

# Basic L<sup>A</sup>T<sub>E</sub>X Typesetting

William J. Turner

January 15, 2022

### **Abstract**

This paper is a short introduction to using L<sup>A</sup>T<sub>E</sub>X at Wabash College. It introduces the basics of a L<sup>A</sup>T<sub>E</sub>X document.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Getting Started</b>	<b>3</b>
2.1	PCTeX . . . . .	4
2.2	TeXworks . . . . .	4
2.3	L <sup>A</sup> T <sub>E</sub> X on Your Personal Computer . . . . .	5
2.4	L <sup>A</sup> T <sub>E</sub> X Online . . . . .	5
<b>3</b>	<b>Basic L<sup>A</sup>T<sub>E</sub>X</b>	<b>6</b>
3.1	Commands and Comments . . . . .	6
3.2	File Layout . . . . .	6
3.3	Titles, Authors, Dates . . . . .	6
3.4	Sections . . . . .	7
3.5	Words, Paragraphs, and Whitespace . . . . .	7
3.6	Quotation Marks . . . . .	8
3.7	Tables . . . . .	8
<b>4</b>	<b>Mathematics</b>	<b>9</b>
4.1	Text and Display Styles . . . . .	9
4.2	AMS L <sup>A</sup> T <sub>E</sub> X . . . . .	9
4.3	Superscripts and Subscripts . . . . .	10
4.4	Binary Relations and Negations . . . . .	10
4.5	Function Names . . . . .	10
4.6	Integrals, Sums, etc . . . . .	10
4.7	Fractions . . . . .	11
4.8	Automatic Sizing of Bracket Symbols . . . . .	11
4.9	Matrices . . . . .	13
4.10	Other Useful Commands . . . . .	13
<b>5</b>	<b>Theorems and Proofs Using the amsthm Package</b>	<b>14</b>
5.1	Proofs . . . . .	14
5.2	Theorem-Type Environments . . . . .	15
5.3	Changing Numbers . . . . .	15
5.4	Theorem Styles . . . . .	16
5.5	Specific Names . . . . .	16
5.6	Examples . . . . .	16
5.6.1	Divisibility . . . . .	17
5.6.2	Greatest Common Divisor . . . . .	18
<b>6</b>	<b>More Advanced Topics</b>	<b>19</b>
6.1	References . . . . .	19
6.2	Spaces, Lengths, and Dimensions . . . . .	19
6.3	Including Graphics . . . . .	19
6.4	Floating Environments . . . . .	20

6.5 Citations and the Bibliography . . . . .	21
6.5.1 <i>BIB</i> Files . . . . .	22
6.5.2 Citing Sources . . . . .	23
6.5.3 BibTeX . . . . .	24
6.5.4 Author-Year Bibliographies with natbib . . . . .	24
<b>7 More Information</b>	<b>24</b>
<b>A Mathematical Symbols</b>	<b>25</b>

## 1 Introduction

Have you ever tried to write a mathematical paper in Microsoft Word? Newer versions of Word contain an equation editor that makes it possible, but you probably find moving the mouse to click on each symbol annoying and tedious. Don't despair; there's another way!

L<sup>A</sup>T<sub>E</sub>X is the standard tool to typeset papers in mathematics, physics, economics, and other disciplines. It is so common that most scientific and mathematical journals give special instructions on how to format articles for submission. In fact, it is very hard to find a mathematical journal that does not explicitly prefer L<sup>A</sup>T<sub>E</sub>X submissions over any other type of electronic submission.

L<sup>A</sup>T<sub>E</sub>X is actually not a program. It is a set of macros for Donald Knuth's T<sub>E</sub>X typesetting program, which takes a plain text file encoded with special commands and outputs a typeset document. Some people prefer to use plain T<sub>E</sub>X, but we will concentrate on L<sup>A</sup>T<sub>E</sub>X for its widespread use and ease of learning.

Because L<sup>A</sup>T<sub>E</sub>X uses just plain text files, you will use the keyboard to input everything. You might find this more difficult at first to remember all the commands, but eventually you will find it much faster to write a mathematical paper using L<sup>A</sup>T<sub>E</sub>X than Microsoft Word.

The best way to learn L<sup>A</sup>T<sub>E</sub>X is to experiment. Take a file that already works and change it and see what happens. Use this document or any L<sup>A</sup>T<sub>E</sub>X document you find on the web. You can even start from scratch.

Throughout this paper, we describe the various parts of this document. Section 2 describes how to get started on Wabash's public Windows machines, and Section 3 describes basic L<sup>A</sup>T<sub>E</sub>X files and commands. Sections 4 and 6 describe how to typeset mathematical equations and how to incorporate more advanced environments such as automatic references, graphics, and floating figures and tables. Finally, since we have only scratched the surface of L<sup>A</sup>T<sub>E</sub>X's capabilities, Section 7 describes how to find more information.

## 2 Getting Started

To begin, you need only a few programs: a basic text editor and a L<sup>A</sup>T<sub>E</sub>X processor. On a Unix machine, one often runs these separately. For example, you could run an editor such as emacs or vi to edit the file and then run the command `latex` to process the file. You will find most L<sup>A</sup>T<sub>E</sub>X documentation describing this method. Because it is so widely described, and because you will likely use a public Windows machine to typeset your document, we will concentrate on the two integrated development environments (IDEs) installed on all public PCs on campus: PCTeX and ProTeXt. Section 7 describes a website where you can find links to other free and commercial versions of L<sup>A</sup>T<sub>E</sub>X for any platform.

## 2.1 PCTeX

Wabash College has a site license for the commercial software PCTeX. You can find it on the public Windows machines under mathematics software. You can also buy a personal copy from the website <http://www.pctex.com>, or you can get a copy to evaluate free for 30 days. This might be a good way to use L<sup>A</sup>T<sub>E</sub>X on your personal computer in the short term.

Once you start the program, it acts like most standard Windows applications. You can use the *File* menu in the upper left to open L<sup>A</sup>T<sub>E</sub>X files. PCTeX comes with several sample files in a directory named *samples*. The public machines may open to it automatically. If not, look for a directory named *PCTeX* on the local drive. The *samples* directory will be in it, probably a level or two down.

After you have opened a L<sup>A</sup>T<sub>E</sub>X file, or have created your own, you can typeset it with the *Typeset* button in the middle of the row below the menus near the top of the window. Be sure you see L<sup>A</sup>T<sub>E</sub>X in the pull-down menu next to it. It might not be activated yet. In which case, you need to go to the *Typeset* menu and run *INITeX*. Choose the L<sup>A</sup>T<sub>E</sub>X macros and then run *INITeX*.

By typesetting your document, you create a file called a *DVI* file that is independent of the printer you will use. This file should appear, and you can see the results of your efforts. You may encounter error in typesetting your document. For example, we often find misspelled commands that L<sup>A</sup>T<sub>E</sub>X doesn't understand. Fortunately, PCTeX will tell us where it encounters errors so we can fix them.

You might also have to typeset the program a second time. This often happens when you are using L<sup>A</sup>T<sub>E</sub>X's automatic referencing capabilities. If PCTeX suggests you typeset the document again, you should do it.

Once you successfully typeset your program and create a DVI file, you can print it or you can export it to various document types. If you want to send it to someone else, we suggest exporting it as a PDF file by using the *Export* command in the *File* menu.

## 2.2 TeXworks

Wabash College also installed a copy of the free software TeXworks on all public PCs. You can find it on the public Windows machines under mathematics software:

Start / Courses / Mathematics / TeXworks .

You can also download a personal copy from the website <http://www.tug.org>. This may be your best long-term way to use L<sup>A</sup>T<sub>E</sub>X on your personal computer.

