The LaTeX Mathematics Companion

Helin Gai
Duke University

Coleen's Workgroup

CONTENTS

LIST OF TABLES

CHAPTER 1

An Introduction to Math Typesetting

1.1 A Few Words on Math Typesetting

In the preface to *The TEXbook* [9], Donald Knuth writes: "TEX [is] a new typesetting system intended for the creation of beautiful books—and especially for books that contain a lot of mathematics." And rightly so! Today, most of the world's best-typeset books are created by TEX or its offspring LATEX.

However, writing good mathematics involves a lot of *devotion*. Owning TEX doesn't necessarily follow that you can create the best formulas—you only have the potential to do so. Shredder of the CTEX Community[1] once said, "The most beautiful books are prepared by TEX, and so are the ugliest ones!"

You might ask, "Isn't it true that it is the content that really matters?" Well, yes. But the way you "present" your math is equally important. A well-typeset equation increases the readability, and helps your readers to comprehend your "content" more easily and effectively.

Let's take a look at three examples of the same content, but presented in different ways:

Example 1.1 So far we have used the notation f' to stand for the derivative of the function f. An alternative notation for derivatives was introduced by the German mathematician Wilhelm Gottfried Leibniz (1646–1716). If the variable y depends on the variable x, that is, if y=f(x), then he wrote dy/dx for the derivative, so dy/dx=f'(x).

Example 1.2 So far we have used the notation f' to stand for the derivative of the function f. An alternative notation for derivatives was introduced by the German mathematician Wilhelm Gottfried Leibniz (1646–1716). If the variable y depends on the variable x, that is, if $y = f(x)$, then he wrote dy/dx for the derivative, so $dy/dx = f'(x)$.

Example 1.3 So far we have used the notation f' to stand for the derivative of the function f. An alternative notation for derivatives was introduced by the German mathematician Wilhelm Gottfried Leibniz (1646–1716). If the variable y depends on the variable x, that is, if $y = f(x)$, then he wrote $\mathrm{d}y/\mathrm{d}x$ for the derivative, so

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f'(x).$$

Most people will agree that example 1.3 is the best for three reasons: (1) You can easily distinguish mathematics from the surrounding text; (2) The last equation is "displayed" because it is an important conclusion and by doing so, the readers will be able to locate this conclusion very easily when they need it; (3) The differential sign 'd' is set in upright type, so under no circumstance will '$\mathrm{d}x$' be mistaken for '$d \times x$'.

[1] The CTEX Community is the largest online Chinese TEX community. Its official website is http://www.ctex.org.

1.2 A Few Words on This Book

This book assumes that you already have some LaTeX experience. If you are a *total* green hand, I'd like to recommend the following:

- *The Not So Short Introduction to LaTeX 2ε 4.17*
  This is the best booklet to get started with. It is explicit, easy to read, and most important of all, *short*! You can download it at http://people.ee.ethz.ch/%7Eoetiker/lshort/lshort.pdf.
- *LaTeX — A Document Preparation System*
  Written by Leslie Lamport, father of LaTeX.
- *Guide to LaTeX, Fourth Edition*
  A great book for people of all levels! It is an optional textbook used by Pratt School of Engineering of Duke University.

So technically, my book doesn't talk about basic LaTeX commands, like `\documentclass`. Nor does it cover commands used in text mode only. Instead, it focuses on LaTeX's math mode, telling you at least three things:

- What is the correct way of writing a formula? For example, should you use $\vec{a}$, $\boldsymbol{a}$, or $\mathbf{a}$ to denote a vector?
- What is the "code" that you enter to get the formula in question?
- How can you further improve the appearance of a formula?

Oriented at answering these equations, the book is further divided into four chapters.

Chapter 2 focuses on the standard LaTeX commands and environments. It also covers a large amount of information about the amsmath package designed by the American Mathematical Society. Where necessary, some other essential packages are also introduced. After reading this chapter, you should be able to typeset all the math equations you will need in a "normal" math book/paper/document.

Chapter 3 talks about how you can customize your math formulas. Again, I focus on commands and environments either coming along with LaTeX or provided by the amsmath package. (There is one section that exceeds the limit.)

Chapter 4 gives an overview of over 30 packages that enable you to do more amazing things. But be cautious, some of the packages also create evils.

Chapter 5 gives further directions on resources either in print or on the Internet.

I also really want you to *do* the exercises! I myself have the habit of skipping all the exercises when reading anything *technical*. But I strongly recommend that you do the ones in my book—they're really good. Always remember to load the amsmath and amsfonts packages when you do the exercises (and even when you compile own documents), as most of the them require these two packages.

Finally, enjoy reading this book!

CHAPTER 2

The Essence of Mathematical Typesetting

2.1 Extremely Simple Formulas

The simplest formula is a single letter, like 'x' or a single number, like '3'. The typographic difference between the two is straightforward: the letter 'x' is printed in italic type, while the number '3' is set in an upright font. This demonstrates a very basic rule in math composition: *all variables are printed in italic type, while numerals are always set in upright type.* The good news is that you don't have to pay much attention to the font switching—TEX will handle it for you. All you have to do is to tell TEX that what you are typing *is* a math formula by enclosing it in between special math brackets, the dollar sign '`$`' ("Because mathematics is supposedly expensive," said Knuth). Therefore, in order to get 'x' and '3', simply type '`$x$`' and '`$3$`'.

