Integration	`\int x^2\,dx`, `\int_a^b xy\,dx`	$\int x^2\,dx$, $\int_a^b xy\,dx$
Multiple integration	`\iint\limits_s`, `\iiint\limits_v`	$\iint_s$, $\iiint_v$
	`\iiiint`	$\iiiint$
Set of integrations	`\idotsint`	$\int\cdots\int$
Cyclic integration	`\oint`	$\oint$
Fraction	`\frac{x}{y}`	$\frac{x}{y}$
Derivative	`\nabla{f}`, `\frac{dx}{dy}`	∇f, $\frac{dx}{dy}$
Partial derivative	`\frac{\partial{y}}{\partial{x}}`	$\frac{\partial y}{\partial x}$
Root	`\sqrt{x}`, `\sqrt[5]{xyz}`	$\sqrt{x}$, $\sqrt[5]{xyz}$
Limit	`\lim_{x\to 0}`	$\lim_{x\to 0}$
	`\underset{x\to 0}{\lim}`	$\underset{x\to 0}{\lim}$
Exists/not exists	`\exists`, `\nexists`	$\exists$, $\nexists$
Modes	`\mod{n^2}`, `\bmod{n^2}`	$\mod n^2$, $\bmod n^2$
	`\pmod{n^2}`, `\pod{n^2}`	$\pmod{n^2}$, $\pod{n^2}$
Binomial expression	`\binom{n}{k}`	$\binom{n}{k}$

where the equation can be inserted as `\(x^2 + y^2 = r^2\)` or `\begin{math}x^2 + y^2 = r^2\end{math}`, while the variables x, y and r as `$x$`, `$y$` and `$r$` respectively.

Table 8.2: Greek letters (math-mode).

Sym. command	Sym. command	Sym. command	Sym. command
Lowercase	κ \kappa	υ \upsilon	Ξ \Xi
α \alpha	λ \lambda	ϕ \phi	Π \Pi
β \beta	μ \mu	φ \varphi	Σ \Sigma
γ \gamma	ν \nu	χ \chi	Υ \Upsilon
δ \delta	ξ \xi	ψ \psi	Φ \Phi
ϵ \epsilon	π \pi	ω \omega	Ψ \Psi
ε \varepsilon	ϖ \varpi	**Uppercase**	Ω \Omega
ζ \zeta	ρ \rho	Γ \Gamma	**AMS Greek**
η \eta	ϱ \varrho	Δ \Delta	$\digamma$ \digamma
θ \theta	σ \sigma	Θ \Theta	$\varkappa$ \varkappa
ϑ \vartheta	ς \varsigma	Λ \Lambda	
ι \iota	τ \tau		

Table 8.3: Basic delimiters (math-mode).

Del.	Command	Del.	Command
$\left(\frac{x}{y}\right)$	\left(\frac{x}{y} \right)	$\bigl(\frac{x}{y}\bigr)$	\bigl(\frac{x}{y} \bigr)
$\left(\frac{x}{y}\right.$	\left(\frac{x}{y} \right.	$\Bigl(\frac{x}{y}$	\Bigl(\frac{x}{y}
$\left.\frac{x}{y}\right)$	\left. \frac{x}{y} \right)	$\frac{x}{y}\biggr)$	\frac{x}{y} \biggr)
$\left\{\frac{x}{y}\right\}$	\left\{ \frac{x}{y} \right\}		
$\left\{\frac{x}{y}\right.$	\left\{ \frac{x}{y} \right.	$\biggl\{\frac{x}{y}\biggr\}$	\biggl\{ \frac{x}{y} \biggr\}
$\left.\frac{x}{y}\right\}$	\left. \frac{x}{y} \right\}		
$\left[\frac{x}{y}\right]$	\left[\frac{x}{y} \right]	$\Biggl(\frac{x}{y}\Biggr)$	\Biggl(\frac{x}{y} \Biggr)
$\left[\frac{x}{y}\right.$	\left[\frac{x}{y} \right.		
$\left.\frac{x}{y}\right]$	\left. \frac{x}{y} \right]	$\frac{x}{y}\Biggr\}$	\frac{x}{y} \Biggr\}
$\left\|\frac{x}{y}\right\|$	\left| \frac{x}{y} \right|		
$\left\|\frac{x}{y}\right.$	\left| \frac{x}{y} \right.	$\frac{x}{y}\biggr\|$	\frac{x}{y} \biggr|
$\left.\frac{x}{y}\right\|$	\left. \frac{x}{y} \right|		

Table 8.4: Basic binary operators.

Symbol	Command	Symbol	Command	Symbol	Command
$\pm$	\pm	$\diamond$	\diamond	$\wr$	\wr
$\mp$	\mp	$\Diamond$	\Diamond	$\setminus$	\setminus
$\div$	\div	$\triangle$	\triangle	$\amalg$	\amalg
$\times$	\times	$\triangle$	\bigtriangleup	$\dagger$	\dagger
$*$	\ast	$\triangledown$	\bigtriangledown	$\ddagger$	\ddagger
$\star$	\star	$\triangleleft$	\triangleleft	$\bigcirc$	\bigcirc
$\cdot$	\cdot	$\triangleright$	\triangleright	$\bigcap$	\bigcap
$\circ$	\circ	$\lhd$	\lhd	$\bigcup$	\bigcup
$\bullet$	\bullet	$\rhd$	\rhd	$\bigsqcup$	\bigsqcup
$\cap$	\cap	$\unlhd$	\unlhd	$\biguplus$	\biguplus
$\cup$	\cup	$\unrhd$	\unrhd	$\bigvee$	\bigvee
$\sqcap$	\sqcap	$\odot$	\odot	$\bigwedge$	\bigwedge
$\sqcup$	\sqcup	$\oplus$	\oplus	$\bigodot$	\bigodot
$\uplus$	\uplus	$\ominus$	\ominus	$\bigoplus$	\bigoplus
$\vee$	\vee	$\otimes$	\otimes	$\bigotimes$	\bigotimes
$\wedge$	\wedge	$\oslash$	\oslash		

Table 8.5: Basic relation operators.