Once you start the program, it acts like most standard Windows applications. You can use the *File* menu in the upper left to open L<sup>A</sup>T<sub>E</sub>X files. You can use the "New from Templates" entry in the *File* menu to open templates or sample files.

After you have opened a  $\LaTeX$  file, or have created your own, you can typeset it with the click of a button. Look for the *Typeset* menu in the middle of the menus near the top of the window. When you open the menu, you can choose which program(s) you want to use to create the output. It will probably default to *pdfLaTeX+MakeIndex+BibTex*. I suggest you either use that or *pdfLaTeX*.

You may encounter error in typesetting your document. For example, we often find misspelled commands that  $\LaTeX$  doesn't understand. Fortunately, TeXworks will tell us where it encounters errors so we can fix them. Notice the log at the bottom of the screen. At the very end, it should tell you how many errors and warnings it encountered. Errors will keep your document from compiling, while warnings will make it appear incorrectly. You must fix errors, but you can ignore warnings if you do not care about the appearance. You can scroll back up through the log to see what happened.

You might also have to typeset the program a second time. This often happens when you are using  $\LaTeX$ 's automatic referencing capabilities. TeXworks will often automatically typeset the document a second time, but if you find any undefined references—which will be marked by “??”—you might first try typesetting it again.

## 2.3 $\LaTeX$ on Your Personal Computer

You can also use  $\LaTeX$  on your personal computer. The TeX Users Group website [9] contains information on many commercial and free TeX and  $\LaTeX$  implementations for all operating systems. If you have a PC, you can install a demo version of PCTeX on your computer, or you can other free versions such as ProTeXt, TeX Live, fpTeX, emTeX, or MikTeX. On a Windows machine, I suggest you install the latest version of MikTeX, which should come with TeXworks. (You might also want to install Ghostscript and GSView to view postscript files.) On a Mac, my colleagues in the Physics Department suggest TeXShop., which is very similar to TeXworks. You can find its website by searching on Google.

## 2.4 $\LaTeX$ Online

It is becoming increasingly more popular to use  $\LaTeX$  online. Overleaf is an online, collaborative  $\LaTeX$  editor that is available for free for students. It also has more advanced memberships with more features for a monthly or annual fee. You can find it here: <https://www.overleaf.com/>. It also has documentation to help you learn  $\LaTeX$  at <https://www.overleaf.com/learn>.

Overleaf may act a little differently from the other compilers I mentioned above. In particular, students have told me the behavior of the Q.E.D. symbol differs from what I describe in Section 5.1, and that the `\qedhere` command is not supported.

In order to submit an assignment that you write in Overleaf, you will probably need to download the output or possibly the source file. You can download either of them by clicking on the **Menu** button at the top left of the project

screen. The source files will be in a zipper file, which you may need to extract. The output should be in a single PDF file.

## 3 Basic L<sup>A</sup>T<sub>E</sub>X

### 3.1 Commands and Comments

As you look at any L<sup>A</sup>T<sub>E</sub>X file, you will see various funny-looking strings of letters. In several places, you will see the character `%`. This signifies the rest of the line is a comment, which L<sup>A</sup>T<sub>E</sub>X will ignore. You can use comments to take out statements you no longer want, or to document your work.

L<sup>A</sup>T<sub>E</sub>X commands begin with the character `\` and run until the first whitespace. Some commands require arguments, such as the `\documentclass{...}` command that begins the file. Other commands, such as `\maketitle` require no arguments. For example, we can print the special word L<sup>A</sup>T<sub>E</sub>X with the command `\LaTeX`. However, L<sup>A</sup>T<sub>E</sub>X recognizes the whitespace after the command as ending the command, not as space between words. Notice there is no space after the word L<sup>A</sup>T<sub>E</sub>X in this sentence. For this reason, we use the braces `{` and `}` to keep the command and the whitespace separate. Notice, throughout the file we use the command `\LaTeX{}`. We could have also used the command `{\LaTeX}` to achieve the same result.

### 3.2 File Layout

Every L<sup>A</sup>T<sub>E</sub>X file consists of two parts: the preamble and the body. The *preamble* begins with the `\documentclass`, usually the first line of the file, and sets certain information about the document. In this case, the command

```
\documentclass{article}
```

says we are writing an article. The command also may take options. For example, we could use the command `\documentclass[12pt]{...}` to indicate a 12 point font, or `\documentclass[10pt,a4paper]{...}` to indicate 10 point font on A4 paper. We could have also used `\documentclass{report}` to create a report or `\documentclass{book}` to create a book. Different document classes behave differently. We will concentrate on the article class here. Our preamble also includes commands to use additional packages.

The *body* begins with the command `\begin{document}` and ends with the command `\end{document}`, and it contains all the text of the document. L<sup>A</sup>T<sub>E</sub>X will ignore anything after the `\end{document}` command.

### 3.3 Titles, Authors, Dates

The command `\maketitle` that appears near the beginning of the body of our file tells L<sup>A</sup>T<sub>E</sub>X to make the title for the document. It uses information that we input in commands `\title{...}`, `\author{...}`, and possibly `\date{...}`



sometime before the `\maketitle` command. We placed these commands in the preamble, but we could have just as easily placed them in the body before the `\maketitle` command.

L<sup>A</sup>T<sub>E</sub>X requires us to use two of these commands: namely `\title{...}` and `\author{...}`. The `\date{...}` is special because L<sup>A</sup>T<sub>E</sub>X will use today's date if we don't specify a date. You might not specify a date so L<sup>A</sup>T<sub>E</sub>X will tell you the date you last L<sup>A</sup>T<sub>E</sub>Xed the file, or you might want to specify the date so L<sup>A</sup>T<sub>E</sub>X will not change it on you.

### 3.4 Sections

You can divide a L<sup>A</sup>T<sub>E</sub>X document into sections using the `\section{...}` command. This command automatically numbers the sections using an internal counter. L<sup>A</sup>T<sub>E</sub>X also has other sectioning commands to produce other levels of headings. Descending from `\section{...}`, L<sup>A</sup>T<sub>E</sub>X also has the commands

```
\subsection{...}
\subsubsection{...}
\paragraph{...}
\subparagraph{...}
```

for increasingly finer divisions.

Each of these sectioning commands also has an unnumbered version

```
\section*{...}
\subsection*{...}
\subsubsection*{...}
\paragraph*{...}
\subparagraph*{...}
```

These commands produces the same heading style without a counter. For example, the References section after Section 7 is produced with in this way, although the mechanism is actually hidden within the `\bibliography{...}` command. You might find this useful for typesetting your homework where you might not want the section numbers to appear.

L<sup>A</sup>T<sub>E</sub>X also has a `\part` command that lies above the `\section` command in this hierarchy. The book and report classes also have a `\chapter` command between parts and sections.

This automatic numbering is helpful when you rearrange sections. It is also useful when you want to automatically reference a section as described in Section 6.5. You can also create a table of contents, although you will probably not encounter that in a paper.

### 3.5 Words, Paragraphs, and Whitespace

L<sup>A</sup>T<sub>E</sub>X interprets *blank spaces*, *tabs*, and *carriage returns* as the end of words and commands. It does not matter how many spaces or tabs you put between

words or commands.  $\LaTeX$  will ignore this when it places the space between words in the final output. This means you can leave whitespace (for example through indenting a line) only for your own convenience.

$\LaTeX$  interprets one or more *empty lines* as the end of a paragraph. Once again, the spacing between paragraphs is independent of the number of empty lines. However, you can only leave a blank line if you want to start a new paragraph. If you want to leave a blank line for your convenience without starting a new paragraph, you can cheat by making the line contain a blank comment by starting it with a comment character `%`. (See Section 3.1.)