With this little knowledge, you can already handle a lot of formulas. Here are some examples:

```
$-a+2b=3c-4d(5e+6f)$\\
$(x+y)/(x-y)$\\
$\{a,b,c,d,e\}$
```

$-a+2b=3c-4d(5e+6f)$
$(x+y)/(x-y)$
$\{a,b,c,d,e\}$

I'd like to call your attention to the second example: If you look closely, you'll realize that there is some extra space surrounding the + and − sign, but none around the / sign. That's because TEX regards such expressions as "1 / 2" to be incorrect. Spacing in equations can be rather challenging, but TEX has pretty good mechanism to cope with it automatically. So most of the time, you don't need to bother about that. As a matter of fact, TEX even prevents you from doing stupid things by ignoring any spaces that you put between `$`'s. For example,

```
$(x + y)/(x - z)$
```

$(x + y)/(x - z)$

However, if you really need a blank space in your formula, you can type '`\␣`'. For example, the output of '`$2\␣a$`' is '$2\ a$', which doesn't make much sense (a little sense though).

More tips and rules about spacing is given in section 2.9.

OK, now that you know how to get '$a+b=c$', what about '$\alpha+\beta\neq\gamma$'? Well, you'll find that most symbols can be obtained simply by putting their names after '\'. For example, 'α' can be obtained by typing '`\alpha$`', 'β' by '`\beta`', etc. Others might need to be memorized, but normally they are not

that hard to remember. For example, '$\neq$' is obtained by typing '`\neq`', which is short for "**n**ot **eq**ual to." The symbol '$\in$' which means "is included **in**" can be obtained from '`\in`'.

There is an amazing document called "The Comprehensive LaTeX Symbol List," and you can download it from http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-letter.pdf. You can find virtually all the symbols you need to write anything—both good and horrifying mathematics. I assume that you have a copy of that document in hand, so this book does not offer you a list of symbols.

So far, we've been talking about *inline* equations (also called *in-text* equations). What if you want to center an equation on an individual line (the so-called *displayed* equation)? There are a few LaTeX environments that can assist you:

```
\begin{equation}
\delta\times\varepsilon=\theta
\end{equation}
\begin{equation*}
\varphi-\rho\neq\kappa
\end{equation*}
```

$$\delta \times \varepsilon = \theta \tag{2.1}$$

$$\varphi - \rho \neq \kappa$$

The `equation` environment not only centers the equation and puts it on an individual line, it also numbers the equation automatically. The `equation*` environment is a variant of `equation`. It does pretty much the same thing except that it doesn't number the equation.

In addition to `equation*`, you may also try out the `displaymath` environment. You can even type a simple `\[...\]`. Some people might tell you to use `$$...$$`. Well, don't (unless you're using plain TeX, not LaTeX)! It probably gives the same result as you want *now*, but later it might cause you much headache as it is not compatible with some LaTeX commands.

POST

'$\times$' (`\times`) or '$\cdot$' (`\cdot`)? Well, there is no quick answer to this question—most of the time, it's simply a problem of personal taste. But still, here are some general principles you might want to take into consideration:

1. Always use '$\times$' for numbers, e.g., 2×3. There is no harm in writing '$2 \cdot 3$', but '$2 \cdot 3.3$' looks rather eccentric. For consistency, I recommend using '$\times$' all the time when numbers are involved.
2. Use nothing between a number and a letter, or between a letter and a letter, e.g., $2a$, ab, but $a \times 2$.
3. Use either '$\times$' or '$\cdot$' between a number/an expression and a fraction, e.g., $2 \times \frac{1}{2}$. Reason: $2 \times \frac{1}{2} = 1$ while $2\frac{1}{2} = 2.5$. (Anyhow, I *loath* the notation $2\frac{1}{2}$.)
4. '$\times$' and $\cdot$' mean quite differently in vector operations. So treat them differently.
5. In other cases, use whatever you want. (I tend to give more preference to '$\cdot$' because it's cuter.)

Exercise

1. What's the difference between an inline equation and a displayed one? What kind of equations should be displayed?

2. Compare '-2' with '$1-2$'. Are the negative and the minus signs really the same? Does TEX have good mechanism to treat the difference?
3. How do you get '$\phi+\psi\notin\lambda$'?
4. Typeset '`\not=`', '`\not\subset`', and '`\not\sim`' in your computer. So what does the command '`\not`' do? Compare the results of '`\not\in`' and '`\notin`' *very* carefully. Are they the same?
5. Typeset '`\begin{math} a+b \end{math}`' and '`\( a+b \)`' in your computer, and see what happens.
6. What's wrong with the following sentences? How will you improve them?
   (a) For x in radians, $\frac{d}{dx}(\sin x)=\cos x$, $\frac{d}{dx}(\cos x)=-\sin x$. [Hint: A possible solution might be "For x in radians, $(\mathrm{d}/\mathrm{d}x)(\sin x)=\cos x$, $(\mathrm{d}/\mathrm{d}x)(\cos x)=-\sin x$." A better solution is to *rewrite* it: "For x in radians, $(\sin x)'=\cos x$, $(\cos x)'=-\sin x$." However, it's merely better. What do you think would be the best?]
   (b) For any constant k, $\lim\limits_{x\to c} k=k$. [Hint: Placing the limits under lim might disturb the line spacing. An alternative is "For any constant k, $\lim_{x\to c} k=k$."]
   (c) If $F(x)=\int_0^x f(t)dt$ and $G(x)=\int_0^x g(t)d(t)$, then $F(x)+G(x)=\int_0^x (f(t)+g(t))dt$. [Hint: A better solution could be "If $F(x)=\int_0^x f(t)\,\mathrm{d}t$ and $G(x)=\int_0^x g(t)\,\mathrm{d}t$, then $F(x)+G(x)=\int_0^x \big(f(t)+g(t)\big)$."]
7. Enter the following codes into your computer, and compile it.

```
\documentclass[fleqn]{article}
\begin{document}
\[ f(x)=mx+b. \]
\end{document}
```