Symbol	Command	Symbol	Command	Symbol	Command
$\leq$	\leq	$\in$	\in	$\neq$	\not=
$\ll$	\ll	$\notin$	\not\in	$\doteq$	\doteq
$\geq$	\geq	$\ni$	\ni	$\propto$	\propto
$\gg$	\gg	$\vdash$	\vdash	$\mid$	|
$\gtrless$	\gtreqqless	$\dashv$	\dashv	$\models$	\models
$\prec$	\prec	$\equiv$	\equiv	$\perp$	\perp
$\preceq$	\preceq	$\not\equiv$	\not\equiv	$\mid$	\mid
$\succ$	\succ	$\sim$	\sim	$\parallel$	\parallel
$\succeq$	\succeq	$\not\sim$	\not\sim	$\not\parallel$	\not\parallel
$\subset$	\subset	$\simeq$	\simeq	$\bowtie$	\bowtie
$\subseteq$	\subseteq	$\asymp$	\asymp	$\Join$	\Join
$\sqsubseteq$	\sqsubseteq	$\approx$	\approx	$\smile$	\smile
$\supset$	\supset	$\not\approx$	\not\approx	$\frown$	\frown
$\supseteq$	\supseteq	$\cong$	\cong	$\not<$	\not<
$\sqsupseteq$	\sqsupseteq	$\neq$	\neq	$\not>$	\not>

8.4. Simple equations

The very basic math-mode environment for producing an equation is **equation**, which is used for inserting a single equation. Table 8.1 shows three examples of the **equation** environment,

Table 8.6: A simple equation through **equation** environment.

LaTeX input	Output
\begin{equation} x^2 + y^2 = r^2 \label{eq:circ} \end{equation}	$x^2 + y^2 = r^2$ (8.1)
\begin{equation*} x^2 + y^2 = r^2 \end{equation*}	$x^2 + y^2 = r^2$
\begin{equation} x^2 + y^2 = r^2 \nonumber \end{equation}	$x^2 + y^2 = r^2$

where each equation is printed in a separate center-aligned line. As seen in the first example, by default the equation is assigned a serial number in ( ) on its right side. It is also shown that the equation can be labeled by the \label{} command for the purpose of referring it through the \ref{} command (similar to \ref{}, the \eqref{} command may also be used for referring an equation, which put the serial number of the equation in a pair of parentheses). On the other hand, if an equation is not to be numbered, either the \nonumber command may be inserted after the equation, or the equation may be produced through the **equation*** environment instead of the **equation** environment, as illustrated in the second and third examples of Table 8.6.

8.5. Array of equations

LaTeX provides a number of environments for producing an array of equations together, instead of producing each equation by a separate **equation** environment. Such environments along with their alignment structures are listed in Table 8.7 and their

Table 8.7: Environments used for producing arrays of equations.

Environment	Alignment structure
gather and **gather***	Gather equations without alignment.
eqnarray and **eqnarray***	Allow alignment about a single place only.
align and **align***	Allow alignment about a single place only.
alignat and **alignat***	Allow alignment about multiple places.
xalignat and **xxalignat**	Allow alignment about multiple places. Columns and margins are equally spaced in **xalignat**, while margin spacing is ignored in **xxalignat**.
array	Allows alignment at multiple places. It is to be nested in the **equation** environment.

applications are shown in Table 8.8. In these environments, each equation of an array, except the last one, is terminated by \\. The \\[vsize] command can also be used for providing extra vsize vertical space between two equations. The \intertext{atext} command can be used after \\ for inserting a few lines of texts (i.e., atext) in between two equations maintaining their alignments.

The patterns of aligning the equations of an array differ from environment to environment. The **eqnarray** and **eqnarray*** environments enclose the aligning place by a pair of **&** sign (e.g., **&=&**) or simply **&&** for aligning about an empty space. The **align** and **align*** environments use a single **&** on the left side of the aligning place (e.g., **&=**). Similarly, the **alignat**, **alignat***, **xalignat** and **xxalignat** environments (which allow alignment at multiple places) also use a single **&** on the left side of an aligning place, but with a provision for ending the current aligning place by another **&** before starting the alignment at the next place. The **alignat**, **alignat***, **xalignat** and **xxalignat** environments take the number of aligning places as a mandatory argument, e.g., \begin{alignat}{m} with m$\geqslant \frac{n}{2}+1$ if n is even and m$\geqslant \frac{n+1}{2}$ if n is odd, where n is the number of **&** used in an equation. On the other hand, the aligning process in the **array** environment is quite different. Similar to the **tabular** environment used for preparing a table (refer §6.1), the **array** environment creates aligning places through mandatory options of **l** for left alignment, **c** for centered and **r** for right alignment.

The starred forms of the environments (including **xxalignat** which acts like the starred form of **xalignat**) ignore the numbering to any equation, while their non-starred forms (including **xalignat**) assign an individual serial number to each equation. If required, numbering to any equation of an array can be eliminated using the \nonumber or \notag command. On the other

Table 8.8: Array of equations in different forms.

LaTeX input	Output

`\begin{gather}`
`  5x+  2y  =  x+  2z+  3\\`
`130x+  4z  =  y+  2\\`
`  43y+  57z  =  20x+  99`
`\end{gather}`
	$5x + 2y = x + 2z + 3$ (8.2)
	$130x + 4z = y + 2$ (8.3)
	$43y + 57z = 20x + 99$ (8.4)

`\begin{eqnarray}`
`5x+2y  &=&  x+2z+3  \label{eqn1}\\`
`130x+4z  &=&  y+2  \nonumber \\`
`43y+57z  &=&  20x+99  \label{eqn3}`
`\end{eqnarray}`
	$5x + 2y = x + 2z + 3$ (8.5)
	$130x + 4z = y + 2$
	$43y + 57z = 20x + 99$ (8.6)

`\begin{align}`
`5x+2y  &=  x+2z+3  \tag{See \eqref{eqn1}}\\`
`130x+4z  &=  y+2  \label{align2}\\[3mm]`
`43y+57z  &=  20x+99  \notag`
`\end{align}`
	$5x + 2y = x + 2z + 3$ (See (8.5))
	$130x + 4z = y + 2$ (8.7)
	$43y + 57z = 20x + 99$

`\begin{alignat*}{7}`
`  5x&+&  2y&&  &=&  x&+&  &&  2z&+&  3\\`
`\intertext{Please notice the alignment ...}`
`130x&+&  &&  4z&=&  &&  y&+&  &&  2\\`
`  &&  43y&+&  57z&=&  20x&+&  &&  &&  99`
`\end{alignat*}`
	$5x + 2y = x + 2z + 3$
	Please notice the alignment made about each '+' and '=' signs of these equations.
	$130x + 4z = y + 2$
	$43y + 57z = 20x + 99$