$\LaTeX$  treats each paragraph as one long string of words. It then breaks up lines automatically independent of the input, and makes the spacing between the words on each line as uniform as possible so that each line is left and right justified. It will also automatically space and indent each paragraph based upon the document class in use.

### 3.6 Quotation Marks

Another trick you might encounter is placing quotation marks around a sentence.  $\LaTeX$  recognizes a difference between `'` and `'`, and it usually interprets `"` as `'`. To make the quotation marks appear normally, use `“` and `”` to quote material. For example, `“this”` produces “this.”

### 3.7 Tables

Sometimes you might want to put information in a table. To do so, you can use the *tabular* environment. For example, the text

```
\begin{tabular}{lccr}
    Name & Test 1 & Test 2 & Average \\
    Bob & 90 & 80 & 85 \\
    Joe & 60 & 90 & 75
\end{tabular}
```

produces the table

Name	Test 1	Test 2	Average
Bob	90	80	85
Joe	60	90	75

Be sure to leave a blank line before and after the environment so it doesn't place the table inside a paragraph.

Here the command `\begin{tabular}{lccr}` begins the *tabular* environment and tells  $\LaTeX$  to make the table have four columns. The `{lccr}` says first one is left-justified, the middle two are centered, and the last is right-justified. The command `\end{tabular}` ends the *tabular* environment. Between these two commands, we place the body of the matrix in a *row-major* form where we read across each row before going to the next row. We separate the entries of each row with ampersands `&` and separate each line with the newline command `\\`.

We can also place vertical and horizontal lines between the rows and columns of the table. We place vertical lines between the columns by placing a `|` between the justification indicators at the beginning of the environment. We place horizontal lines between the rows by using the `\hline` command. This command must come one a new line, so be sure to end any line before it with the newline command `\\`. For example, the text

```
\begin{tabular}{|l|c|c|r|}
\hline
\hline
Name & Test 1 & Test 2 & Average \\
\hline
Bob & 90 & 80 & 85 \\
Joe & 60 & 90 & 75 \\
\hline
\hline
\end{tabular}
```

produces the table

Name	Test 1	Test 2	Average
Bob	90	80	85
Joe	60	90	75

## 4 Mathematics

$\LaTeX$  handles mathematical equations with ease. You enter mathematical equations by switching to *math* mode.

### 4.1 Text and Display Styles

$\LaTeX$  has two types of mathematical equations: *text* and *display* styles.  $\LaTeX$  changes slightly how equations are displayed depending on the style, but you enter mathematics the same way in both.

You can enter equations in text style to occur in the line of the text by placing a dollar sign `$` on either side. For example, `$x + y = z$` produces  $x + y = z$ . You can also enter equations in display style to occur centered on their own line by placing two dollar signs `$$` on either side. For example, `$$x + y = z$$` produces

$$x + y = z$$

### 4.2 AMS $\LaTeX$

The American Mathematical Society created additional  $\LaTeX$  macros to make typesetting mathematical equations even easier [1]. In this document, we are using the `amsmath`, `amsfont`, and `amssymb` packages. The manual for the `amsmath` package [3] describes more features of this package than we will go into here. In

particular, you might be interested in how to number equations, how to split equations over several lines in display mode, or how to create new mathematical operators such as `sin`. The AMS also created a short guide to typesetting mathematics with these three packages [2].

The AMS also has a fourth package, `amsthm`, that you can use to typeset theorems and proofs [4]. You might find these features more useful in theory classes and when writing research papers. See Section 5.2 for more information.

### 4.3 Superscripts and Subscripts

To enter a superscript such as in an exponent, you use the `^` character. For example, `e^x` produces  $e^x$ . To use more complicate exponents, you must group the exponent with braces. For example, `e^{x+1}` produces  $e^{x+1}$ .

Similarly you can enter subscripts with the `_` character. For example, `x_0` produces  $x_0$ , and `x_{i+3}` produces  $x_{i+3}$ .

### 4.4 Binary Relations and Negations

We can compare two mathematical quantities by connecting them with a relation such as `=`, `<=`, or `<`. Table 3 in Appendix A lists several common binary operators and relations.

We can indicate the opposite, or negated, meaning of the relation with a slash `'/` through the symbol. We can do this in  $\LaTeX$  by putting a `\not` before the relation command. For example, `\leq` produces  $\leq$ , and `\not\leq` produces  $\not\leq$ . Similarly `\not=` produces  $\neq$ , and `\not>` produces  $\not>$ .  $\LaTeX$  also has a few negated symbols already, such as `\neq`, which produces  $\neq$ , and `\notin`, which produces  $\notin$ .

### 4.5 Function Names

In math mode,  $\LaTeX$  interprets strings of characters as separate variables. For example, `sin(x)` produces  $sin(x)$ , which should be read as  $s \times i \times n \times (x)$ . This is probably not what you want to happen. Instead, you might use one of  $\LaTeX$ 's built-in *operators* such as `\sin`. In this case, `\sin(x)` produces  $\sin(x)$ . Most mathematical functions you know have operators in  $\LaTeX$ . (See Table 6 in Appendix A.)

You can attach limits to some of these functions, such as `\min`, `\max`, and `\lim`, through the subscript command. For example `\lim_{x \to a} x^2` produces  $\lim_{x \rightarrow a} x^2$  and

$$\lim_{x \rightarrow a} x^2.$$

### 4.6 Integrals, Sums, etc

Use the command `\int` to produce the integral sign in  $\LaTeX$ . For example, you might use `\int x^3 \ dx` to produce  $\int x^3 dx$ . Here I use the “`\`” to produce an extra space between the  $x^3$  and the  $dx$  to emphasize the  $dx$ . To produce

the limits of a definite integral, just use the super- and subscripts described in Section 4.3. For example, `\int_0^{10} x^3 \ dx` produces  $\int_0^{10} x^3 dx$  in text mode and

$$\int_0^{10} x^3 dx$$

in display mode.

Similarly you can use `\sum` to produce a summation. For example,

`\sum_{k=0}^{10} k^3`

produces  $\sum_{k=0}^{10} k^3$  and

$$\sum_{k=0}^{10} k^3.$$

L<sup>A</sup>T<sub>E</sub>X also has a `\prod` command for products.

## 4.7 Fractions

To enter fractions in text mode, you may want to use the slash `/` such as `x/y`. However, in display mode you probably want a horizontal bar such as

$$\frac{x}{y}.$$

You can accomplish this by using the `\frac{...}{...}` command. For example, we used `\frac{x}{y}` to produce this fraction. You can use much more complicated expressions in both the numerator and denominator, such as  $\frac{dy}{dx}$  or

$$\frac{e^x}{x^3 + 5x - 1}.$$

## 4.8 Automatic Sizing of Bracket Symbols

Often in mathematical writing you want to use bracketing symbols, usually in pairs to enclose part of a formula. To make these symbols an appropriate size for the enclosed formula, L<sup>A</sup>T<sub>E</sub>X provides a pair of commands `\left` and `\right`. To use these commands, you place them immediately before the corresponding left- and right-hand bracketing symbols. For example,

`\left[ \frac{1}{4} x^4 \right]_0^{10}`

produces  $\left[\frac{1}{4}x^4\right]_0^{10}$  and

$$\left[\frac{1}{4}x^4\right]_0^{10}.$$

The bracketing symbols don't have to match, but you do have to use both the `\left` and `\right` commands. For example, you could use the command `\left| \int \right|` to produce  $|\int|$ , although it usually doesn't make sense

to mix the brackets. Table 7 in Appendix A provides a list of delimiters you can use for this purpose.