   What does the option `fleqn` do?
   Now get a ruler ready and add '`\setlength{\mathindent}{2in}`' into the preamble. What does `\mathindent` control?
8. Now enter the following into your computer:

```
\documentclass[fleqn]{article}
\begin{document}
$$ f(x)=mx+b. $$
\end{document}
```

   Do you see why I tell you not to use `$$...$$` now?

2.2 Su$_{\text{b}}^{\text{per}}$scripts

Su$_{\text{b}}^{\text{per}}$scripts prevail in mathematics. In computers and calculators, we frequently use '`^`' to indicate a superscript, and '`_`' to indicate a subscript. The same method is adopted in LATEX:

```
$z^2$, $b_n$,\\
$x^2y^2$, $x ^2y ^2$\\
$x_12$, $x^12$
```

z^2, b_n,
x^2y^2, x^2y^2
$x_1 2$, $x^1 2$

Notice that '`^`' and '`_`' apply only to the next *single* character. What if you want more than one characters to become $\text{su}_{\text{b}}^{\text{per}}$scripted? Just enclose them in braces:

```
$x^{2y}$, $y_{3z}$\\
$x^{x^2}$, $y_{y_2}$, $y_{x^2}$
```

x^{2y}, y_{3z}
x^{x^2}, y_{y_2}, y_{x^2}

Notice that it is illegal to type '`x^y^z`' and '`x_y_z`'. Even human beings cannot tell the exact meaning of these notations—obviously, `${x^y}^z$` (x^{y^z}) and `$x^{y^z}$` (x^{y^z}) are different. You have to tell TEX which one you want. As a matter of fact, the former is quite inappropriate, if not totally wrong. You should use $(x^y)^z$, which reduces ambiguity.

Sometimes, you might need to type something like '${}_2F_3$', in which the subscript '2' follows nothing. You can just type '`$_2F_3$`'. However, the best way would be to insert an empty group: '`${}_2F_3$`'. (Do you know why?)

POST

The paragraph you are reading is rather bad—the line spacing is uneven due to the math formula $\mathrm{e}^{\frac{a^2+b^2}{c^2+d^2}}$. There are two ways to fix this problem: (1) display the formula; or (2) rewrite it as $\exp\big((a^2+b^2)/(c^2+d^2)\big)$.

Keeping the line spacing even (at least within a paragraph) is a very essential typographic principle. We'll frequently revisit this rule (and very soon).

✎ Exercise

1. How do you get $((x^2)^3)^4$ and $\big((x^2)^3\big)^4$ respectively? Which one is better? [Hint: Use "grouping."]
2. How do you get ${x+}_2F_3$? Now, you can see the value of the empty group.

But how do we get superscripts and subscripts *simultaneously*, like $\text{su}_{\text{b}}^{\text{per}}$scripts? Well, you may enter the subscript and superscript in any order you want:

```
$x^{31415}_{92}+\pi$,
$x_{92}^{31415}+\pi$\\
$F_2^2$, $F{}_2^2$
```

$x^{31415}_{92}+\pi$, $x_{92}^{31415}+\pi$
F_2^2, $F{}_2^2$

One more problem about this topic: primes. To get a prime, simply enter '`''`':

```
$y_1'+y''_2$
```

$y_1'+y''_2$

✎ Exercise

1. How do you get $x^{y_b^a}_{y_b^a}$?
2. How do you get $y_3'''+g'^2$?

3. What about $B_{n+1}^{d_{i+j}-1}$?
4. (From *The TEXbook* [9]) Mathematicians sometimes use "tensor notation" in which subscripts and superscripts are staggered, as in '$R_i{}^{jk}{}_l$'. How can do you produce it? [Hint: Use "grouping." Also, section 4.3.3 introduces a package to facilitate the typesetting of tensors.]

