to0pt Math into LATEX An Introduction to LATEX and A Short Course

to0pt 1pt plus 1fill1pt plus 1fillThis book is dedicated to those who worked so hard
and for so long to bring these important tools to us:

The LATEX3 team
and in particular
Frank Mittelbach (project leader) and David Carlisle

The team
and in particular
Michael J. Downes (project leader) and David M. Jones

to0pt
to0pt height 600pt width .5pt
to 0ptto600pt

eorge Grätzer

ath into LATEX

An Introduction to LATEX and

BIRKHÄU SER
to 17pc BOSTON* BASEL*BERLIN

to**Library of Congress Cataloging-in-Publication Data**

Grätzer, George A.
Math into LaTeX : an introduction to LaTeX and AMS-LaTeX /
George Grätzer
p. cm.
Includes index.
ISBN 0-8176-3805-9 (pbk. : alk. paper)
1. AMS-LaTeX. 2. Mathematics printingmComputer programs.
3. Computerized typesetting. I. Title.

to 317pt Z253.4A65G69 199595-36881 to 317pt 688.2j2544536mdc20CIP

Contents

List of Tables

Chapter

Preface

It is indeed a lucky author who is given the opportunity to completely rewrite a book barely a year after its publication. Writing about software affords such opportunities (especially if the original edition sold out), since the author is shooting at a moving target.

and improved dramatically with the release of the new standard (called LATEX2e) in June of 1994 and the revision of (version 1.2) in February of 1995. The change in is profound. LATEX2e made it possible for to join the world. One of the main points of the present book is to make this clear. This book introduces as a tool for mathematical typesetting, and treats as a set of enhancements to the standard , to be used in conjunction with hundreds of other LATEX2e enhancements. I am not a expert. Learning the mysteries of the system has given me great respect for those who crafted it: Donald Knuth, Leslie Lamport, Michael Spivak, and others did the original work; David Carlisle, Michael J. Downes, David M. Jones, Frank Mittelbach, Rainer Schöpf, and many others built on the work of these pioneers to create the new and . Many of these experts and a multitude of others helped me while I was writing this book. I would like to express my deepest appreciation and heartfelt thanks to all who gave their time so generously. Their story is told in the Afterword.

This version was prepared for inclusion with the OzTEXdistribution. The book was designed by Merry Sawdey with the Adobe Gaillard font. Since this font is not widely available, I substituted for it the Computer Modern font. The result is not as pleasing, but certainly readable. If you want to see the same pages as originally designed, you may find the file `mil.pdf` on the CTAN (Comprehensive TeX Archive Network) in the

```
/tex-archive/info/mil
```

directory. It is in Adobe Portable Document Format, so all the fonts and design features are the same as in the published book. Read `mil.txt` for instructions.

Andrew Trevorrow suggested to keep the size of this distribution as small as possible, so I deleted all the illustrations from these pages. In particular, pages article1 and article2 are the typeset version of the sample article `intrart.tex` (as illustrations) ; so you are requested to typeset this article, and insert the two typeset pages in this document.

Please send correctionsnand suggestions for improvementsnto me at the following address:

Department of Mathematics
University of Manitoba
Winnipeg MB, R3T 2N2
Canada
e-mail: gratzer@ccu.umanitoba.ca

Chapter

Introduction

Is this book for you?

This book is for the mathematician, engineer, scientist, or technical typist who wants to write and typeset articles containing mathematical formulas but does not want to spend much time learning how to do it.

I assume you are set up to use , and you know how to use an editor to type a document, such as:

```
\documentclassarticle
\begindocument
The square root of two: $\sqrt2$.  I can type math!
\enddocument
```

I also assume you know how to typeset a document, such as this example, with to get the printed version:

The square root of two: . I can type math! and you can view and print the typeset document.

And what do I promise to deliver? I hope to provide you with a solid foundation in , the enhancements, and some standard enhancements, so typing a mathematical document will become second nature to you.

How to read this book?

Part I gives a short course in . Read it, work through the examples, and you are ready to type your first paper. Later, at your leisure, read the other parts to become more proficient.

The rest of this section introduces , LATEX, and , and then outlines what is in this book. If you already know that you want to use to typeset math, you may choose to skip it.

, , and

is a typesetting language created by Donald E. Knuth; it has extensive capabilities to typeset math. is an extension of designed by Leslie Lamport; its major features include

a strong focus on document structure and the logical markup of text;

automatic numbering and cross-referencing.

distills the decades-long experience of the American Mathematical Society () in publishing mathematical journals and books; it *adds to* a host of features related to mathematical typesetting, especially the typesetting of multiline formulas and the production of finely-tuned printed output.

Articles written in (and) are accepted for publication by an increasing number of journals, including all the journals of the .