Sometimes you might want an *invisible* bracket. You can do this by using a period ‘.’ as the bracketing symbol. For example, we can use

```
\int_0^{10} x^3 \ dx
= \left. \frac{1}{4} x^4 \right|_0^{10}
```

to produce

$$\int_0^{10} x^3 dx = \frac{1}{4}x^4 \Big|_0^{10}.$$

Another case where you might use this is for a piece-wise defined function. For example,

```
y = \left\{
      \begin{array}{r@{\quad:\quad}l}
        -1 & x < 0, \\
         0 & x = 0, \\
         1 & x > 0.
      \end{array}
\right.
```

produces

$$y = \begin{cases} -1 & : x < 0, \\ 0 & : x = 0, \\ +1 & : x > 0. \end{cases}$$

Here, we used \{ to produce the opening brace. Notice the escape character \, which L<sup>A</sup>T<sub>E</sub>X requires to produce the symbol because otherwise it thinks of it as grouping things together. We also used the *array* environment, which is basically a *tabular* environment in math mode. We also used the command @{\quad:\quad} to produce the spacing (by \quad) and the colon between the two columns of the array. AMS-L<sup>A</sup>T<sub>E</sub>X provides another way to do define a piecewise function this using a *cases* environment. For example,

```
y = \begin{cases}
      -1, & \text{if } x < 0, \\
       0, & \text{if } x = 0, \\
       1, & \text{if } x > 0.
    \end{cases}
```

produces

$$y = \begin{cases} -1, & \text{if } x < 0, \\ 0, & \text{if } x = 0, \\ +1, & \text{if } x > 0. \end{cases}$$

## 4.9 Matrices

We actually have a couple of ways to produce matrices in our papers. We could use the bracketing symbols and arrays as we did for the piece-wise defined functions in Section 4.8. This requires us to know how many columns are needed before we start writing the matrix.

The `amsmath` package gives us another way to incorporate matrices into our papers. We can do this by using one of several matrix *environments*. For example, the text

```
$$
\begin{pmatrix}
a & b \\
c & d
\end{pmatrix}
$$
```

produces the output

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Here the command `\begin{pmatrix}` begins the *pmatrix* environment and the command `\end{pmatrix}` ends the *pmatrix* environment. Between these two commands, we place the body of the matrix in a *row-major* form where we read across each row before going to the next row. We separate the entries of each row with ampersands `&` and separate each line with the newline command `\`. AMS- $\text{\LaTeX}$  has other environments to surround the matrices with brackets (*bmatrix*:  $[a]$ ), braces (*Bmatrix*:  $\{a\}$ ), single lines (*vmatrix*:  $|a|$ ), and double lines (*Vmatrix*:  $\|a\|$ ).

## 4.10 Other Useful Commands

Finally, we present a few other mathematical commands you might find useful. You might use the command `\sqrt{...}` to produce a square-root as in  $\sqrt{x^2 + 1}$ . You can produce a cube-root with this command. For example, `\sqrt[3]{...}` produces  $\sqrt[3]{x^2 + 1}$ . Here the brackets designate an optional parameter to the command.

In combinatorics or statistics, the command `{a \choose b}` might be useful to produce  $\binom{a}{b}$ . This command is really a  $\text{\TeX}$  command, and so its format differs from the  $\text{\LaTeX}$  commands in the rest of this paper. In this command, the braces are important to let the `\choose` command know what goes above and below it. The command `\atop` produces the same output without the parentheses. For example,  $\binom{a}{b}$ .

Appendix A includes a few tables of mathematical symbols you might find useful, as well as how to use other alphabets to produce symbols such as  $\mathbb{R}$  and  $\mathbb{Z}$ .

## 5 Theorems and Proofs Using the `amsthm` Package

Perhaps the simplest way to incorporate theorems and proofs into a  $\text{\LaTeX}$  paper is to use the `amsthm` package. This package provides a predefined `proof` environment and the ability to define environments for theorems, lemmas, corollaries, propositions, definitions, examples, remarks, notes, etc.

The first thing you must do is load the package in the preamble by using the command

```
\usepackage{amsthm}
```

Unlike other AMS packages, the `amsmath` package does not automatically load it.

### 5.1 Proofs

Once you have access to the `proof` environment.

```
\begin{proof}  
...  
\end{proof}
```

This environment provides an unnumbered structure with the title *Proof*. The environment terminates with the Q.E.D. symbol  $\square$ . You may alter the symbol by redefining the `\qedsymbol` command, and you may print it at any time with the command `\qed`.

Normally the Q.E.D. symbol appears on the last line of text flush with the right margin:

*Proof.* ... Thus the proof is complete.  $\square$

However, if a proof ends with an equation in display mode, the Q.E.D. symbol will appear on the next line by itself:

*Proof.* ... Thus we get the equation

$$x^2 + y^2 = z^2.$$

$\square$

To avoid this you may use the `\qedhere` command within the math mode for the final equation to force the Q.E.D. symbol to appear immediately after the equation:

*Proof.* ... Thus we get the equation

$$x^2 + y^2 = z^2. \quad \square$$



## 5.2 Theorem-Type Environments

The `amsthm` package does not directly provide any environments for writing theorems, but you may easily create them with the `\newtheorem` command. For example, the command

```
\newtheorem{theorem}{Theorem}
```

provides a `theorem` environment

```
\begin{theorem}  
...  
\end{theorem}
```

that has the numbered *Theorem* label. Similarly you can create any other theorem-like environments you want. For example, the commands

```
\newtheorem{lemma}{Lemma}  
\newtheorem{corollary}{Corollary}
```

create environments labeled *Lemma* and *Corollary*.

## 5.3 Changing Numbers

By default, every theorem-like environment uses different numbering sequences and are number sequentially throughout the document. For example, the code used in the previous section provides environments numbered *Theorem 1*, *Theorem 2*, *Lemma 1*, *Lemma 2*, *Corollary 1*, *Corollary 2*, etc. The `\newtheorem` command takes two optional arguments to change this. Be careful, you cannot use both of these optional arguments at the same time.

If the optional argument of an existing theorem-like definition comes before the new title, the new environment will be numbered in the same sequence as the first. For example, the code

```
\newtheorem{theorem}{Theorem}  
\newtheorem{lemma}[theorem]{Lemma}  
\newtheorem{corollary}[theorem]{Corollary}
```

provides environments numbered *Theorem 1*, *Theorem 2*, *Lemma 3*, *Lemma 4*, *Corollary 5*, *Corollary 6*, etc.

On the other hand, if the optional argument of a counter name like chapter or section appears after the new title, incrementing this counter will reset the counter of the environment. For example, the code

```
\newtheorem{theorem}{Theorem}[section]
```

will provide an environment numbered within sections as in *Theorem 1.1*, *Theorem 1.2*, *Theorem 2.1*, *Theorem 3.1*, etc.

## 5.4 Theorem Styles

The package has three predefined theorem styles:

**plain** The title and number are in bold and the text italic.

**definition** The title and number are in bold and the text normal.

**remark** The title and number are in italic and the text normal.

By default, the `amsthm` package uses the `plain` theorem style, but you may use the `\theoremstyle` command to activate a new style. The package will use this style until the next style is activated. For example, the code

```
\newtheorem{theorem}{Theorem}
\newtheorem{lemma}{Lemma}

\theoremstyle{definition}
\newtheorem{definition}{Definition}
\newtheorem{example}{Example}

\theoremstyle{remark}
\newtheorem{remark}{Remark}
\newtheorem{note}{Note}
```

creates six environments. The first two, `theorem` and `lemma` use the plain style; the middle two, `definition` and `example` use the definition style; and the last two, `remark` and `note` use the remark style

## 5.5 Specific Names

Sometimes you might want to add a specific name to a theorem or proof. To do so, use an optional argument when you start the environment. For example, replace the commands

```
\begin{theorem} ... \end{theorem}
\begin{proof} ... \end{proof}
```

with

```
\begin{theorem}[Theorem Name] ... \end{theorem}
\begin{proof}[Proof of Theorem Name] ... \end{proof}
```

The theorem will add the optional name in parentheses after the numbered label, and the proof will replace the title with the optional one.