2.3 $\sqrt{\text{Roots}}$

Roots are produced by '`\sqrt[...]{...}`':

```
$\sqrt2$, $\sqrt2y$, $\sqrt{2y}$\\
$\sqrt[3]{2}$, $\sqrt[n+1]{x+y}$
```

$\sqrt{2}$, $\sqrt{2}y$, $\sqrt{2y}$
$\sqrt[3]{2}$, $\sqrt[n+1]{x+y}$

Some people might find the standard $\sqrt[\beta]{k}$ unacceptable. You can tune the position of the index with the amsmath package.

```
$\sqrt[\leftroot{2}\uproot{4}\beta]{k}$,
$\sqrt[\leftroot{1}\uproot{3}\beta]{k}$
```

$\sqrt[\beta]{k}$, $\sqrt[\beta]{k}$

Some obsessive ones might even find $\sqrt{x}+\sqrt{y}+\sqrt{z}$ unacceptable. (I'm *not* among them.) Two commands should be of help: (1) The command `\mathstrut` produces an invisible box whose width is zero and whose height and depth are the height and depth of a parenthesis '('. (2) The command `\smash{...}` typesets its contents but ignores both their height and depth. The amsmath package provides an optional argument, used as follows: `\smash[t]{...}` ignores the height of the box's contents, but retains the depth, while `\smash[b]{...}` ignores the depth and keeps the height. Compare:

```
$\sqrt{x}+\sqrt{y}+\sqrt{z}$\\
$\sqrt{x}+\sqrt{\mathstrut y}+\sqrt{z}$\\
$\sqrt{x}+\sqrt{\smash[b]{y}}+\sqrt{z}$
```

$\sqrt{x}+\sqrt{y}+\sqrt{z}$
$\sqrt{x}+\sqrt{\mathstrut y}+\sqrt{z}$
$\sqrt{x}+\sqrt{\smash[b]{y}}+\sqrt{z}$

POST

Like I mentioned in the previous section, we should try our best to make the line spacing even. Here's another case you have to take care of: For some special reasons, you have to put the equation

$$\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+x}}}}}?$$

in inline mode. Obviously, it's going to damage the line spacing drastically. One solution is to type $(1+(1+(1+(1+(1+x)^{1/2})^{1/2})^{1/2})^{1/2})^{1/2}$. But still, this looks rather eccentric. The best solution is still to display it, for the sake of clarity as well as aesthetic.

✎ Exercise

1. How do you get $\sqrt[n+2]{\alpha \times \beta'}$?
2. How do you get $\sqrt[k_n]{a}$? You might have to tune the position of the index k_n.
3. How do you get $\sqrt{a} + \sqrt{d} + \sqrt{y}$? [Hint: Compare your result with mine carefully!]

2.4 $\binom{\text{Fractions}}{\text{Binomials}}$—A Big Challenge!

Let's now turn to something more challenging—fractions. A fraction is obtained by typing

```
\frac{numerator}{denominator}
```

What is challenging about this? Well, if you try typing a fraction in inline mode and in display mode, you'll find that the results are different:

```
$\frac{1}{2}$,
\begin{equation*}
\frac{1}{2}
\end{equation*}
```

$\frac{1}{2}$,

$$\frac{1}{2}$$

LaTeX does this for a good reason: an inline $\frac{a+b}{c+d}$ ruins the line spacing, as you can see here; a displayed

$$\frac{a+b}{c+d}$$

is equally unacceptable. However, you *can* change LaTeX's behavior by using a few commands provided by the amsmath package: (1) `\dfrac` always typesets a fraction as if it is being typeset in the display mode; (2) `\tfrac` always typesets a fraction as if it is being typeset in the inline mode. For example:

```
This is an inline formula: $\dfrac{1}{2}$.
But avoid it! Replace it with
$\frac{1}{2}$ or $1/2$.

Instead of \begin{equation*}
\dfrac{1}{2}(a+b),
\end{equation*}
you can try \begin{equation*}
\tfrac{1}{2}(a+b).
\end{equation*}
```

This is an inline formula: $\frac{1}{2}$. But avoid it! Replace it with $\frac{1}{2}$ or $1/2$.
Instead of

$$\frac{1}{2}(a+b),$$

you can try

$$\tfrac{1}{2}(a+b).$$

Although amsmath makes it fairly easy to achieve whatever you want in your manuscript, it is easily abused. In my opinion, the command `\dfrac` should *never* be used, and I could think of only one occasion where you would need the command `\tfrac` (you will soon find it in the "post"). So over 90 percent of the time, you should be using LaTeX's `\frac` command, or you should rewrite your formula.

POST

In discussing subscripts, superscripts, and roots, we've already seen that line spacing within a paragraph should be even if at all possible. Fractions can easily destroy line spacing, if not treated carefully. But with some tricks, you can almost always fix the problem.

- There are two types of fractions: (1) slashed fractions, e.g., a/b; (2) built-up fractions, e.g.,
$$\frac{a}{b}.$$
- Inline fractions are set in the slashed form, e.g., a/b, $\pi/2$, $\sin(a/x)$. However, common numerical fractions can be set in built-up form, e.g., $\frac{1}{2}$, $\frac{2}{3}$, $\frac{1}{10}$.

  In Coleen's Workgroup, all numerical fractions are set in built-up form.
- In displayed expressions, all fractions should be built up unless they are part of a numerator or denominator or in a subscript or superscript (in subformulas):
$$\left|x^2 \sin\frac{1}{x}\right| < \frac{1}{10}, \qquad \frac{\int_0^{\pi/2} \sin^{2n} x \,\mathrm{d}x}{\int_0^{\pi/2} \sin^{2n+1} x \,\mathrm{d}x}.$$
- If there are no built-up fractions in the display, numerical fractions may be set as built-up fractions (this is the only occasion I could think of where you need `\tfrac`):
$$\tfrac{1}{2}x + \tfrac{3}{4}y + \tfrac{5}{6}z.$$
- Fractions in subformulas should always use the slash, both in inline and display:
$$x^{a/b}, \qquad y_{3/2}, \qquad \frac{(a/b)+c}{(p/q)+r}.$$
- Despite all the fine rules, clarity should always be the first priority. If a fraction, after being converted into its slashed form, is not readable, e.g., $\big(\big((a+b)/(c+d)\big)/(e+f)\big)/(g+h)$, you should display it.