Look at the typeset sample articles: sampart.tex (in Appendix C) and intrart.tex (on pages article1marticle2). You can begin creating such high-quality typeset articles after completing Part I.

What is document markup?

Most word processing programs are WYSIWYG (what you see is what you get); as you work, the text on the computer monitor is shown, more or less, as itjll look when printed. Different fonts, font sizes, italics, and bold face are all shown.

A different approach is taken by a *markup language*. It works with a text editor, an editing program that shows the text, the *source file*, on the computer monitor with only one font, in one size and shape. To indicate that you wish to change the font in the printed copy in some way, you must kmark upl the source file. For instance, to typeset the phrase kSmall Capsl in small caps, you type

```
\textscSmall Caps
```

The textsc command is a markup command, and the printed output is
SMALL CAPS
is a markup language; is another markup language, an extension of . Actually, itjs quite easy to learn how to mark up text. For another example, look at the abstract of the sampart.tex sample article (page 364), and the instruction

```
\emphcomplete-simple distributive lattices
```

to emphasize the phrase kcomplete-simple distributive latticesl, which when typeset looks like *complete-simple distributive lattices*

On pages 364m371 we show the source file and the typeset version of the sampart.tex sample article together. The markup in the source file may appear somewhat bewildering at first, especially if you have previously worked on a WYSIWYG word processor. The typeset article is a rather pleasing-to-the-eye polished version of the sample marked up material.

Of course, markup languages have always dominated typographic work of high quality. On the Internet, the most trendy communications on the World Wide Web are written in a markup language called HTML (HyperText Markup Language).

has excellent typesetting capabilities. It deals with mathematical formulas as well as text. To get in a formula, type `\sqrta^2 + b^2`. There is no need to worry about how to construct the square root symbol that covers .

A tremendous appeal of the language is that a source file is *plain text*, sometimes called an ASCII file.

ASCII stands for American Standard Code for Information Interchange. Therefore articles containing even the most complicated mathematical expressions can be readily transmitted *electronically*nto colleagues, coauthors, journals, editors, and publishers.

is *platform independent*. You may type the source file on a , and your coauthor may make improvements to the same file on an personal computer; the journal publishing the article may use a DEC minicomputer. The form of , a richer version, used to typeset documents is called *Plain* . Ijll not try to distinguish between the two.

however, is a programming language, meant to be used by programmers.

is much easier and safer to work with than ; it has a number of built-in safety features and a large set of error messages.

building on , provides the following additional features:

An article is divided into *logical units* such as an abstract, sections, theorems, a bibliography, and so on. The logical units are typed separately. After all the units have been typed, organizes the *placement* and *formatting* of these elements.

Notice line 4 of the source file of the sampart.tex sample article

```
\documentclassamsart
```

on page 364. Here the general design is specified by the amsart kdocument class", which is the article document class. When submitting your article to a journal that is equipped to handle articles (and the number of such journals is increasing rapidly), only the *name of the document class* is replaced by the editor to make the article conform to the design of the journal.

relieves you of tedious *bookkeeping chores*. Consider a completed article, with theorems and equations numbered and properly cross-referenced. Upon final reading, some changes must be madenfor example, section 4 has to be placed after section 7, and a new theorem has to be inserted somewhere in the middle. Such a minor change used to be a major headache! But with , it becomes almost a pleasure to make such changes. automatically redoes all the numbering and cross-references.

Typing the same *bibliographic references* in article after article is a tedious chore. With you may use , a program that helps you create and maintain bibliographic databases, so references need not be retyped for each article. will select and format the needed references from the databases.

All the features of are made available by the LaTeX format, which you should use to typeset the sample documents in this book.

The enhanced the capabilities of in three different areas. You decide which of these are important to you.

1. **Math enhancements.** The first area of improvement is a wide variety of tools for typesetting math. provides

   excellent tools to deal with *multiline math formulas* requiring special alignment. For instance, in the following formula, the equals sign (=) is vertically aligned and so are the explanatory comments:

[Sorry. Ignored `\begincmr ... \endcmr`]

numerous constructs for typesetting math, exemplified by the following formula:

[Sorry. Ignored `\begincmr ... \endcmr`]

special spacing rules for dozens of formula types, for example

*a***b**

If the above formula is typed inline, it becomes: *a***b**; the spacing is automatically changed.

multiline ksubscriptsl as in

user-defined symbols for typesetting math, such as

[Sorry. Ignored `\begincmr ... \endcmr`]

formulas numbered in a variety of ways:

automatically,

manually (by tagging),

by groups, with a group number such as (2), and individual numbers such as (2a), (2b), and so on.

the proof environment and three theorem styles; see the sampart.tex sample article (pages 361m363) for examples.

2. Document classes. provides a number of document classes, including the article document class, amsart, which allows the input of the title page information (author, address, e-mail, and so on) as separate entities. As a result, a journal can typeset even the title page of an article according to its own specifications without having to retype it.