## 5.6 Examples

Sections 5.6.1 and 5.6.2 provide examples of the theorem and proof environments. Section 5.6.1 uses the standard environments described in Section 5.2,

and Section 5.6.2 uses the environments with optional arguments to include specific names described in Section 5.5.

All of these examples come in modified versions from Section 1.2 of *An Introduction to the Theory of Numbers*, Fifth Edition, by Ivan Niven, Herbert S. Zuckerman, and Hugh L. Montgomery.

### 5.6.1 Divisibility

**Definition 1.** An integer  $b$  is divisible by an integer  $a \neq 0$  if there exists an integer  $x$  such that  $b = ax$ . We write  $a \mid b$ . In case  $b$  is not divisible by  $a$ , we write  $a \nmid b$ .

Instead of saying  $b$  is divisible by  $a$ , we may also say that  $a$  divides  $b$ , that  $a$  is a divisor of  $b$ , or that  $b$  is a multiple of  $a$ . If  $a \mid b$  and  $0 < a < b$ , then  $a$  is called a *proper divisor* of  $b$ . It is understood that we never use 0 as the left member of the pair of integers in  $a \mid b$ . On the other hand, not only may 0 occur as the right member of the pair, but also in such instances we always have divisibility. Thus,  $a \mid 0$  for every integer  $a \neq 0$ .

**Lemma 1.**

1.  $a \mid b$  implies  $a \mid bc$  for any integer  $c$ ;
2.  $a \mid b$  and  $b \mid c$  imply  $a \mid c$ ;
3.  $a \mid b$  and  $a \mid c$  imply  $a \mid (bx + cy)$  for any integers  $x$  and  $y$ ;
4.  $a \mid b$  and  $b \mid a$  imply  $a = \pm b$ ;
5.  $a \mid b$  and  $b > 0$  imply  $a \leq b$ ;
6. if  $m \neq 0$ , then  $a \mid b$  implies and is implied by  $ma \mid mb$ .

The proofs of these results follow from the definition of divisibility.

The next result is a formal statement of the outcome when an integer  $b$  is divided by any positive integer. For example, if 25 is divided by 7, the quotient is 3 and the remainder is 4. These numbers are related by the equality  $25 = 7 \cdot 3 + 4$ . Now we reformulate this in the general case.

**Theorem 1.** *The division algorithm. Given any integers  $a$  and  $b$ , with  $a > 0$ , there exist unique integers  $q$  and  $r$  such that  $b = qa + r$  and  $0 \leq r < a$ . If  $a \nmid b$ , then  $r$  satisfies the stronger inequalities  $0 < r < a$ .*

*Proof.* Consider the arithmetic progression

$$\dots, b - 3a, b - 2a, b - a, b, b + a, b + 2a, b + 3a, \dots$$

extending indefinitely in both directions. In this sequence, select the smallest nonnegative member and denote it  $r$ . Thus by definition  $r$  satisfies the inequalities of the theorem, but also  $r$ , being in the sequence, is of the form  $b - qa$ , and thus  $q$  is defined in terms of  $r$ .

To prove uniqueness of  $q$  and  $r$ , suppose there is another pair  $q_1$  and  $r_1$  satisfying the same conditions. First we prove  $r_1 = r$ . For if not, we may presume  $r < r_1$  so that  $0 < r_1 - r < a$ , and then we see that  $r_1 - r = a(q - q_1)$  and so  $a \mid (r_1 - r)$ , a contradiction to Lemma 1. Hence  $r = r_1$  and  $q = q_1$ .  $\square$

We stated the theorem with the assumption  $a > 0$ . This hypothesis is not necessary, and we may reformulate the theorem without it: given any integers  $a$  and  $b$ , with  $a \neq 0$ , there exist unique integers  $q$  and  $r$  such that  $b = qa + r$  and  $0 \leq r < |a|$ .

### 5.6.2 Greatest Common Divisor

**Definition 2** (Greatest Common Divisor). The integer  $a$  is a *common divisor* of  $b$  and  $c$  in case  $a \mid b$  and  $a \mid c$ . Since there is only a finite number of divisors of any nonzero integer, there is only a finite number of common divisors of  $b$  and  $c$ , except in the case  $b = c = 0$ . If at least one of  $b$  and  $c$  is not zero, the greatest among their common divisors is called the *greatest common divisor* of  $b$  and  $c$  and is denoted  $\gcd(b, c)$ . Similarly we denote the greatest common divisor  $g$  of the integers  $b_1, b_2, \dots, b_n$ , not all zero, by  $\gcd(b_1, b_2, \dots, b_n)$ .

**Theorem 2** (Greatest Common Divisor as Linear Combination). *If  $g$  is the greatest common divisor of  $b$  and  $c$ , then there exist integers  $x_0$  and  $y_0$  such that  $g = \gcd(b, c) = bx_0 + cy_0$ .*

Another way to state this very fundamental result is that the greatest common divisor of two integers  $b$  and  $c$  is expressible as a linear combination of  $b$  and  $c$  with integral multipliers  $x_0$  and  $y_0$ . This assertion holds for not just two integers but for any finite collection.

*Proof of Theorem 2.* Consider the linear combinations  $bx + cy$  where  $x$  and  $y$  range over all integers. This set of integers  $\{bx + cy\}$  includes positive and negative values, and also 0 by the choice  $x = y = 0$ . Choose  $x_0$  and  $y_0$  so that  $bx_0 + cy_0$  is the least positive integer  $\ell$  in the set; thus  $\ell = bx_0 + cy_0$ .

Next we prove  $\ell \mid b$  and  $\ell \mid c$ . We establish the first of these, and the second follows by analogy. We give an indirect proof that  $\ell \mid b$ ; that is, we assume  $\ell \nmid b$  and obtain a contradiction. From  $\ell \nmid b$  it follows by Theorem 1 that there exist integers  $q$  and  $r$  such that  $b = \ell q + r$  and  $0 < r < \ell$ . Hence we have

$$r = b - \ell q = b - q(bx_0 + cy_0) = b(1 - qx_0) + c(-qy_0)$$

and thus  $r$  is in the set  $\{bx + cy\}$ . This contradicts the fact that  $\ell$  is the least positive integer in the set  $\{bx + cy\}$ .

Now since  $g$  is the greatest common divisor of  $b$  and  $c$ , we may write  $b = gB$ ,  $c = gC$ , and

$$\ell = bx_0 + cy_0 = g(Bx_0 + Cy_0).$$

Thus,  $g \mid \ell$ , and so by Lemma 1 we conclude that  $g \leq \ell$ . Now  $g < \ell$  is impossible since  $g$  is the *greatest* common divisor, so

$$g = \ell = bx_0 + cy_0. \quad \square$$

## 6 More Advanced Topics

In this section, we introduce a few advanced topics that may make your papers easier to create and read. We only scratch the surface. Please see Section 7 for how to learn even more.

### 6.1 References

L<sup>A</sup>T<sub>E</sub>X allows us to embed automatic references into our document. For example, the `\section{...}` command not only formats the section title and assigns it a number from a counter, it also sets that counter so that we can reference it. When we started Section 1, we placed the command `\label{intro}` immediately after the `\section{Introduction}` command. We can now reference that section anywhere in the document by using the `\ref{intro}` command. For example, we can use `Section~\ref{intro}` to reference Section 1. The `~` is a non-breaking space and forces the section number to lie on the same line as the word “Section.”