`\begin{xxalignat}{7}`
`  5x&+&  2y&&  &=&  x&+&  &&  2z&+&  3\\`
`130x&+&  &&  4z&=&  &&  y&+&  &&  2\\`
`  &&  43y&+&  57z&=&  20x&+&  &&  &&  99`
`\end{xxalignat}`
	$5x + 2y = x + 2z + 3$
	$130x + 4z = y + 2$
	$43y + 57z = 20x + 99$

`\begin{equation}`
`\left.\begin{array}{rrrrrrrrrrrr}`
`  5x&+&  2y&&  &=&  x&+&  &&  2z&+&  3\\`
`130x&+&  &&  4z&=&  &&  y&+&  &&  2\\`
`  &&  43y&+&  57z&=&  20x&+&  &&  &&  99`
`\end{array}\right\}`
`\end{equation}`
	$\left.\begin{array}{rrrrrrrrrrrr} 5x+ & 2y & = & x+ & 2z+ & 3 \\ 130x+ & & 4z= & & y+ & 2 \\ & 43y+ & 57z= & 20x & & 99 \end{array}\right\}$ (8.8)

hand, since the **array** environment is nested in the **equation** environment, it is treated as a single equation and hence the entire array of equations is assigned a single serial number as a whole.

9. Bibliography with BIBTEX

An elegant bibliographic list can be produced in a document by incorporating the BibTeX program with LaTeX. It prints a reference only if it is cited somewhere in the document. Moreover, the BibTeX program follows certain structures for different types of references according to the chosen bibliography style as discussed in §9.2.

9.1. Preparation of BIBTEX compatible reference database

The entry of a reference in the BibTeX program consists of three mandatory parts – (1) type of the reference, (2) a user-defined citation key which can be used for citing the reference, and (3) detail of the reference.

As shown in Table 9.1, there are around 14 types of defined references, which are **article** (articles in journals or magazines), **book** (books), **booklet** (booklet type references), **inbook** (chapters or parts of books), **incollection** (parts of a book with separate titles), **inproceedings** (articles in conference proceedings), **conference** (articles in conference proceedings), **manual** (technical documentations), **mastersthesis** (Master theses), **phdthesis** (Ph.D theses), **misc** (uncommon references), **proceedings** (proceedings of an event), **techreport** (technical reports or working papers), and **unpublished** (unpublished references).

A reference-type command is preceded by **@** and it takes a mandatory argument, e.g., **@article**`{ckey, rf1, rf2, ...}`, where `ckey` is the mandatory citation key, and `rf1`, `rf2`, etc., are other mandatory and optional fields detailing the reference.

As shown in Table 9.1, a reference-type command takes some out of around 21 reference fields, which are **address**, **author**, **booktitle**, **chapter**, **edition**, **editor**, **howpublished**, **institution**, **journal**, **month**, **note**, **number**, **organization**, **pages**, **publisher**, **school**, **series**, **title**, **type**, **volume**, and **year**. These fields can be entered in any order, which will be arranged automatically according to the chosen bibliography style. There is no harm if an extra field is inserted or an acceptable field is left blank. A redundant field or a field with no data is automatically skipped by BibTeX.

Table 9.1: Types and fields of references under BibTeX program.

	article	book	booklet	inbook	incollection	inproceedings, conference	manual	mastersthesis, phdthesis	misc	proceedings	techreport	unpublished
address	×	O	O	O	O	O	O	O	×	O	O	×
author	M	M_1	O	M_1	M	M	O	M	O	×	M	M
booktitle	×	×	×	×	M	M	×	×	×	×	×	×
chapter	×	×	×	M_2	O	×	×	×	×	×	×	×
edition	×	O	×	O	O	×	O	×	×	×	×	×
editor	×	M_1	×	M_1	O	O	×	×	×	O	×	×
howpublished	×	×	O	×	×	×	×	×	O	×	×	×
institution	×	×	×	×	×	×	×	×	×	×	M	×
journal	M	×	×	×	×	×	×	×	×	×	×	×
month	O	O	O	O	O	O	O	O	O	O	O	O
note	O	O	O	O	O	O	O	O	O	O	O	M
number	O	O_1	×	O_1	O_1	O_1	×	×	×	O_1	O	×
organization	×	×	×	×	×	O	O	×	×	O	×	×
pages	O	×	×	M_2	O	O	×	×	×	×	×	×
publisher	×	M	×	M	M	O	×	×	×	O	×	×
school	×	×	×	×	×	×	×	M	×	×	×	×
series	×	O	×	O	O	O	×	×	×	O	×	×
title	M	M	M	M	M	M	M	M	O	M	M	M
type	×	×	×	O	O	×	×	O	×	×	O	×
volume	O	O_1	×	O_1	O_1	O_1	×	×	×	O_1	×	×
year	M	M	O	M	M	M	O	M	O	M	M	O

M → mandatory field M_1 → one of them is mandatory
M_2 → either one or both are mandatory
O → optional field O_1 → one of them (optional) ×→ not required

Data of all the fields of a reference-type command are processed in text-mode. Hence, accented and special characters are to be put in proper way, e.g., '`Jos\'e`' for producing '`José`' or '`Hungerl$\ddot{\mathrm{a}}$nder`' for producing '`Hungerländer`'. Data of a field can be inserted either in a pair of quotes or curly braces, e.g., `title="A Practical Guide to \LaTeX"` or `title={A Practical Guide to \LaTeX}`. Further processes for entering data in the argument of a reference-type command are explained below:

- *Citation key*: It is a combination of any number of alphabets, numerals and/or signs without any gap between two characters, e.g., `Even-etal-1976` or `Even+:1976`.
- **address**: It could be the city or country of a publisher, venue of a conference, address of an institution or school, or URL of a webpage. A URL may be inserted through the `\url{}` command defined in the **url** package.
- **author**: The name of an author is processed in two parts only, the first name and the surname.
  1. In the case of all words capitalized, the last word is treated as the surname and rest as the first name.
  2. In the presence of any non-capitalized word, all the remaining words starting from the first non-capitalized one are treated as the surname and rest as the first name.
  3. If a surname contains multiple words, those may be inserted either in curly braces or at the starting with a comma after the last word.
  4. In the case of a multi-author reference, the names of every two authors are to be separated by the word '`and`' (not by a comma as generally done).
  5. If a list is to be truncated with the name(s) of the first or few author(s), '`and others,`' may be added after that(those) name(s), which is converted to '`et al.`'.
- **booktitle**: Title of the book or proceedings.
- **chapter**: Serial number of the referred unit/chapter.
- **edition**: Edition number, e.g., '`Second`' or '`2nd`'.
- **editor**: Names of the editors (in the same way as **author**).
- **howpublished**: Type of publication.
- **institution**: Name of the publishing Institute.
- **journal**: Name of the journal.
- **month**: Month of publication, e.g., '`Jan.`' or '`January`'.
- **note**: A short note on the document, e.g., the abstract.
- **number**: Issue or serial number of the publication.
- **organization**: Name of the organizing or sponsoring body.
- **pages**: Serial number of the referred pages, e.g., '`24--35`'.
- **publisher**: Name of the publisher.
- **school**: Name of the thesis submitting Institute.
- **series**: Name of the publishing series.
- **title**: Title of the reference.
- **type**: Type of the reference, e.g., '`Ph.D thesis`'.
- **volume**: Volume number of the publication.
- **year**: Year of publication, e.g., `2018`.

According to above, an illustrative bibliographic reference database file is shown in Table 9.2, where two fields of a reference (including the citation key) are separated by a comma.

9.2. Standard bibliographic styles of LaTeX

The formatting of bibliographic references in the BibTeX program is controlled by an associated bibliographic style, which is defined through the `\bibliographystyle{astyle}` command with `astyle` as a bibliography style as listed in Table 9.3. The `\bibliographystyle{astyle}` command is followed by the `\bibliography{dbib}` command, where `dbib` is the name of the bibliography database file without its `bib` extension. For example, the following are the commands to load the bibliography database file '`mybib2.bib`' of Table 9.2 in **plain** style:

```
\bibliographystyle{plain}
\bibliography{mybib2}
```

Under all the bibliography styles given in Table 9.3, a reference is cited through the `\cite{ckey}` command, where `ckey` is the citation key of the reference. Multiple references can also be cited through a single `\cite{}` command separating two citation keys by a comma, e.g., `\cite{Datta-1998, Even-etal-1976}`.

Table 9.2: BibTeX compatible bibliographic reference database.

```
% mybib2.bib
@article{Datta-Figueira-2013,
  author = {Dilip Datta and Jos\'e Rui Figueira},
  title = {{A real-integer-discrete-coded differential
evolution}},
  journal = {Applied Soft Computing},
  volume = {13},
  number = {9},
  pages = {3884--3893},
  year = {2013}
}

@book{Deb-2001,
  author = {Kalyanmoy Deb},
  title = {{Multi-Objective Optimization using Evolutionary
Algorithms}},
  publisher = {John Wiley \& Sons Ltd.},
  address = {Chichester, England},
  year = {2001}
}

@inproceedings{Burke-etal-1996,
  author = {Edmund Burke and Dave Elliman and Peter Ford
Rupert Weare},
  title = {{Examination Timetabling in British Universities -
A Survey}},
  booktitle = {Proceedings of Practice and Theory of
Automated Timetabling},
  publisher = {Springer},
  series = {Lecture Notes in Computer Science (LNCS)},
  editor = {Edmund K. Burke and Peter Ross},
  year = {1996},
  volume = {1153},
  pages = {76--90}
}

@mastersthesis{Datta-1998,
  author = {Dilip Datta},
  title = {{Optimal Shape Design System for Plates under
Dynamic Loads}},
  school = {Indian Institute of Technology, Delhi},
  month = {December},
  year = {1998},
  note = {Master thesis}
}

@techreport{Colorni-etal-1992,
  author = {Alberto Colorni and Marco Dorigo and Vittorio
Maniezzo},
  title = {{A Genetic Algorithm to Solve the Timetable
Problem}},
  number = {90-060 revised},
  institution = {Politecnico di Milano, Italy},
  year = {1992}
}
```

Citations are marked in the contents of a document by the identifiers of the cited references, e.g., `[1]`, `[2]`, ..., or `[Deb01]`, `[DF13]`, ..., or `[Datta and Figueira, 2013]`, `[Deb, 2001]`, ..., as shown in Table 9.3.

Only those references, cited in the document, are printed in the bibliographic reference list under the heading '`References`'. If a reference is to be printed in the bibliography list without citing it in the document, the `\nocite{}` command may be used somewhere in the document, e.g., `\nocite{Datta-1998}` for printing the reference under the citation key '`Datta-1998`' without citing it in the document. On the other hand, the `\nocite{*}` command may be used for listing all the references of a database file without citing even a single one.

Table 9.3: Some standard bibliographic styles of LaTeX.

LaTeX style	Function
plain	References are listed in alphabetic order of the surnames (last or family names) of authors, and labeled by Arabic numerals in `[]`, e.g., the first two references of Table 9.2 will be produced as follows: [1] Dilip Datta and José Rui Figueira. A real-integer-discrete-coded differential evolution. *Applied Soft Computing*, 13(9):3884–3893, 2013. [2] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons Ltd., Chichester, England, 2001.
unsrt	Same with plain, except that the references are listed in order of their citations in the document.
alpha	A reference is labeled by an identifier generated from the surnames of the authors and the year of publication, and the references are listed in alphabetic order of their identifiers, e.g., the first two references of Table 9.2 will be produced as follows: [Deb01] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons Ltd., Chichester, England, 2001. [DF13] Dilip Datta and José Rui Figueira. A real-integer-discrete-coded differential evolution. *Applied Soft Computing*, 13(9):3884–3893, 2013.
abbrv	Same with plain, except that a reference is made compact by abbreviating the given (or first and middle) names of authors, e.g., the first two references of Table 9.2 will be produced as follows: [1] D. Datta and J. R. Figueira. A real-integer-discrete-coded differential evolution. *Applied Soft Computing*, 13(9):3884–3893, 2013. [2] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons Ltd., Chichester, England, 2001.
acm	Same with plain, but the surname of an author is printed first in small capital letters, followed by the abbreviated given name, e.g., the first two references of Table 9.2 will be produced as follows: [1] Datta, D., and Figueira, J. R. A real-integer-discrete-coded differential evolution. *Applied Soft Computing 13*, 9 (2013), 3884–3893. [2] Deb, K. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons Ltd., Chichester, England, 2001.
apalike	The surname of an author is printed first, followed by the abbreviated given name, and a reference is labeled by an identifier generated from the surnames of authors and the year of publication, e.g., the first two references of Table 9.2 will be produced as follows: [Datta and Figueira, 2013] Datta, D. and Figueira, J. R. (2013). A real-integer-discrete-coded differential evolution. *Applied Soft Computing*, 13(9):3884–3893. [Deb, 2001] Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons Ltd., Chichester, England.