Many users prefer the visual design of the amsart document class to the simpler design of the classical article document class.

3. Fonts. There are hundreds of binary operations, binary relations, negated binary relations, bold symbols, arrows, extensible arrows, and so on, provided by , which also makes available additional math alphabets such as Blackboard bold, Euler Fraktur, Euler Script, and math bold italic. Here are just a few examples:

[Sorry. Ignored `\begincmr ... \endcmr`]

We have barely scratched the surface of this truly powerful set of enhancements.

What is in the book?

Part I (Chapter 1) will help you get started quickly with ; if you read it carefully, youjll certainly be ready to start typing your first article and tackle in more depth.
Part I guides you through:

marking up text, which is quite easy;

marking up math, which is not so straightforward (four sections ease you into mathematical typesetting: the first discusses the basic building blocks; the second shows how to build up a complicated formula in simple steps; the third is a formula gallery; and the fourth deals with equations and multiline formulas);

the anatomy of an article;

how to set up an article template;

typing your first article.

Part II introduces the two most basic skills in depth: *typing text* and *typing math*.

Chapters 2 and 3 introduce *text* and *displayed text*. Chapter 2 is very important; when typing your document, you spend most of your time typing text. The topics covered include special characters and accents, hyphenation, fonts, and spacing. Chapter 3 covers displayed text including *lists* and *tables*, and for the mathematician, proclamations (theorem-like structures) and proofs.

Chapters 4 and 5 discuss *math* and *displayed math*. Of course, typing math is the heart of any mathematical typesetting system. Chapter 4 discusses this topic in detail, including basic constructs, operators, delimiters, building new symbols, fonts, and grouping of equations. Chapter 5 presents one of the major contributions of : aligned multiline formulas. This chapter also contains other multiline formulas.

Part III discusses the parts of a document. In Chapter 6, you learn about the structure of a document. The most important topics are sectioning and cross-referencing. In Chapter 7, the standard document classes are presented: article, report, book, and letter, along with a description of the standard distribution. In Chapter 8, the document classes are discussed. In particular, the title page information for the amsart document class and a description of the standard distribution is presented.

Part IV (Chapter 9) introduces techniques to *customize* to speed up typing source files and typesetting of documents. really speeds up with user-defined commands, user-defined environments, and custom formats. Youjll learn how parameters that effect the behavior of are stored in *counters* and *length commands*, how to change them, and how to design custom lists.

In Part V (Chapters 10 and 11), wejll discuss two programs: and *MakeIndex* that complement the standard distribution; they give a helping hand in making large bibliographies and indices.

Appendices A and B will probably be needed quite often in your work: they contain *math symbol tables* and *text symbol tables*.

Appendix C presents the sample article, sampart.tex, first in typeset form (pages 361m363), then in kmixedl form, showing the source file and the typeset article together (pages 364m371). You can learn a lot about and just by reading the source file a paragraph at a time and see how that paragraph looks typeset. Then Appendix D rewrites this sample article utilizing the user-defined commands collected in lattice. sty of section 9.5.

Appendix E relates some historical background material on : how did it develop and how does it work. Appendix F is a brief introduction to the use of PostScript fonts in a document. Appendix G shows how you can obtain and , and how you can keep them up-to-date through the Internet. A work session is reproduced (in part) using kanonymous ftpl (file transfer protocol).

Appendix H will help those who have worked with (Plain) , version 2.09, , or version 1.1, programs from which the new and developed. Some tips are given to smooth the transition to the new and .

Finally, Appendix I points the way for further study. The most important book for extending and customizing is , the work of Michel Goossens, Frank Mittelbach, and Alexander Samarin.

Typographical conventions

This book is about typesetting math in . So often you are told to type in some material and shown how itjll look typeset.

```
I use this font, Computer Modern typewriter style, to show
what
you have to type.  All characters have the same width so
it's
easy to distinguish it from the other fonts used in this
book.
```

I use the same font for commands (parbox), environments (align), documents (sampart.tex), document classes (article), directories and folders (work), counters (tocdepth), and so on.

The names of packages (amsmath), extensions of , are printed in a sans serif font, as traditional.

If the typeset material is a separate paragraph (or paragraphs), I make it visually stand out by adding the little corner symbols on the margin to offset it.

When I give explanations in the text: kTo get iiffj type iffl, I set off the typeset version with single quotes. If the typeset material is a math formula, there is no need for the single quotes, so they are dropped, as in kto get , type `$\sqrta$`.

Commands are introduced, as a rule, with examples:

```
\\[0.5in]
```

However, sometimes itjs necessary to more formally define the syntax of a command. For instance:

`\\[`length]

where length is a *placeholder*: it represents the length you have to type in. I use the Computer Modern typewriter style italic font for placeholders.