The amsmath package also provides a command for typesetting continued fractions, `\cfrac`. It can also be following by an optional `[r]` or `[l]` to specify the position of the numerator:

```
\begin{equation*}
a_0+\cfrac{b_1}{
a_1+\cfrac[l]{b_2}{
a_2+\cfrac[r]{b_3}{
a_3+\cdots}}}
\end{equation*}
```

$$a_0 + \cfrac{b_1}{a_1 + \cfrac{b_2}{a_2 + \cfrac{b_3}{a_3 + \cdots}}}$$

You may also use the alternative form:

```
\[a_0+\frac{b_1}{a_1+}
      \frac{b_2}{a_2+}
      \frac{b_3}{a_3+}\cdots\]
```

$$a_0 + \frac{b_1}{a_1+}\frac{b_2}{a_2+}\frac{b_3}{a_3+}\cdots$$

Exercise

1. The following displayed equations need some fixing if they are put in inline mode. But how?

$$\frac{a+1}{b}, \quad \left(\frac{a}{x+1}\right)^3, \quad \sin\left(\frac{a}{x}\right), \quad \frac{\sin a}{x}, \quad \frac{\partial}{\partial\theta}F(u,\theta).$$

   [Hint: Converting $\frac{\sin a}{x}$ to $\sin a/x$ is mathematically correct, but is it perfect?]

2. For each of the problems below, decide which one of the listed is *perfect* to type. Then typeset them in your computer:

   (a) Inline: $a^{1/2}+b^{2/3}$, $a^{\frac{1}{2}}+b^{\frac{2}{3}}$.

   (b) Inline: $\frac{a+b}{c+d}$, $\frac{a+b}{c+d}$, $(a+b)/(c+d)$.

   (c) Display:

$$\frac{a+b}{c+d} < \tfrac{1}{10}, \quad (a+b)/(c+d) < 1/10, \quad \frac{a+b}{c+d} < \frac{1}{10}.$$

   (d) Display:

$$\frac{1}{2}a+\frac{3}{4}b, \quad \tfrac{1}{2}a+\tfrac{3}{4}b.$$

   (e) Display:

$$\frac{(a/3)+t^{1/2}}{5}, \quad \frac{\frac{a}{3}+t^{\frac{1}{2}}}{5}.$$

3. How do you get the following formula?

$$\frac{\sqrt{a+(b/c)}}{(c+d)/2}$$

4. There is actually a third kind of fractions—fractions written with exponents, e.g., $ab(cd)^{-1}$. How will you typeset $(ab)/(cd)^2$ in the alternative notation?

Binomial coefficients, like fractions, ought to be treated very carefully. The most basic command for producing a binomial coefficient is '`\binom{...}{...}`':

```
In inline mode: $\binom{k}{2}$.\\
In display mode:
\begin{equation*} \binom{k}{2}
\end{equation*}
```

In inline mode: $\binom{k}{2}$.
In display mode:

$$\binom{k}{2}$$

I recommend that you use this command *all the time*. But if you do want to do some crazy things, you can also use the commands `\dbinom` and `\tbinom` provided by the amsmath package (Think a million times before you do so!!!):

```
In inline mode: $\dbinom{k}{2}$,
  which is horrible.\\
In display mode:
\begin{equation*} \tbinom{k}{2}.
\end{equation*}
```

In inline mode: $\dbinom{k}{2}$, which is horrible.
In display mode:

$$\tbinom{k}{2}.$$

✎ Exercise

1. What do you think of the following notation? How can you improve it?

$$\binom{k}{\frac{a}{b}}.$$

It might be helpful to introduce the concept of "styles." In math mode, there are four styles:

display For normal symbols in a displayed formula.

text For normal symbols in an in-text formula.

script For subscripts and superscripts.

scriptscript For further levels of sub- and superscripting, such as subscripts of superscripts.

Take a look at the following examples:

```
Compare the small superscript in $a^{x}$
with the large one in $a^{\textstyle x}$.

Instead of using \verb"\dfrac", you can
do it this way: $\displaystyle\frac{1}{2}$.

And instead of using \verb"\tfrac", you
can try this:\begin{equation*}
\textstyle\frac{1}{2}.
\end{equation*}
```

Compare the small superscript in a^{x} with the large one in $a^{\textstyle x}$.

Instead of using `\dfrac`, you can do it this way: $\displaystyle\frac{1}{2}$.

And instead of using `\tfrac`, you can try this:

$$\textstyle\frac{1}{2}.$$

POST

In case you want to use fractions of display style in your manuscript (sigh, why???), I'd also like to introduce two more commands that would be helpful.

First take a look at the following output:

This is a test. This is a test. This is a test. This is a test. This is a test. This is a test. This is a test. This is a test. This is a test. This is a test. This is a test. $\frac{1}{2}$ This is a test. This is a test. This is a test. This is a test. This is a test. This is a test. This is a test. This is a test.