9.3. Compiling BIBTeX based LaTeX input file

If bibliography is generated through the BibTeX program, a LaTeX file is to be compiled through the following 4 lines of commands:

```
$ latex myarticle
$ bibtex myarticle
$ latex myarticle
$ latex myarticle
```

where `myarticle` is the name of the LaTeX input file without its `tex` extension. The bibtex command compiles the bibliography file included in `myarticle.tex`. The last two latex commands link the BibTeX generated bibliographic ref-

erences with LaTeX. The above commands will produce five files, namely `myarticle.aux`, `myarticle.log`, `myarticle.dvi`, `myarticle.bbl` and `myarticle.blg`. As mentioned in §1.2, the `myarticle.dvi` file can be viewed in a document viewer or can be used to produce a `pdf` file.

10. Article Preparation

Templates of articles for publishing in journals, proceedings, magazines, etc., vary from publisher to publisher. Many publishers provide their own templates for maintaining uniformity in a volume, and an author needs just to insert the contents of an article in the given template. If not given, authors can prepare articles in their own templates. A number of such LaTeX based standard templates are discussed here.

An article can be prepared in the document-class of **article** or **amsart**. Generally an article starts with a title and the list of author(s), which are inserted as the arguments of the `\title{}` and `\author{}` commands respectively. These commands are activated using the `\maketitle` command in the **document** environment before inserting any content of the article. The `\title{}` and `\author{}` commands can be used either in the preamble or even in the **document** environment, but must be before the `\maketitle` command. Following the `\maketitle` command, the abstract of the article is inserted in the **abstract** environment. Then the actual contents of the article are inserted through a series of standard formatting, such as `\section{}`, `\subsection{}`, `\subsubsection{}`, `\paragraph{}`, `\subparagraph{}` commands, as well as other applicable commands and environments discussed up to the previous Section.

The general format of an article, in both the document-classes of **article** and **amsart**, are shown in Tables 10.1 and 10.2 respectively, where the differences in the outputs of the two document-classes are clearly visible. The document-class **article** by default prints the compilation date of the article (which is prevented in Table 10.1 through the `\date{}` command with empty argument).

10.1. List of authors

Tables 10.1 and 10.2 show only one author in the article, which is printed center-alignment. When the number of authors is more than one, these may be printed one below another, side-by-side, or in any other user-defined format. A format for two authors, printed one below another, is shown Table 10.3. Another format is shown in Table 10.4, where two authors are printed side-by-side through the **tabular** environment with two columns. Also, the contents of each column are center-aligned and two columns are separated by extra 30 mm space through the `@{\extracolsep{30mm}}` command. A third format is shown in Table 10.5, where the detail of an author is printed at the bottom of the page through the `\thanks{}` command after the name of each author. The same effect can be obtained through the `\footnote{}` command also, instead of the `\thanks{}` command. The `\and` command is used between the names of the two authors to separate them by a big gap in the output.

Table 10.1: Article in the document-class **article**.

LaTeX input	Output

`% myarticle.tex (in 'article')`
`\documentclass[12pt]{article}`
`\date{}`
`\title{My First Article in \LaTeX}`
`\author{Author's Name and Address}`
`\begin{document}`
`\maketitle`
`%`
`\begin{abstract}`
`The article explains ...`
`\end{abstract}`
`%`
`\section{First Section}`
`First level of numbered section.`
`\subsection{First subsection}`
`Second level of numbered section.`
`\subsubsection{First sub-subsection}`
`Third and last level of numbered section.`
`\section{Second Section}`
`Texts of the second section ...`
`%`
`\end{document}`
	My First Article in LaTeX
	Author's Name and Address
	Abstract
	The article explains ...
	1 First Section
	First level of numbered section.
	1.1 First subsection
	Second level of numbered section.
	1.1.1 First sub-subsection
	Third and last level of numbered section.
	2 Second Section
	Texts of the second section ...
	1

Table 10.2: Article in the document-class **amsart**.

LaTeX input	Output

`% myarticle.tex (in 'amsart')`
`\documentclass[12pt]{amsart}`
`\title{My First Article in \LaTeX}`
`\author{Author's Name and Address}`
`\begin{document}`
`\maketitle`
`%`
`\begin{abstract}`
`The article explains ...`
`\end{abstract}`
`%`
`\section{First Section}`
`First level of numbered section.`
`\subsection{First subsection}`
`Second level of numbered section.`
`\subsubsection{First sub-subsection}`
`Third and last level of numbered section.`
`\section{Second Section}`
`Texts of the second section ...`
`%`
`\end{document}`
	MY FIRST ARTICLE IN LaTeX
	AUTHOR'S NAME AND ADDRESS
	Abstract. The article explains ...
	1. First Section
	First level of numbered section.
	1.1. **First subsection.** Second level of numbered section.
	1.1.1. *First sub-subsection.* Third and last level of numbered section.
	2. Second Section
	Texts of the second section ...
	1

10.2. Title and abstract on separate pages

Some publishers may ask to produce title, list of authors and abstract of an article on separate pages, particularly for the re-

Table 10.3: Authors in articles one below another.

LaTeX input	Output

`\author`
`{`
`  {\bf 1st author's name}\\`
`  Affiliation\\`
`  Address\\[2mm]`
`  %`
`  {\bf 2nd author's name}\\`
`  Affiliation\\`
`  Address`
`}`
	1st author's name
	Affiliation
	Address
	
	2nd author's name
	Affiliation
	Address

Table 10.4: Authors side-by-side through the `tabular` environment.