Chapter 1

[

Typing your first article]1. Typing
your first article

In this chapter, youjll start writing your first article. All you have to do is to type the (electronic) *source file*; does the rest.

In the next few sections, Ijll introduce you to the most important commands for typesetting text and math by working through examples. Go to the latter parts of this book for more detail.

The source file is made up of *text*, *math* (for instance,), and *instructions* to LATEX. This is how you type the last sentence:

```
The source file is made up of \emphtext, \emphmath (for
instance, $\sqrt5$), and \emphinstructions to \LaTeX.
```

In this sentence,

```
The source file is made up of \emphtext, \emphmath (for
instance,
```

is text,

```
$\sqrt5$
```

is math, and
`emph`*`text`*
is an instruction (a command). Commands, as a rule, start with a backslash and are meant to instruct ; this particular command, emph, emphasizes *text* given as its *argument* (between the braces). Another kind of instruction is called an *environment*. For instance,

```
\beginflushright
```

and

```
\endflushright
```

bracket a flushright environmentnwhat is typed inside this environment comes out right justified (lined up against the right margin) in the printed form.
In practice, text, math, and instructions are intertwined. For example,

```
\emphMy first integral $\int \zeta^2(x) \, dx$
```

which produces *My first integral*
is a mixture of all three. Nevertheless, to some extent I try to introduce the three topics: typing text, typing math, and giving instructions to (commands and environments) as if they were separate topics.
I introduce the basic features of by working with a number of sample documents. If you wish to obtain these documents electronically, create a subdirectory (folder) on your computer, say, ftp, and proceed to download all the sample files as described in section G.6. Also create a subdirectory (folder) called work. Whenever you want to use one of these documents, copy it from the ftp subdirectory (folder) to the work subdirectory (folder), so that the original remains unchanged; alternatively, type in the examples as shown in the book. *In this book, the* ftp *directory and the* work *directory will refer to the directories (folders) you hereby create without further elaboration.* In this distribution, the sample documents relevant to Part I can be found in the folder sample_files.

1.1 Typing a very short karticlel

First we discuss how to use the keyboard in , and then type a very short karticlel containing only text.

1.1.1 The keyboard

In , to type text, use the following keys:

`a-z` **`A-Z 0-9`**

`+ = / ( ) [ ]`

You may also use the punctuation marks

and the spacebar, the tab key, and the return (or enter) key.
There are thirteen special keys (on most keyboards):

`# $ % & ~ _ ^ @ " |`

1.1.2 Your first note

We start our discussion on how to type a note in with a simple example. Suppose you want to use to produce the following:
[Sorry. Ignored `\beginloutput ... \endloutput`]
Create a new file in the work directory with the name note1.tex and type the following (if you prefer not to type it, copy the file from the ftp directory or the sample_files folder; see page ftp):

```
% Sample file: note1.tex
% Typeset with LaTeX format
\documentclassarticle

\begindocument
It is of some concern to me   that
the terminology used in  multi-section
 math courses is not uniform.

In several sections of the course on
matrix theory, the  term
 ``hamiltonian-reduced'' is used.
  I, personally, would rather call these ``hyper-simple''. I
invite others to comment on this  problem.

Of special concern to me is the terminology in the course
by Prof.~Rudi Hochschwabauer.
  Since his field is new, there is
 no accepted
terminology.   It is imperative
that we arrive at a satisfactory solution.
\enddocument
```

1.1.3 Lines too wide

reads the text in the source file one line at a time and when the end of a paragraph is reached, typesets it (see section E.2 for a more detailed discussion). Most of the time, there is no need for corrective action. Occasionally, however, gets into trouble splitting the paragraph into typeset lines. To illustrate this, modify note1.tex: in the second sentence replace kterml by kstrange terml, and in the fourth sentence delete k`Rudi` l. Save this modified file with the name note1b.tex in the work directory. (Youjll find note1b.tex in the ftp directory or the sample_files foldernsee page ftp).

Typesetting note1b.tex, you get:
[Sorry. Ignored `\beginloutput ... \endloutput`]
The first line of paragraph two is about 1/4 inch too wide. The first line of paragraph three is even wider. On your monitor, displays the message:

```
Overfull \hbox (15.38948pt too wide) in paragraph at lines 10--15
[]\OT1/cmr/m/n/10 In sev-eral sec-tions of the course on ma-trix
the-ory, the strange term ``hamiltonian-
 []
Overfull \hbox (23.27834pt too wide) in paragraph at lines 16--22
[]\OT1/cmr/m/n/10 Of spe-cial con-cern to me is the ter-mi-nol-ogy
in the course by Prof. Hochschwabauer.
 []
```