Similarly we can reference other divisions of the paper, theorems, definitions, equations if we use AMS-L<sup>A</sup>T<sub>E</sub>X to number them, figures and other floating environments described in Section 6.4. However, we must ensure each label has a unique marker. Also, the way L<sup>A</sup>T<sub>E</sub>X handles this automatic referencing often requires us to process the file a second or third time to get all the references right. Otherwise, L<sup>A</sup>T<sub>E</sub>X leaves ?? for references it doesn’t understand.

### 6.2 Spaces, Lengths, and Dimensions

L<sup>A</sup>T<sub>E</sub>X contains commands to provide extra spacing. The `\hspace` and `\vspace` command inserts horizontal and vertical spacing, respectively, into your document. Both of these commands take an argument to specify the *length* of the space. A length, which may be positive or negative, is a decimal number followed by a dimensional unit. Permissible dimensions include `in`, `cm`, `mm`, `em` (the width of the letter ‘M’), and `ex` (the height of the letter ‘x’). For example, `\hspace{2.5in}` inserts two and a half inches of horizontal space. Similarly `\vspace{-3.6cm}` inserts negative 3.6 cm of vertical space. The starred versions of the commands insert space even at the beginning of the line or page. For example, `\vspace*{3in}` leaves three inches of vertical space, even at the beginning of the page, whereas `\vspace{3in}` only works if it is not at the beginning of the page.

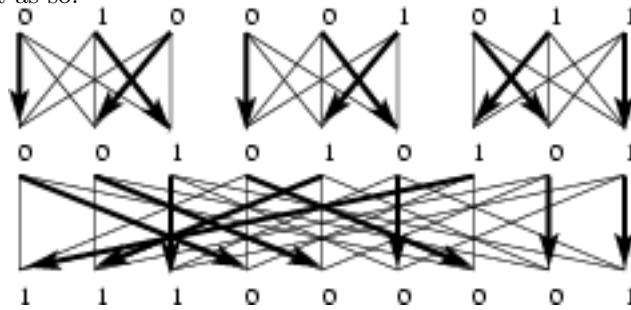
### 6.3 Including Graphics

The easiest way to include graphics in a L<sup>A</sup>T<sub>E</sub>X document is by using the *graphicx* package. You may use this package to include an image that you have already created. You may create this image with any graphics program, but you should save it as an encapsulated postscript (EPS) file. Once you have saved the EPS file, you can use the `\includegraphics{...}` command to include it in your

document. For example, we have the image file `radix3_network_2` from a paper we wrote. We can use the command

```
\centerline{\includegraphics{radix3_network_2}}
```

to include it as so:

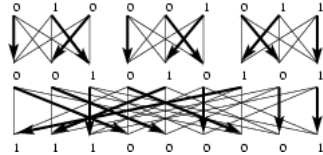


Here, the `\centerline` command centers the image on the line. Be sure to leave a blank line before and after this so it doesn't put it at the end or beginning of a paragraph.

The `\includegraphics` command also takes some optional arguments. You might find the `scale` option to scale the image. For example,

```
\centerline{\includegraphics[scale=0.5]{radix3_network_2}}
```

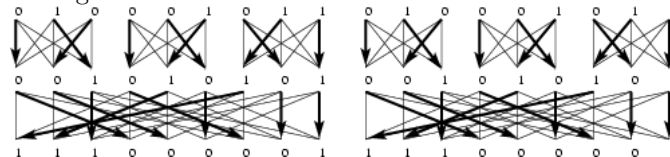
produces the following image that is scaled to half the original size.



You can also include more than one image on a line. The commands

```
\centerline{
  \includegraphics[scale=0.5]{radix3_network_2}
  \quad
  \includegraphics[scale=0.5]{radix3_network_2}
}
```

produces the following pair of images. Here `\quad` produces a little spacing between the images.



## 6.4 Floating Environments

Unfortunately, image files are not smart enough to break to a new page, and you may lose part of your image if you place it too close to the end of a page.

The solution is to use a *floating environment* called the *figure* environment. A floating environment floats to a convenient location in the document. You can specify where to allow the figure, and  $\text{\LaTeX}$  will accommodate you as best it can. For example, the text

```
\begin{figure}[hbt]
  \centerline{\includegraphics{radix3_network_2}}
  \caption{A test figure with the caption below.}
  \label{test fig}
\end{figure}
```

produces the Figure 1.

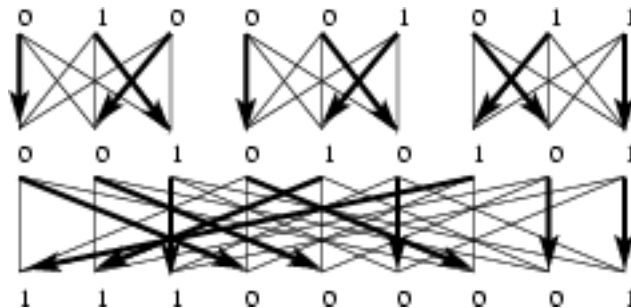


Figure 1: A test figure with the caption below.

Here, the options `[hbt]` tell  $\text{\LaTeX}$  to place the image *here*, at the *bottom* of the page, at the *top* of the next page, or on a *page* of only images, in that order of preference. You may remove some of these designators if you want to limit where  $\text{\LaTeX}$  may place the image.

The `\caption{...}` command places a caption on the figure. It also updates the figure counter, so be sure and place it before the `\label{...}` command. By placing these commands at the end of the figure environment, we tell  $\text{\LaTeX}$  to place the caption after the figure. If you prefer, you can place these commands just after the `\begin{figure}` command to place the caption above the figure as in Figure 2.

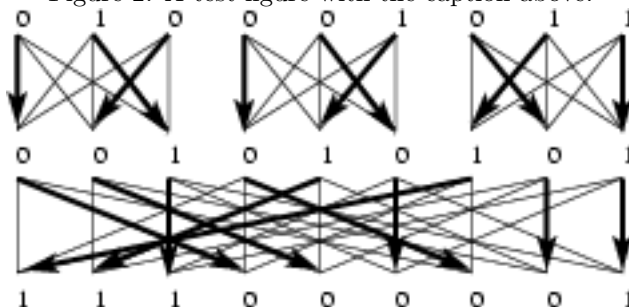
$\text{\LaTeX}$  also has a *table* environment for tables.

## 6.5 Citations and the Bibliography

You may want to include a bibliography or list of references at the end of your paper. Happily,  $\text{\LaTeX}$  will help you automatically generate a bibliography from a database file, called a *BIB* file. (For example, Appendix 6.5 has the *BIB* file for this paper.) This is another plain text file that contains information about the sources you are using. A program called BibTeX takes the information from the database and includes only the sources you cite.

In our document, we use the commands

Figure 2: A test figure with the caption above.



```
\bibliographystyle{plain}
\bibliography{latex}
```

to tell  $\text{\LaTeX}$  what form the bibliography should take and where to find the content. In this case, we are using the *plain* style, which alphabetizes the entries by author and typesfaces them in a certain way. A different style might give a different typeface, a different method of sorting, or both.

The `\bibliography{latex}` tells  $\text{\LaTeX}$  to use the file *latex.bib* for the content information. This command may take more than one argument separated by commas. For example, `\bibliography{latex,foo}` tells  $\text{\LaTeX}$  to also use the file *foo.bib*

### 6.5.1 *BIB* Files

The *BIB* file contains an entry for each reference, and each reference has a corresponding type. For example, the *BIB* file for this document contains contains one book with the code

```
@Book{guide,
  author = "Helmut Kopka and Patrick W. Daly",
  title = "Guide to {LaTeX}",
  edition = "4th",
  publisher = "Pearson Education",
  address = "Upper Saddle River, NJ",
  year = 2003
}
```

Here, `guide` is the label we use for citing this reference. It is followed by several fields giving information about the book: its author, title, edition, publisher, address, and year of publication. Four of these fields (author or editor, title, publisher, and year) are required, and the others are optional. We could also include fields for the volume or number, series, month, and note [6, Appendix B]. Each field has a value given by either a number—as the year—or a string in quotes. Notice the title contains `{LaTeX}` in the string. This forces  $\text{\LaTeX}$  to



use this part of the string exactly as it appears instead of changing it to meet whatever capitalization rules are specified in our particular bibliography style.