LaTeX input	

	`\author`
	`{`
	`  \begin{tabular}[t]{c@{\extracolsep{30mm}}c}`
	`  {\it Author-1} & {\it Author-2}\\`
	`  Affiliation & Affiliation\\`
	`  Address & Address\\`
	`  e-mail & e-mail\\`
	`  \end{tabular}`
	`}`

Output	*Author-1* Affiliation Address e-mail *Author-2* Affiliation Address email

Table 10.5: Author details at the bottom of a page through `\thanks{}` command.

LaTeX input	Output

`\documentclass[12pt]{article}`
`\date{}`
`\title{My First Article in \LaTeX}`
`\author`
`{`
`  Mr.\,X\thanks{X's Address}`
`  \and`
`  Mr.\,Y\thanks{Y's Address}`
`}`
`\begin{document}`
`\maketitle`
`%`
`\begin{abstract}`
`The article explains ...`
`\end{abstract}`
`%`
`\section{Introduction}`
`Introduction to the problem ...`
`%`
`\end{document}`
	My First Article in LaTeX
	
	Mr. X* Mr. Y†
	
	Abstract
	The article explains ...
	
	1 Introduction
	Introduction to the problem ...
	
	*X's Address
	†Y's Address
	
	1

view purpose. These can be achieved by using the `titlepage` and `abstract` options in the `\documentclass[]{}` command, i.e., as `\documentclass[titlepage,abstract]{article}`. The `titlepage` option instructs `\maketitle` to produce the title and list of authors on a separate page, while the `abstract` option instructs to produce the abstract on another separate page.

10.3. Articles in multiple columns

Many publishers want articles to be produced in multiple columns. The `twocolumn` option in the `\documentclass[]{}` command produces an article in two columns. In the document-class `article`, as shown in Table 10.6, the title and list of authors

Table 10.6: Article in two columns through the `twocolumn` option.

LaTeX input	

	`\documentclass[a4paper,12pt,twocolumn]{article}`
	`\date{}`
	`\title{My First Article in \LaTeX}`
	`\author{Author's Name and Address}`
	`\begin{document}`
	`\maketitle`
	`%`
	`\begin{abstract}`
	`Abstract of the article ...`
	`\end{abstract}`
	`%`
	`\section{Introduction}`
	`Introduction to the work ...`
	`\end{document}`

Output	My First Article in LaTeX Author's Name and Address **Abstract** Abstract of the article ... Abstract of the article ... Abstract of the article ... Abstract of the article ... **1 Introduction** Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ...

are printed in single-column, and the abstract and other contents of the article are printed in two columns. In the case of the document-class `amsart`, however, the title and list of authors are also printed in two columns.

Sometime the title, author and abstract are to be printed in a single column, while the rest of the article in two columns. In that case, instead of the `twocolumn` option in `\documentclass[]{}`, the `\twocolumn[]` command may be used. As shown in Table 10.7, `\twocolumn[]` is used after `\begin{document}`, putting the `\maketitle` command and the `abstract` environment in `[]` of `\twocolumn[]`. After `\end{abstract}`, the `\vspace{1.0cm}` command is used in Table 10.7 to leave 1.0 cm vertical blank space before starting the two-column mode.

11. Thesis preparation

Preparation of an academic thesis (also a book or a report) is similar with that of an article discussed in §10. The only difference is that a thesis contains a number of chapters, where each chapter is like an independent article.

Table 10.7: Article in two columns through the \twocolumn[] command.

LaTeX input

```
\documentclass[a4paper,12pt]{article}
\date{}
\title{My First Article in \LaTeX}
\author{Author's Name and Address}
\begin{document}
\twocolumn
[
    \maketitle
    \begin{abstract}
    Abstract of the article ...
    \end{abstract}
    \vspace{1.0cm}
]
\section{Introduction}
Introduction to the work ...
\end{document}
```

Output

My First Article in LaTeX

Author's Name and Address

Abstract

Abstract of the article ... Abstract of the article ... Abstract of the article ... Abstract of the article ... Abstract of the article ... Abstract of the article ...

1 Introduction

Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ... Introduction to the work ...

11.1. Template of a thesis

A thesis can be prepared through the document-class of **book** or **report**. Since a thesis usually contains a large number of pages, practically it will be difficult to work with a single file accommodating all the contents of the entire thesis. Hence, as a convenient way, a thesis is generally prepared in a number of input files of smaller sizes. Usually each part of a thesis (such as the cover page, title page, abstract, certificate, acknowledge, dedication, chapters, appendix, and bibliography) is prepared in a separate input file and then all the individual files are compiled together to produce a single output file as the final thesis. Moreover, the preamble (i.e., the required packages and other global settings) may also be prepared in a separate input file. The input file containing the list of bibliographic references is named with `bib` extension, say `mybib.bib`. All other input files are named with `tex` extension, such as, `preamble.tex`, `title.tex`, `abstract.tex`, `chapter1.tex`, `chapter2.tex`, etc. Finally, all the individual input files are linked to a single root file, say `mythesis.tex`, compilation of which produces the final thesis. Note that neither the \documentclass[]{} command nor the **document** environment is required in the individual input files. The \documentclass[]{} command is included in the preamble file, and the **document** environment is created in the root file in which these individual files are linked.

The input files (other than the bibliography file) can be linked to the root file either through the \input{} or \include{} command with the name of an input file (without its extension) as the argument of the command. The bibliography file is included through the \bibliography{} command following the \bibliographystyle{} command. Such a sample root file of a thesis is shown in Table 11.1, whose different segments are as follows:

Table 11.1: Root file linking individual input files of a thesis.

```
% File name: mythesis.tex
\input{preamble}
\begin{document}
    % Cover page, title, abstract, certificate, acknowledge, and dedication
    \thispagestyle{empty}
        \include{coverpage} \clearpage
    \pagenumbering{roman}    % Page numbering in Roman numerals
        \addcontentsline{toc}{chapter}{Title}
            \thispagestyle{empty} \include{title} \clearpage
        \addcontentsline{toc}{chapter}{Abstract}
            \thispagestyle{empty} \include{abstract} \clearpage
        \addcontentsline{toc}{chapter}{Certificate}
            \thispagestyle{empty} \include{certif} \clearpage
        \addcontentsline{toc}{chapter}{Acknowledgement}
            \thispagestyle{empty} \include{acknowl} \clearpage
        \addcontentsline{toc}{chapter}{Dedication}
            \thispagestyle{empty} \include{dedicate} \clearpage
        % Contents, List of Tables and List of Figures
        \addcontentsline{toc}{chapter}{Contents}
            \thispagestyle{empty} \tableofcontents \clearpage
        \addcontentsline{toc}{chapter}{List of Tables}
            \thispagestyle{empty} \listoftables \clearpage
        \addcontentsline{toc}{chapter}{List of Figures}
            \thispagestyle{empty} \listoffigures \clearpage
    % Main matters
    \pagenumbering{arabic}    % Page numbering in Arabic numerals
        % Chapters
        \include{chap_introduction}
        \include{chap_literature}
        \include{chap_problem}
        \include{chap_formulation}
        \include{chap_result}
        \include{chap_conclusion}
        % Appendix
        \appendix
        \include{app_derivation}
        \include{app_proof}
        % Bibliography
        \addcontentsline{toc}{chapter}{Bibliography}
            \bibliographystyle{plain}
            \bibliography{mybib}
\end{document}
```

- ▷ At the very beginning, the preamble file (`preamble.tex`) is included through the \input{preamble} command, and then all other '`.tex`' files (without extension) are included in the **document** environment through the \include{} command.
- ▷ The BibTeXformatted bibliography file (`mybib.bib`) is inserted through the \bibliography{mybib} command, preceded by a bibliography style through the \bibliographystyle{} command, say \bibliographystyle{plain}.
- ▷ The \thispagestyle{empty} command is used in some places for not assigning any page number to those pages.
- ▷ The \clearpage command is used in many places for starting the following unit on the next odd numbered page.

- ▷ The inclusions of the unnumbered units are preceded by the \addcontentsline{}{}{} command for including them in the Contents list. The arguments of \addcontentsline{}{}{} are respectively the location where the unit is to be entered (toc for Contents list), the type how the unit is to be treated (chapter) and the name of the unit to be printed in the Contents list.
- ▷ The \tableofcontents, \listoftables and \listoffigures commands are used for automatically generating three lists – Contents list of sectional units, List of Tables and List of Figures, respectively.
- ▷ The \appendix command is used for printing the included following chapters as appendices.
- ▷ The \pagenumbering{roman} command is used for numbering the following pages in Roman numerals, while the \pagenumbering{arabic} command is used for numbering its following pages in Arabic numerals.

11.2. Compilation of thesis

Since a thesis is generally composed of a number of chapters, one or more appendices, and bibliography, it (the root file mythesis.tex as given in Table 11.1) is to be compiled using the following commands:

```
$ latex mythesis
$ bibtex mythesis
$ latex mythesis
$ latex mythesis
```

The latex command in the first line compiles mythesis.tex as well as all other '.tex' files included in mythesis.tex. It also generates the output in the form of mythesis.dvi. However, the bibliography remains uncompiled. It is required to be compiled separately, which is done through the bibtex command in the second third line. Then the latex command is repeated in the third line for linking all the compilation. As the last step, the same command is repeated once again in the fourth line for producing 'mythesis.dvi' as the complete and final output file.

12. Slide Preparation

The LaTeX platform is applicable for preparing slides also, which can be presented like those prepared in popularly known Microsoft PowerPoint package. In LaTeX, slides are widely prepared through the document classe of beamer.

12.1. Frames in presentation

In the document-class beamer, a presentation consists of a number of frames (or slides). A frame is created either by the \frame[]{} command or the frame environment as \begin{frame}[]...\end{frame}. A frame generally consists of some or all of the following eight components:

(1) Headline and footline: These are generated automatically for displaying presentation related various information.

(2) Sidebars: Sidebars are generated automatically for displaying mainly the table of contents of the presentation.

(3) Navigation bars: Navigation bars are produced for displaying the status of presentation, as well as to jump to a particular frame by clicking on the corresponding link.

(4) Navigation symbols: Eight number of default navigation symbols are provided in the bottom right corner of every slide in order to jump other frames or slides.

(5) Logo: A logo can be printed globally in all the frames through the \logo{} command in the preamble.

(6) Frame title: A title and a subtitle can be assigned to a frame using \frametitle{} and \framesubtitle{} in the mandatory argument of \frame[]{} or inside the frame environment.

(7) Background: Each frame has a background, which consists of a background canvas and the main background.

(8) Frame contents: The main contents of a frame could be LaTeX supported any text, including the frame title and subtitle. The contents of a frame are inserted in the mandatory argument of \frame[]{} or inside the frame environment. By default the contents of a frame are vertically center aligned, which can be changed by the option t (top alignment) or b (bottom alignment), e.g., \frame[t]{} or \begin{frame}[t] for top alignment.

12.2. Sectional units in presentation

The frames of a presentation can be put under different sections and subsections produced through the \section[]{} and \subsection[]{} commands. Unlike in standard LaTeX, \section[]{} and \subsection[]{} here do not create any heading at their positions, rather they add entries in the table of contents and navigation bars. Generally the full headings under the mandatory arguments of \section[]{} and \subsection[]{} are added in the table of contents, while the short headings under their optional arguments are added in navigation bars.

12.3. Presentation structure

A simple presentation input file consisting of six frames under the JuanLesPins presentation theme (§12.5 discusses themes) is shown in Table 12.1 and its output column-wise in Table 12.2. Just to show their applications, frames are created through both the \frame[]{} command and frame environment. The input file is started with \documentclass{beamer}, followed by the JuanLesPins presentation theme loaded through \usetheme{}. The other major component in the preamble is the title page related commands (refer §12.4 for detail). Then the frames, containing different components of a presentation under various sectional units, are prepared in the document environment. The title page is generated in the very first frame. The next frame, producing the table of contents through \tableofcontents, is put under \section*{} for excluding its entry in the table of contents. Then the main contents of a presentation are inserted in intermediate frames, as shown in Table 12.1 producing frames 3

Table 12.1: A simple presentation input file.