Youjll find the same message in the log file note1b.log (see texfiles).
The reference

```
Overfull \hbox (15.38948pt too wide) in paragraph at lines 10--15
```

is made to paragraph two (lines 10m15); the typeset version has a line (line number unspecified within the typeset paragraph) which is 15.38948pt too wide. uses *points* (pt) to measure distances; there are about 72 points to an inch. The next two lines

```
[]\OT1/cmr/m/n/10 In sev-eral sec-tions of the course on ma-trix
the-ory, the strange term ``hamiltonian-
```

identify the source of the problem: would not hyphenate

hamiltonian-reduced,

since it only (automatically) hyphenates a hyphenated word only at the hyphen. You may wonder what \OT1/cmr/m/n/10 signifies. It says that the current font is the Computer Modern roman font at size 10 points (see section 2.6.1).
The second reference

```
Overfull \hbox (23.27834pt too wide) in paragraph at lines 16--22
```

is made to paragraph three (lines 16m22). The problem is with the word

Hochschwabauer

which the hyphenation routine of canjt handle. (If you use a German hyphenation routine, itjll have no difficulty hyphenating Hochschwabauer.)

If you encounter such a problem, try to reword the sentence or add an optional hyphen \-, which encourages to hyphenate at this point if necessary. For instance, rewrite Hochschwabauer as

```
Hoch\-schwabauer
```

and the second problem goes away.

Sometimes a small horizontal overflow is difficult to spot. The draft document class option is very useful in this case: itjll paint an ugly slug on the margin to mark an overfull line; see sections 7.1.2 and 8.4 for document class options. You may invoke this option by changing the documentclass line to

```
\documentclass[draft]article
```

Youjll find this version of note1b.tex under the name noteslug.tex in the ftp directory or the sample_files folder.

1.1.4 More text features

Next youjll produce the following note in :

[Sorry. Ignored \beginloutput ... \endloutput]

Type in the following source file, save it as note2.tex in the work directory (youjll also find note2.tex in the ftp directory or the sample_files folder):

```
% Sample file: note2.tex
% Typeset with LaTeX format
\documentclassarticle

\begindocument
\beginflushright
   \today
\endflushright
\textbfFrom the desk of George Gr\"atzer\\[10pt]

February~7--21 \emphplease use my temporary e-mail address:
\begincenter
   \texttt George\_Gratzer@umanitoba.ca
\endcenter
\enddocument
```

This note introduces several additional features of :

The today command displays todayjs date.

Use environments to *right justify* or *center* text. Use the emph command to *emphasize* text; the text to be emphasized is surrounded by and . Use textbf for bold text; the text to be made bold is also surrounded by and . Similarly, use texttt for typewriter style text. emph, textbf, and texttt are examples of *commands with arguments*. Note that command names are case sensitive; do not type Textbf or TEXTBF in lieu of textbf.

commands (almost) always start with followed by the command name, for instance, \textbf. The command name is terminated by the first non-alphabetic character.

Use double hyphens for number ranges (en-dash): 7m21 prints i7m21j; use triple hyphens (n) for the kem-dashl punctuation marknsuch as the one in this sentence.

If you want to create additional space between lines (as in the last note under the line **From the desk ?**), use the command \\[10pt] with an appropriate amount of vertical space. (\\ is the newline commandnsee section 2.7.1; the variant used in the above example is the newline with additional vertical space.) The distance may be given in points, centimeters (cm), or inches (in). (72.27 points make an inch.)

There are special rules for *accented characters* and some *European characters*. For instance, iäj is typed as \"a. Accents are explained in section 2.4.6 (see also the tables in ATS).

Youjll seldom need to know more than this about typing text. For more detail, however, see Chapters 2 and 3. All text symbols are organized into tables in ATS.

1.2 Typing math

Now you can start mixing text with math formulas.

1.2.1 The keyboard

In addition to the regular text keys (kbsh), three more keys are needed to type math:

| < >

|(is the shifted key on many keyboards.)

1.2.2 A note with math

Youjll begin typesetting math with the following note:

[Sorry. Ignored \beginloutput ... \endloutput]

To create the source file for this mixed math and text note, create a new document with an editor. Name it math.tex, place it in the work directory, and type in the following source filenor copy math.tex from the ftp directory or the sample_files folder:

```
% Sample file: math.tex
% Typeset with LaTeX format
\documentclassarticle

\begindocument
In first year Calculus, we   define intervals  such as
$(u, v)$ and $(u, \infty)$.  Such an interval is a
\emphneighborhood of  $a$
if  $a$ is in the interval.  Students should
realize that  $\infty$ is only a
symbol, not a number.  This is important since
we soon introduce concepts
 such as $\lim_x \to \infty f(x)$.

When we introduce the derivative
\[
   \lim_x \to a \fracf(x) - f(a)x - a,
\]
we assume that the function is defined and continuous
in a neighborhood of  $a$.
\enddocument
```

This note introduces the basic techniques of typesetting math with :

There are two kinds of math formulas and environments: *inline* and *displayed*.