The numerator and denominator almost touch the text above and below. But after adding the following two lines:

```
\lineskiplimit=3pt
\lineskip=4pt
```

things are much better:

This is a test. This is a test. This is a test. This is a test. This is a test. This is a test. This is a test. This is a test. This is a test. This is a test. This is a test. $\frac{1}{2}$ This is a test. This is a test. This is a test. This is a test. This is a test. This is a test. This is a test. This is a test.

Here is an interpretation of the two commands from *The TEXbook* [9]:

> Whenever a box is added to a vertical list, TEX inserts "interline glue" intended to make the distance between the baseline of the new box and the baseline of the previous box exactly equal to the value of `\baselineskip`. But if the interline glue calculated by this rule would cause the top edge of the new box to be closer than `\lineskiplimit` to the bottom edge of the previous box, then `\lineskip` is used as the interline glue. In other words, the distance between adjacent baselines will be the `\baselineskip` setting, unless that would bring the boxes too close together; the `\lineskip` glue will separate adjacent boxes in the latter case.

✎ Exercise

1. Try doing more crazy things with the "style commands."

2.5 Sum and Integration

POST

The following is adapted from *Mathematics into Type* [8]:

> In text, the subscripts and superscripts [of sums and products] usually follow the symbols, while in display they are normally printed above and below:
>
> IN TEXT: $\sum_{a=1}^{\infty}$ $\prod_{a=1}^{\infty}$ IN DISPLAY: $\displaystyle\sum_{a=1}^{\infty}$ $\displaystyle\prod_{a=1}^{\infty}$
>
> In the past some publishers used to place subscripts and superscripts to the right of sums and products both in text and display to avoid extra typesetting costs. Aesthetically it is preferable to set them in display as shown above, and with modern computer typesetting systems it is no more expensive. . . .
>
> When a single integral is used, the subscripts and superscripts always follow the symbol. For multiple integrals used in display, the subscripts and superscripts may be centered above and below them.
>
> $$\int_a^b \quad \iint_A \quad \iint_A \quad \iiint_{-\infty}^{\infty} \quad \int_0^1\int_0^1$$

Good news! TEX can handle the above rules most of the time. For example,

```
Inline: $\sum_{n=1}^k$, $\prod_{n=1}^k$,
        $\int_a^b$\\
Display: \begin{equation*}
\sum_{n=1}^k,\qquad \prod_{n=1}^k,
        \qquad\int_a^b.
\end{equation*}
```

Inline: $\sum_{n=1}^{k}$, $\prod_{n=1}^{k}$, $\int_{a}^{b}$
Display:

$$\sum_{n=1}^{k}, \qquad \prod_{n=1}^{k}, \qquad \int_{a}^{b}.$$

Perfect output! However, TEX doesn't treat multiple integrals according to the rules given in the post. Here's how you can fix the problems:

```
\begin{equation*}
\iint_A,\qquad \iint\limits_A
\end{equation*}
```

$$\iint_A, \qquad \iint\limits_A$$

There are a few more commands for producing different integral signs:

```
\begin{equation*}
\iiint\limits_V,
\idotsint\limits_V, \oint_V
\end{equation*}
```

$$\iiint\limits_V, \idotsint\limits_V, \oint_V$$

There is a special symbol representing the Cauchy principal value of $\int_a^b f(x)\,\mathrm{d}x$. It is not built into LATEX, and is so far not provided by any packages available on the Internet. But here's how you can construct it:

```
\def\Xint#1{\mathchoice
{\XXint\displaystyle\textstyle{#1}}%
{\XXint\textstyle\scriptstyle{#1}}%
{\XXint\scriptstyle
      \scriptscriptstyle{#1}}%
{\XXint\scriptscriptstyle
      \scriptscriptstyle{#1}}%
\!\int}
\def\XXint#1#2#3{{
   \setbox0=\hbox{$#1{#2#3}{\int}$}
\vcenter{\hbox{$#2#3$}}\kern-.5\wd0}}
\def\dashint{\Xint-}
\[\dashint_a^bf(x)\,\rd x\]
```

$$⨍_a^b f(x)\,\mathrm{d}x$$

Sometimes, you might have to produce limits of more than one line. The amsmath package provides the command '`\substack`' which is helpful:

```
\[\sum_{\substack{0\leq i\leq m\\
                  0<j<n}} P(i,j)\]
```

$$\sum_{\substack{0\leq i\leq m\\ 0<j<n}} P(i,j)$$

You could stop reading here. But if you want to do more crazy things, continue.
You can try the "style commands" introduced in section 2.4 to change the behavior of TEX:

```
Inline: $\displaystyle\sum_{n=1}^k$,
        $\displaystyle\int_a^b$.
        Don't do these!\\
Display: \begin{equation*}
\textstyle\sum_{n=1}^k,\int_a^b.
        \end{equation*}
```

Inline: $\displaystyle\sum_{n=1}^{k}$, $\displaystyle\int_{a}^{b}$. Don't do these!
Display:

$$\textstyle\sum_{n=1}^{k}, \int_{a}^{b}.$$

The opposite of `\limits` is `\nolimits` (you need a *really* good reason to use it):

```
Inline: $\sum\limits_{n=1}^k$,
        $\int\limits_a^b$\\
Display: \begin{equation*}
\sum\nolimits_{n=1}^k,\int\limits_a^b
\end{equation*}
```