Our BIB file also contains one journal article [5]:

```
@Article{CEKSTV02,
  author = "Li Chen
    and Wayne Eberly
    and Erich Kaltofen
    and B. David Saunders
    and William J. Turner
    and Gilles Villard",
  title = "Efficient Matrix Preconditioners for Black Box
    Linear Algebra",
  journal = "Linear Algebra and Applications",
  volume = "343-344",
  pages = "119--146",
  note = "Special issue on \emph{Infinite Systems of
    Linear Equations Finitely Specified}, edited by
    P. Dewilde, V. Olshevsky and A. H. Sayed",
  year = 2002,
  month = "March"
}
```

This entry type requires four fields: author, title, journal, and year. All the other fields are optional. Notice multiple authors are always separated by the word “and.” This allows the BibTeX program to know how many authors the paper has.

Our BIB file contains several entries for miscellaneous sources:

```
@Misc{tug,
  title = "{\TeX} Users Group web page",
  author = "{TeX Users Group}",
  howpublished = "Web page",
  note = "\url{http://www.tug.org}"
}
```

These entry types actually requires no fields, but you may specify authors, title, year, month, how published, and note.

### 6.5.2 Citing Sources

To cite a source, we use the `\cite{...}` command. For example, the command `\cite{CEKSTV02}` cites the article by Chen et al. [5]. We can also use optional arguments to give a specific location in the citation. For example, the command `\cite[Theorem 6.1]{CEKSTV02}` cites Theorem 6.1 of the article by Chen et al. [5, Theorem 6.1] and the command `\cite[Appendix B]{guide}` cites Appendix B of the book by Kopka and Daly [6, Appendix B]. We can also cite several sources at once as with the command `\cite{guide, CEKSTV02}`, which cites the `amsmath` user’s guide and the AMS- $\LaTeX$  website [6, 5].

### 6.5.3 BibTeX

Once you have created a BIB file and cited some sources, you can use the BibTeX program to create a bibliography. To do so, place the `\bibliography{...}` command where you want the bibliography to appear and place the command `\bibliographystyle{...}` anywhere before it. Then typeset the file once with  $\LaTeX$ . Finally run *BibTeX* from the *Tools* menu and  $\LaTeX$  the file again at least once or twice more.

### 6.5.4 Author-Year Bibliographies with natbib

The standard  $\LaTeX$  bibliography implementation numbers references and cites them by their numbers. An alternative style that many journals and texts use references works by citing the author's name and year of publication. This may take the familiar form of the APA style that many standard English style guides include, or other related forms. There are a number of packages developed to deal with this situation, but the most universal of these is the `natbib` package by Patrick W. Daly. It is compatible with the standard bibliography styles as well as a number of author-year styles. See §11.3.4 of the *Guide to LaTeX* by Helmut Kopka and Patrick W. Daly [6] for more information.

## 7 More Information

You can easily find more information about  $\LaTeX$ . The T<sub>E</sub>X Users Group maintains a website [9] that contains links to all the information you might ever want to know about T<sub>E</sub>X and  $\LaTeX$ . You can find information about obtaining freeware and commercial software for any computer platform, tutorials and documents describing how to do anything you'd ever want to do, and books about T<sub>E</sub>X and  $\LaTeX$ . If you are interested in a good book about  $\LaTeX$ , check out the book *Guide to LaTeX* by Helmut Kopka and Patrick W. Daly [6].

## References

- [1] American Mathematical Society. AMS-LaTeX. Web page. <http://www.ams.org/tex/amslatex.html>.
- [2] American Mathematical Society. Short math guide for  $\LaTeX$ . <ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf>.
- [3] American Mathematical Society. User's guide for the `amsmath` package. <ftp://ftp.ams.org/pub/tex/doc/amsmath/amslatex.pdf>.
- [4] American Mathematical Society. Using the `amsthm` package. <ftp://ftp.ams.org/pub/tex/doc/amslatex/amsthdoc.pdf>.
- [5] Li Chen, Wayne Eberly, Erich Kaltofen, B. David Saunders, William J. Turner, and Gilles Villard. Efficient matrix preconditioners for black box

linear algebra. *Linear Algebra and Applications*, 343-344:119–146, March 2002. Special issue on *Infinite Systems of Linear Equations Finitely Specified*, edited by P. Dewilde, V. Olshevsky and A. H. Sayed.

- [6] Helmut Kopka and Patrick W. Daly. *Guide to LaTeX*. Pearson Education, Upper Saddle River, NJ, 4th edition, 2003.
- [7] Tobias Oetiker. The not so short introduction to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. <http://www.ctan.org/tex-archive/info/lshort/english/>.
- [8] Scott Perkins. The comprehensive L<sup>A</sup>T<sub>E</sub>X symbol list. <http://www.ctan.org/tex-archive/info/symbols/comprehensive/>.
- [9] TeX Users Group. T<sub>E</sub>X users group web page. Web page. <http://www.tug.org>.

## A Mathematical Symbols

The following tables give some of the basic mathematical symbols you might want to use. They demonstrate some of the symbols normally accessible from *math mode*. You can consult other L<sup>A</sup>T<sub>E</sub>X references for more symbols. In particular, Oetiker [7] and Kopka and Daly [6] provide tables of mathematical symbols. Perkins [8] provides one of the most comprehensive symbol lists available.

Table 1: Math Mode Accents.

$\hat{a}$	<code>\hat{a}</code>	$\check{a}$	<code>\check{a}</code>	$\tilde{a}$	<code>\tilde{a}</code>	$\acute{a}$	<code>\acute{a}</code>
$\grave{a}$	<code>\grave{a}</code>	$\dot{a}$	<code>\dot{a}</code>	$\ddot{a}$	<code>\ddot{a}</code>	$\breve{a}$	<code>\breve{a}</code>
$\bar{a}$	<code>\bar{a}</code>	$\vec{a}$	<code>\vec{a}</code>	$\widehat{A}$	<code>\widehat{A}</code>	$\widetilde{A}$	<code>\widetilde{A}</code>

Table 2: Greek Letters.

$\alpha$	<code>\alpha</code>	$\theta$	<code>\theta</code>	$o$	<code>o</code>	$v$	<code>\upsilon</code>
$\beta$	<code>\beta</code>	$\vartheta$	<code>\vartheta</code>	$\pi$	<code>\pi</code>	$\phi$	<code>\phi</code>
$\gamma$	<code>\gamma</code>	$\iota$	<code>\iota</code>	$\varpi$	<code>\varpi</code>	$\varphi$	<code>\varphi</code>
$\delta$	<code>\delta</code>	$\kappa$	<code>\kappa</code>	$\rho$	<code>\rho</code>	$\chi$	<code>\chi</code>
$\epsilon$	<code>\epsilon</code>	$\lambda$	<code>\lambda</code>	$\varrho$	<code>\varrho</code>	$\psi$	<code>\psi</code>
$\varepsilon$	<code>\varepsilon</code>	$\mu$	<code>\mu</code>	$\sigma$	<code>\sigma</code>	$\omega$	<code>\omega</code>
$\zeta$	<code>\zeta</code>	$\nu$	<code>\nu</code>	$\varsigma$	<code>\varsigma</code>		
$\eta$	<code>\eta</code>	$\xi$	<code>\xi</code>	$\tau$	<code>\tau</code>		
$\Gamma$	<code>\Gamma</code>	$\Lambda$	<code>\Lambda</code>	$\Sigma$	<code>\Sigma</code>	$\Psi$	<code>\Psi</code>
$\Delta$	<code>\Delta</code>	$\Xi$	<code>\Xi</code>	$\Upsilon$	<code>\Upsilon</code>	$\Omega$	<code>\Omega</code>
$\Theta$	<code>\Theta</code>	$\Pi$	<code>\Pi</code>	$\Phi$	<code>\Phi</code>		

Table 3: Binary Relations and Operators.