Inline math environments open and close with $.

Displayed math environments open with \[and close with \].

ignores the spaces you insert in math environments with two exceptions: spaces that delimit commands (see section 2.3.1) and spaces in the argument of commands that temporarily revert into text mode. (mbox is such a command; see section 4.5.) Thus spacing in math is important only for the readability of the source file. To summarize:

[Sorry. Ignored `\beginRule ... \endRule`]

The same formula may be typeset differently depending on which math environment itjs in. The expression *x**a* is typed as a subscript to in the inline formula , typed as `$\lim_x \to a f(x)$`, but itjs placed below in the displayed version:

typed as

```
\[
   \lim_x \to a f(x)
\]
```

A math symbol is invoked by a command. Examples: the command for * is infty and the command for * is to. The math symbols are organized into tables in AMS.

To access most of the symbols listed in AMS by name, use the amssymb package; in other words, the article should start with

```
\documentclassarticle
\usepackageamssymb
```

The amssymb package loads the amsfonts package, which contains the commands for using the AMSFonts (see section 4.14.2).

Some commands such as sqrt need *arguments* enclosed in and . To typeset , type `$\sqrt5$`, where sqrt is the command and 5 is the argument. Some commands need more than one argument. To get

type

```
\[
   \frac3+x5
\]
```

frac is the command, 3+x and 5 are the arguments.

There are many mistakes you can make, even in such a simple note. Youjll now introduce mistakes in math.tex, by inserting and deleting % signs to make the mistakes visible to one at a time. Recall that lines starting with % are ignored by . Type the following source file, and save it under the name mathb.tex in the work directory (or copy over the file mathb.tex from the ftp directory or the sample_files folder).

```
% Sample file: mathb.tex
% Typeset with LaTeX format
\documentclassarticle

\begindocument
In first year Calculus, we   define intervals  such as
%$(u, v)$ and $(u, \infty)$.  Such an interval is a
 $(u, v)$ and    (u, \infty)$.  Such an interval is a
 \emphneighborhood of $a$
if $a$ is in the interval.  Students should
realize that  $\infty$ is only a
symbol, not a number.  This is important since
we soon introduce concepts
 such as $\lim_x \to \infty f(x)$.
%such as $\lim_x \to \infty f(x)$.

When we introduce the derivative
\[
    \lim_x \to a \fracf(x) - f(a)x - a
   %\lim_x \to a \fracf(x) - f(a)  x - a
\]
we assume that the function is defined and continuous
in a neighborhood of  $a$.
\enddocument
```

Exercise 1 Note that in line 8, the second $ is missing. When you typeset the mathb.tex file, sends the error message:

```
! Missing $ inserted.
<inserted text>
 $
l.8 ..., v)$ and    (u, \infty
  )$.  Such an interval is a
?
```

Since you omitted `$`, reads `(u, \infty)` as text; but the infty command instructs to typeset a math symbol, which can only be done in math mode. So offers to put a $ in front of infty. suggests a cure, but in this example it comes too late. Math mode should start just prior to (u.

Exercise 2 In the mathb.tex file, delete % at the beginning of line 7 and insert a % at the beginning of line 8 (this eliminates the previous error); delete % at the beginning of line 15 and insert a % at the beginning of line 14 (this introduces a new error: the closing brace of the subscript is missing). Save the changes, and typeset the note. You get the error message:

```
! Missing   inserted.
<inserted text>

l.15 ...im_x \to \infty f(x)$
   .
?
```

is telling you that a closing brace is missing, but itjs not sure where. noticed that the subscript started with and it reached the end of the math formula before finding . You must look in the formula for a that is not closed, and close it with .

Exercise 3 Delete % at the beginning of line 14 and insert a % at the beginning of line 15, which removes the last error, and delete % at the beginning of line 20 and insert a % at the beginning of line 19 (introducing the final error: deleting the closing brace of the first argument of frac). Save and typeset the file. You get the error message:

```
! LaTeX Error: Bad math environment delimiter.

l.21 \]
```

There is a bad math environment delimiter in line 21, namely, `\]`. So the reference to

```
! Bad math environment delimiter.
```

is to the displayed formula. Since the environment delimiter is correct, something must have gone wrong with the displayed formula. This is what happened: was trying to typeset

```
\lim_x \to a \fracf(x) - f(a)  x - a
```

but frac needs two arguments. found `f(x) - f(a)  x - a` as the first argument. While looking for the second, it found `\]`, which is obviously an error (it was looking for a `)`.