Inline: $\sum\limits_{n=1}^{k}$, $\int\limits_{a}^{b}$
Display:

$$\sum\nolimits_{n=1}^{k}, \int\limits_{a}^{b}$$

Exercise

1. Explain how you can get the followings:
   (a) $\iint_{-\infty}^{\infty} f(x)\,\mathrm{d}x$. (Use `\text{d}` to get the upright 'd'.)
   (b) $\sum_{n=1}^{k} n(n+1)$.
   (c) $\prod_{n=1}^{k} k$. ($\prod$ is produced by `\prod`.)
   (d) $\bigcap_{n=1}^{k}$. ($\bigcap$ is produced by `\bigcap`.)
2. Now typeset them in display mode.
3. Load the esint package. See what the outputs of the following are: `\int`, `\iint`, `\iiintop`, `\iiiintop`, `\dotsintop`, `\ointop`, `\oiint`, `\sqint`, `\sqiint`, `\ointctrclockwise`, `\ointclockwise`, `\varointclockwise`, `\varointctrclockwise`, `\fint`, `\varoiint`, `\laudupint`, and `\landdownint`.
4. I'd also like to introduce the `subarray` environment. See what the following codes give you.

```
\[\sum_{\begin{subarray}{l}
i\in\Lambda\\ 0\le i\le m\\
0<j<n
\end{subarray}},\quad
\sum_{\begin{subarray}{c}
i\in\Lambda\\ 0\le i\le m\\
0<j<n
\end{subarray}}\]
```

   Can you figure out what the output will be if you enter '`\begin{subarray}{r}`' instead?
5. Is the following notation good or bad? If it is bad, how can you improve it?

$$\sqrt{\sum_{n=0}^{k} n^2}$$

2.6 Functions

POST

The following is adapted from *ISO 31-11:1992* [5]:

> Functions in general [are typeset in italic type], e.g., f, g.
>
> An explicitly defined function is, however, printed in Roman (upright) type, e.g., sin, exp, ln, $\Gamma\ldots$.
>
> The argument of a function is written in parentheses after the symbol for the function, without a space between the symbol for the function and the first parenthesis, e.g., $f(x)$, $\cos(\omega x + \varphi)$. If the symbol for the function consists of two or more letters and the argument contains no operation sign, the parentheses around the argument may be omitted. In these cases, there should be a thin space between the symbol for the function and the argument, e.g., $\operatorname{ent} 2.4$, $\sin n\pi$, $\operatorname{arcosh} 2A$, $\operatorname{Ei} x$.

Again, LaTeX can handle the rules above most of the time. For example,

```
$5\sin(a+b)$, $8\cos 2A$
```

$5\sin(a+b)$, $8\cos 2A$

All the predefined functions are given in table 2.1.

Table 2.1: Predefined operators and functions

arccos	`\arccos`	arcsin	`\arcsin`	arctan	`\arctan`
arg	`\arg`	cos	`\cos`	cosh	`\cosh`
cot	`\cot`	coth	`\coth`	csc	`\csc`
deg	`\deg`	det	`\det`	dim	`\dim`
exp	`\exp`	gcd	`\gcd`	hom	`\hom`
inf	`\inf`	inj lim	`\injlim`	ker	`\ker`
lg	`\lg`	lim	`\lim`	lim inf	`\liminf`
lim sup	`\limsup`	ln	`\ln`	log	`\log`
max	`\max`	min	`\min`	Pr	`\Pr`
proj lim	`\projlim`	sec	`\sec`	sin	`\sin`
sinh	`\sinh`	sup	`\sup`	tan	`\tan`
tanh	`\tanh`	$\varinjlim$	`\varinjlim`	$\varliminf$	`\varliminf`
$\varlimsup$	`\varlimsup`	$\varprojlim$	`\varprojlim`		

Let's do some more experiments:

```
Inline: $\lim_{n\to0}((\sin^2x)/x^2)=1$\\
Display: \begin{equation*}
\lim_{n\to0}\frac{\sin^2x}{x^2}=1
\end{equation*}
```

Inline: $\lim_{n\to0}((\sin^2 x)/x^2) = 1$
Display:

$$\lim_{n\to0}\frac{\sin^2 x}{x^2} = 1$$

The behavior of the "limits" can be changed (I'm not saying that you should), in the same way we deal with $\int$ and $\sum$:

```
Inline: $\lim\limits_{n\to0}
((\sin^22x)/x^2)=1$\\
Display: \begin{equation*}
\lim\nolimits_{n\to0}\frac{\sin^2x}{x^2}=1
\end{equation*}
```

Inline: $\lim\limits_{n\to0}((\sin^2 2x)/x^2)=1$
Display:

$$\lim\nolimits_{n\to0}\frac{\sin^2 x}{x^2}=1$$

There are two more functions that are useful:

```
$1234567\bmod89=48$,\\
$y\pmod{a+b}$
```

$1234567 \bmod 89 = 48$,
$y \pmod{a+b}$

Occasionally, you'll come across functions that are not predefined, e.g., if you type `\arccot`, you'll get an error message.