You can negate the following symbols by prefixing them with a `\not` command. The `\nmid` command requires the `amssymb` package.

$<$	<code>&lt;</code>	$>$	<code>&gt;</code>	$=$	<code>=</code>
$\leq$	<code>\leq</code> or <code>\le</code>	$\geq$	<code>\geq</code> or <code>\ge</code>	$\neq$	<code>\neq</code> or <code>\ne</code>
$\ll$	<code>\ll</code>	$\gg$	<code>\gg</code>	$\equiv$	<code>\equiv</code>
$\subset$	<code>\subset</code>	$\supset$	<code>\supset</code>	$\approx$	<code>\approx</code>
$\subseteq$	<code>\subseteq</code>	$\supseteq$	<code>\supseteq</code>	$\cong$	<code>\cong</code>
$\in$	<code>\in</code>	$\ni$	<code>\ni</code> , <code>\owns</code>	$\notin$	<code>\notin</code>
$ $	<code>\mid</code>	$\dagger$	<code>\nmid</code>	$\parallel$	<code>\parallel</code>
$\perp$	<code>\perp</code>	$:$	<code>:</code>	$\propto$	<code>\propto</code>
$+$	<code>+</code>	$-$	<code>-</code>	$\circ$	<code>\circ</code>
$\pm$	<code>\pm</code>	$\mp$	<code>\mp</code>	$*$	<code>\ast</code>
$\cdot$	<code>\cdot</code>	$\div$	<code>\div</code>	$\bullet$	<code>\bullet</code>
$\times$	<code>\times</code>	$\setminus$	<code>\setminus</code>	$\dagger$	<code>\dagger</code>
$\cup$	<code>\cup</code>	$\cap$	<code>\cap</code>	$\ddagger$	<code>\ddagger</code>
$\vee$	<code>\vee</code> , <code>\lor</code>	$\wedge$	<code>\wedge</code> , <code>\land</code>		

Table 4: Variable-Sized Operators.

$\sum$	<code>\sum</code>	$\bigcup$	<code>\bigcup</code>	$\bigvee$	<code>\bigvee</code>
$\prod$	<code>\prod</code>	$\bigcap$	<code>\bigcap</code>	$\bigwedge$	<code>\bigwedge</code>
$\int$	<code>\int</code>	$\oint$	<code>\oint</code>		

Table 5: Arrows.

$\leftarrow$	<code>\leftarrow</code> or <code>\gets</code>	$\longleftarrow$	<code>\longleftarrow</code>	$\uparrow$	<code>\uparrow</code>
$\rightarrow$	<code>\rightarrow</code> or <code>\to</code>	$\longrightarrow$	<code>\longrightarrow</code>	$\downarrow$	<code>\downarrow</code>
$\leftrightarrow$	<code>\leftrightarrow</code>	$\longleftrightarrow$	<code>\longleftrightarrow</code>	$\updownarrow$	<code>\updownarrow</code>
$\Leftarrow$	<code>\Leftarrow</code>	$\Lleftarrow$	<code>\Lleftarrow</code>	$\Uparrow$	<code>\Uparrow</code>
$\Rightarrow$	<code>\Rightarrow</code>	$\Rrightarrow$	<code>\Rrightarrow</code>	$\Downarrow$	<code>\Downarrow</code>
$\Leftrightarrow$	<code>\Leftrightarrow</code>	$\Leftrightarrow$	<code>\Leftrightarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
$\mapsto$	<code>\mapsto</code>	$\longmapsto$	<code>\longmapsto</code>		
$\iff$	<code>\iff</code> (bigger spaces)				

Table 6: Log-like Symbols.

<code>\arccos</code>	<code>\cos</code>	<code>\coth</code>	<code>\det</code>	<code>\gcd</code>	<code>\ln</code>	<code>\min</code>	<code>\sinh</code>
<code>\arcsin</code>	<code>\cosh</code>	<code>\csc</code>	<code>\dim</code>	<code>\lim</code>	<code>\log</code>	<code>\sec</code>	<code>\tan</code>
<code>\arctan</code>	<code>\cot</code>	<code>\deg</code>	<code>\exp</code>	<code>\lg</code>	<code>\max</code>	<code>\sin</code>	<code>\tanh</code>

Calling the above “symbols” may be a bit misleading. Each log-like symbol merely produces the eponymous textual equivalent, but with proper surrounding spacing.

Table 7: Delimiters.

$($	<code>(</code>	$)$	<code>)</code>	$\uparrow$	<code>\uparrow</code>	$\Uparrow$	<code>\Uparrow</code>
$[$	<code>[</code> or <code>\lbrack</code>	$]$	<code>] or \rbrack</code>	$\downarrow$	<code>\downarrow</code>	$\Downarrow$	<code>\Downarrow</code>
$\{$	<code>\{</code> or <code>\lbrace</code>	$\}$	<code>\} or \rbrace</code>	$\updownarrow$	<code>\updownarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
$\langle$	<code>\langle</code>	$\rangle$	<code>\rangle</code>	$ $	<code>  or \vert</code>	$\ $	<code>\  or \Vert</code>
$\lfloor$	<code>\lfloor</code>	$\rfloor$	<code>\rfloor</code>	$\lceil$	<code>\lceil</code>	$\rceil$	<code>\rceil</code>
$/$	<code>/</code>	$\backslash$	<code>\backslash</code>	.	(dual. empty)		

Table 8: Miscellaneous Symbols.

$\dots$	<code>\dots</code>	$\cdots$	<code>\cdots</code>	$\vdots$	<code>\vdots</code>	$\ddots$	<code>\ddots</code>
$\infty$	<code>\infty</code>	$\imath$	<code>\imath</code>	$\jmath$	<code>\jmath</code>	$\ell$	<code>\ell</code>
$\forall$	<code>\forall</code>	$\exists$	<code>\exists</code>	$\partial$	<code>\partial</code>	$\nabla$	<code>\nabla</code>
$'$	<code>'</code>	$\prime$	<code>\prime</code>	$\emptyset$	<code>\emptyset</code>	$\angle$	<code>\angle</code>

Table 9: Non-Mathematical Symbols.

These symbols can also be used in text mode.

†	<code>\dag</code>	§	<code>\S</code>	©	<code>\copyright</code>	®	<code>\textregistered</code>
‡	<code>\ddag</code>	¶	<code>\P</code>	£	<code>\pounds</code>	%	<code>\%</code>

Table 10: Math Alphabets.

Example	Command	Required package
ABCDEabcde1234	<code>\mathrm{ABCDE abcde 1234}</code>	
<i>ABCDEabcde1234</i>	<code>\mathit{ABCDE abcde 1234}</code>	
<i>ABCDEabcde1234</i>	<code>\mathnormal{ABCDE abcde 1234}</code>	
<i>ABCDE</i>	<code>\mathcal{ABCDE}</code>	
$\frac{A}{B}CDEabcde1234$	<code>\mathfrak{ABCDE abcde 1234}</code>	<i>amsfonts</i> or <i>amssymb</i>
$\mathbb{A}BCDE$	<code>\mathbb{A}BCDE</code>	<i>amsfonts</i> or <i>amssymb</i>