1.2.3 Building blocks of a formula

A formula is built up from various types of components. We group them as follows:

- Arithmetic
- Subscripts and superscripts
- Accents
- Binomial coefficients
- Congruences
- Delimiters
- Operators
- Ellipses
- Integrals
- Matrices
- Roots
- Sums and products
- Text

Some of the commands in the following examples are defined in the amsmath package; in other words, to typeset these examples with the article document class, the article should start with

```
\documentclassarticle
\usepackageamssymb,amsmath
```

Arithmetic The arithmetic operations *a*+*b*, *a*?*b*, ?*a*, *a*/*b*, *ab* are typed as expected:

```
$a + b$, $a - b$, $-a$, $a / b$, $a b$
```

If you wish to use * or * for multiplication, as in *a***b* or *a***b*, use cdot or times, respectively. The expressions *a***b* and *a***b* are typed as follows:

```
$a \cdot b$  $a \times b$
```

Displayed fractions, such as

are typed with frac:

```
\[
   \frac1 + 2xx + y + xy
\]
```

The frac command is seldom used inline.

Subscripts and superscripts Subscripts are typed with _ (underscore) and superscripts with ^ (caret). Remember to enclose the subscripts and superscripts with and . To get , type the following characters:

Go into inline math mode:	`$`
type the letter a:	a
subscript command:	_
bracket the subscripted 1:	1
exit inline math mode:	`$`

that is, type `$a_1$`. Omitting the braces in this example causes no harm; however, to get , you *must* type `$a_10$`. Indeed, `$a_10$` prints . Further examples: , , are typed as

```
$a_i_1$, $a^2$, $a^i_1$
```

Accents The four most often used math accents are:

atyped as`$\bara$`
âtyped as`$\hata$`
ãtyped as`$\tildea$`
-*a*typed as`$\veca$`

Binomial coefficients The amsmath package provides the binom command for binomial coefficients.For example, *ab*+*c* is typed inline as

```
$\binomab + c$
```

whereas the displayed version

is typed as

```
\[
   \binomab + c \binom\fracn^2 - 12n + 1
\]
```

Congruences The two most important forms are:

*a***v**	typed as `$a \equiv v \pmod\theta$`
*a***v**	typed as `$a \equiv v \pod\theta$`

The second form requires the amsmath package.

Delimiters These are parenthesis-like symbols that vertically expand to enclose a formula. For example: , which is typed as `$(a + b)^2$`, and

which is typed as

```
\[
   \left( \frac1 + x2 + y^2 \right)^2
\]
```

contain such delimiters. The left(and right) commands tell to size the parentheses correctly (relative to the size of the symbols inside the parentheses). Two further examples:

would be typed as:

```
\[
  \left| \fraca + b2 \right|,
 \quad \left\| A^2 \right\|
\]
```

where quad is a spacing command (see section 4.11 and AMS).

Operators To typeset the sine function sin*x*, type: `$\sin x$`. Note that typing `$sin x$` prints: *sinx*, where the typeface of *sin* is wrong, as is the spacing. calls sin an *operator*; there are a number of operators listed in section 4.7.1 and AMS. Some are just like sin; others produce a more complex display:

which is typed as

```
\[
   \lim_x \to 0 f(x) = 0
\]
```

Ellipses The ellipsis (?) in math sometimes needs to be printed as low dots and sometimes as (vertically) centered dots. Print low dots with the ldots command as in , typed as

```
$F(x_1, x_2, \ldots , x_n)$
```

Print centered dots with the cdots command as in , typed as

```
$x_1 + x_2 + \cdots + x_n$
```

If you use the amsmath package, there is a good chance that the command dots will print the ellipsis as desired.

Integrals The command for an integral is int; the lower limit is a subscript and the upper limit is a superscript. Example: is typed as

```
$\int_0^\pi \sin x \, dx = 2$
```

, is a spacing command (see section 4.11 and AMS).

Matrices The amsmath package provides you with a matrix environment:

[Sorry. Ignored `\beginmatrix ... \endmatrix`]

which is typed as follows:

```
\[
   \beginmatrix
 a + b + c & uv    & x - y & 27\\
 a + b& u + v & z& 134
   \endmatrix
\]
```

The matrix elements are separated by &; the rows are separated by `\\`. The basic form gives no parentheses; for parentheses, use the pmatrix environment; for brackets, the bmatrix environment; for vertical lines (determinants, for example), the vmatrix environment; for double vertical lines, the Vmatrix environment. For example,

A=

[Sorry. Ignored `\beginpmatrix ... \endpmatrix`]
[Sorry. Ignored `\beginpmatrix ... \endpmatrix`]
is typed as follows:

```
\[
   \mathbfA =
   \beginpmatrix
 a + b + c & uv\\
 a + b & u + v
   \endpmatrix
   \beginpmatrix
 30 & 7\\
 3 & 17
   \endpmatrix
\]
```

Roots sqrt produces the square root; for instance, is typed as

```
$\sqrt5$
```

and is typed as

```
$\sqrta + 2b$
```

The *n*th root, , is done with two arguments:

```
$\sqrt[n]5$
```

Note that the first argument is in brackets [ ]; itjs an *optional argument* (see section 2.3).