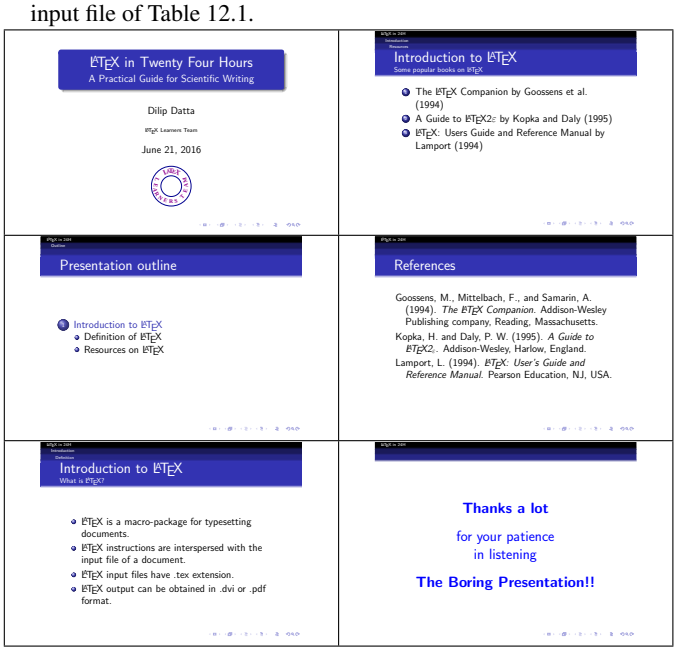
```
\documentclass{beamer}
\usetheme{JuanLesPins}
% Components of the title page
\title[\LaTeX\ in 24H]{\LaTeX\ in Twenty Four Hours}
\subtitle{A Practical Guide for Scientific Writing}
\author[D. Datta]{Dilip Datta}
\institute[\LaTeX-LT]{\LaTeX\ Learners Team}
\date[L24H :: 21-06-2016]{June 21, 2016}
\titlegraphic{\includegraphics[width=20mm]{logo_LA}}
%
\begin{document}
% Frame 1
\frame[plain]{\titlepage}
% Frame 2
\section*{Outline}
\frame[t]{ \frametitle{Presentation outline} \tableofcontents }
% Frames 3 and 4
\section[Introduction]{Introduction to \LaTeX}
\subsection[Definition]{Definition of \LaTeX}
\frame[t]
{    \frametitle{Introduction to \LaTeX}
     \framesubtitle{What is \LaTeX?}
     \begin{itemize}
     \item \LaTeX\ is a macro-package for ...
     \item \LaTeX\ instructions are interspersed ...
     \item \LaTeX\ input files have .tex extension.
     \item \LaTeX\ output can be obtained in ...
     \end{itemize}
}
\subsection[Resources]{Resources on \LaTeX}
\begin{frame}[t]
  \frametitle{Introduction to \LaTeX}
  \framesubtitle{Some popular books on \LaTeX}
  \begin{enumerate}
  \item The \LaTeX\ Companion by \citet{Goossens-etal-1994}
  \item A Guide to \LaTeX2$_\varepsilon$ by \citet{KopDal-97}
  \item \LaTeX: User's Guide and Reference Manual by ...
  \end{enumerate}
\end{frame}
% Frame 5
\section*{}
\begin{frame}[t]
  \frametitle{References}
  \bibliographystyle{apalike} \bibliography{lswbib}
\end{frame}
% Frame 6
\section*{}
\begin{frame}
  \begin{center}
  \Large{\bf\textcolor{blue}{Thanks a lot}}\\[5mm] ...
\end{center}
\end{frame}
\end{document}
```

and 4 under one section and two subsections. Finally, the bibliographic reference list and thanks giving may be inserted in the ending frames. These two frames are prepared under `\section*{}` without any argument so as to skip their entry in the table of contents as well as in headline/footline and sidebars.

12.4. Title page

The title page of a presentation is produced through the `\titlepage` command, which in Table 12.1 is put in `\frame[]{}` for printing the title page in a frame (Frame 1). The **plain** option is used here for omitting headline/footline and sidebars

Table 12.2: Slides under the **JuanLesPins** presentation theme for input file of Table 12.1.



in the frame. The title page generally contains sequentially a title, a subtitle, list of authors, affiliations of the authors, presentation date, and a symbolic affiliation, which are generated in the preamble through the `\title[]{}`, `\subtitle{}`, `\author[]{}`, `\institute[]{}`, `\date[]{}` and `\titlegraphic{}` commands respectively.

12.5. Appearance of a presentation (BEAMER themes)

It is always desirable to make a presentation attractive as much as possible. The appearance of a presentation in the **beamer** document-class can be controlled by five types of themes, which are presentation theme, color theme, font theme, inner theme, and outer theme. A presentation theme generally controls every single detail of the appearance of a presentation. Since every presentation theme uses a default set of other four themes, normally no other theme is required to be specified if a presentation theme is chosen. In order to alter the default setting of a presentation theme, still separate color, font, inner, or outer theme may be used as per requirement or choice.

The above five themes are loaded in the preamble as `\usetheme{tname}`, `\usecolortheme{tname}`, `\usefonttheme{tname}`, `\useinnertheme{tname}`, and `\useoutertheme{tname}`, where `tname` is the name of the chosen presentation/color/font/inner/outer theme.

The following are the available various themes:

1. Presentation themes: **default**, **boxes**, **Bergen**, **Boadilla**, **Madrid**, **AnnArbor**, **CambridgeUS**, **EastLansing**, **Pittsburgh**, **Rochester**, **Antibes**, **JuanLesPins**, **Montpellier**, **Berkeley**, **PaloAlto**, **Goettingen**, **Marburg**, **Hannover**, **Berlin**, **Ilmenau**, **Dresden**, **Darmstadt**, **Frankfurt**, **Singapore**, **Szeged**, **Copenhagen**, **Luebeck**, **Malmoe**, **Warsaw**, etc.

2. Color themes: **default**, **sidebartab**, **structure**, **albatross**, **beetle**, **crane**, **dove**, **fly**, **monarca**, **seagull**, **wolverine**, **beaver**, **spruce**, **lily**, **orchid**, **rose**, **whale**, **seahorse**, **dolphin**, etc.
3. Font theme: **default**, **serif**, **structurebold**, **structureitalicserif**, **structuresmallcapsserif**, etc.
4. Inner themes: **default**, **circles**, **rectangles**, **rounded**, **inmargin**, etc.
5. Outer themes: **default**, **infolines**, **miniframes**, **smoothbars**, **sidebar**, **split**, **shadow**, **tree**, **smoothtree**, etc.