Sums and products The command for sum is sum and for product is prod. The examples

are typed as

```
\[
   \sum_i=1^n x_i^2 \qquad \prod_i=1^n x_i^2
\]
```

qquad is a spacing command; it separates the two formulas (see section 4.11 and AMS).
Sums and products are examples of *large operators*; all of them are listed in section 4.8 and AMS. They display in a different style (and size) when used in an inline formula: .

Text Place text in a formula with an mbox command. For instance,

$$a=b \quad \text{by assumption}$$

is typed as

```
\[
   a = b \mbox\qquad by assumption
\]
```

Note the space command qquad in the argument of mbox. You could also have typed

```
\[
   a = b \qquad \mboxby assumption
\]
```

because qquad works in text as well as in math.

If you use the amsmath package, then the text command is available in lieu of the mbox command. It works just like the mbox command except that it automatically changes the size of its argument as required, as in , typed as

```
$a^ \textpower $
```

If you do not want to use the large amsmath package, the tiny amstext package also provides the text command (see section 8.5).

1.2.4 Building a formula step-by-step

It is simple to build up complicated formulas from the components described in buildingBlocks. Take the formula

for instance. You should build this up in several steps. Create a new file in the work directory. Call it formula.tex and type in the lines:

```
% File: formula.tex
% Typeset with LaTeX format
\documentclassarticle
\usepackageamssymb,amsmath
\begindocument
\enddocument
```

and save it. At present, the file has an empty document environment.

The quickest way to create this file is to open math.tex, save it under the new name formula.tex, and delete the lines in the document environment. Then add the line

usepackageamssymb,amsmath Type each part of the formula as an inline or displayed formula so that you can typeset the document and check for errors.
Step 1 Letjs start with :

```
$\left[ \fracn2 \right]$
```

Type this into formula.tex and test it by typesetting the document.
Step 2 Now you can do the sum:

For the superscript, you can cut and paste the formula created in Step 1 (without the dollar signs), to get

```
\[
   \sum_i = 1^ \left[ \fracn2 \right]
\]
```

Step 3 Next, do the two formulas in the binomial:

Type them as separate formulas in formula.tex:

```
\[
   x_i, i + 1^i^2 \qquad \left[ \fraci + 33 \right]
\]
```

Step 4 Now itjs easy to do the binomial. Type the following formula by cutting and pasting the previous formulas: =2pt

```
\[
   \binom x_i,i + 1^i^2  \left[ \fraci + 33 \right]
\]
```

which prints:

Step 5 Next type the formula under the square root as

```
$\mu(i)^ \frac32  (i^2 - 1)$
```

and then the square root as

```
$\sqrt \mu(i)^ \frac32  (i^2 - 1) $
```

Step 6 The two cube roots, and , are easy to type:

```
$\sqrt[3] \rho(i) - 2 $  $\sqrt[3] \rho(i) - 1 $
```

Step 7 So now get the fraction:

typed, cut, and pasted as

```
\[
   \frac \sqrt \mu(i)^ \frac32 (i^2 -1)
    \sqrt[3]\rho(i) - 2 + \sqrt[3]\rho(i) - 1
\]
```

Step 8 Finally, get the formula

by cutting and pasting the pieces together, leaving only one pair of displayed math delimiters:

```
\[
   \sum_i = 1^ \left[ \fracn2 \right]
 \binom x_i, i + 1^i^2
   \left[ \fraci + 33 \right]
 \frac \sqrt \mu(i)^ \frac32 (i^2 - 1)
  \sqrt[3]\rho(i) - 2 + \sqrt[3]\rho(i) - 1
\]
```

Notice the use of

spacing to help distinguish the braces (note that some editors help you balance the braces);

separate lines for the various pieces.

Keep the source file readable. Of course, this is for your benefit, since does not care. It would also accept

```
\[\sum_i=1^\left[\fracn2\right]\binomx_i,i+1^i^2
\left[\fraci+33\right]\frac\sqrt\mu(i)^\frac3
2(i^2-1)\sqrt[3]\rho(i)-2+\sqrt[3]\rho(i)-1\]
```

Problems arise with this haphazard style when you make a mistake. Try to find the error in the next version:

```
\[\sum_i=1^\left[\fracn2\right]\binomx_i,i+1^i^2
\left[\fraci+33\right]\frac\sqrt\mu(i)^\frac3
2(i^2-1)\sqrt[3]\rho(i)-2+\sqrt[3]\rho(i)-1\]
```

(Answer: `\frac32` should be followed by