The command `\DeclareMathOperator{cmd}{text}` provided by the `amsmath` package defines `cmd` to produce `text` in the appropriate font for "textual operators." If the new function being named is an operator that should, when used in displays, "take limits" (so that any subscripts and superscripts are placed above and below), then use the starred form `\DeclareMathOperator*`. For example, after defining:

```
\DeclareMathOperator{\arccot}{arccot}
\DeclareMathOperator\meas{meas}
\DeclareMathOperator*\esssup{ess\,sup}
```

you can type these commands to get amazing results:

```
\[\arccot x, \quad \meas_1,
  \quad \esssup_{x\in A}\]
```

$$\operatorname{arccot} x, \quad \operatorname{meas}_1, \quad \operatorname*{ess\,sup}_{x\in A}$$

POST

I have some comments on functions:

- 'sin' is a function with more than one letter. Therefore, it's not necessary to type '$\sin(2A)$', a simpler '$\sin 2A$' is clear enough.
- Don't type '$\sin x + y$' when you mean '$\sin(x + y)$'. Also it's better to type '$(\sin x) + y$' for clarity.
- Instead of '$(\sin x)^2$', use '$\sin^2 x$'.
- Don't use '$\sin^{-1} x$', '$\cos^{-1} x$', etc., use 'arcsin', 'arccos', etc. instead. That's because '$\sin^{-1} x$' can be interpreted as '$1/\sin x$', etc.
- Use '$\log x$' only when the base doesn't matter. Replace '$\log_{10}$' with 'lg', '$\log_e$' with 'ln', and '$\log_2$' with 'lb'.
- Instead of typing '$e^{(a+b)/\sqrt{a^2+b^2}}$', use '$\exp\big((a+b)/\sqrt{a^2+b^2}\big)$'. However, '$e^x$' doesn't need to be changed to '$\exp x$'.
- In Coleen's Workgroup, '$\operatorname{proj\,lim}$' and '$\operatorname{inj\,lim}$' are used instead of '$\varprojlim$' and '$\varinjlim$'. Similarly, we use '$\limsup$' and '$\liminf$' instead of '$\overline{\lim}$' and '$\underline{\lim}$'.

- In *ISO 31-11:1992* [5], the notation 'ent a' is used to denote the greatest integer less than or equal to a, e.g., $\text{ent}(-2.4) = -3$. In Coleen's Workgroup, we use $[\![x]\!]$ instead.

✎ Exercise

1. Explain how to get the following equation:

$$\lim_{x\to 0}\frac{\sqrt{1+x}-1}{x}=\lim_{x\to 0}\frac{(\sqrt{1+x}-1)(\sqrt{1+x}+1)}{x(\sqrt{1+x}+1)}=\lim_{x\to 0}\frac{1}{\sqrt{1+x}+1}=\frac{1}{2}. \tag{2.2}$$

2. What about this one?

$$I(z)=\sin(\frac{\pi}{2}z^2)\sum_{n=1}^{\infty}\frac{(-1)^n\pi^{2n}}{1\times 3\times\cdots\times(4n+1)}z^{4n+1}-\cos(\frac{\pi}{2}z^2)\sum_{n=1}^{\infty}\frac{(-1)^n\pi^{2n+1}}{1\times 3\times\cdots\times(4n+3)}z^{4n+3}+\frac{z^2+1}{z-1} \tag{2.3}$$

   What's bad about this output?

3. In *ISO 31-11:1992* [5], the cardinal of the set A is denoted by 'card(A)'. How can you get it?
4. What is the output of `$\mathop{\rm arccot}x`? Talk about the pros and cons of using `\mathop` instead of `\DeclareMathOperator`.

2.7 Delimiters

What's really bad about equation (2.3) in the exercise of section 2.6 is that some of the parentheses are not big enough to "enclose" things! Now, instead of typing '`(`' and '`)`', try '`\left(`' and '`\right)`'. You should get:

$$I(z)=\sin\left(\frac{\pi}{2}z^2\right)\sum_{n=1}^{\infty}\frac{(-1)^n\pi^{2n}}{1\times 3\times\cdots\times(4n+1)}z^{4n+1}-\cos\left(\frac{\pi}{2}z^2\right)\sum_{n=1}^{\infty}\frac{(-1)^n\pi^{2n+1}}{1\times 3\times\cdots\times(4n+3)}z^{4n+3}+\frac{z^2+1}{z-1},$$

which looks much better, though not best.

You can use the same mechanism with other delimiters:

```
\begin{equation*}
\left[\int+\int\right]_{x=0}^{x=1}
\end{equation*}
```

$$\left[\int+\int\right]_{x=0}^{x=1}$$

Table 2.2 gives all the delimiters that are recognized by TEX.

You can even do this:

```
\begin{equation*}
\left(\frac{a}{b}\right[
\end{equation*}
```

$$\left(\frac{a}{b}\right[$$

Table 2.2: Delimiters

Input	Delimiter	Input	Delimiter
`(`	(	`)`	)
`[` or `\lbrack`	[	`]`	]
`\{` or `\lbrace`	{	`\}` or `\rbrace`	}
`\lfloor`	⌊	`\rfloor`	⌋
`\lceil`	⌈	`\rceil`	⌉
`\langle`	⟨	`\rangle`	⟩
`/`	/	`\backslash`	\
`|` or `\vert`	|	`\\|` or `\Vert`	‖
`\uparrow`	↑	`\Uparrow`	⇑
`\downarrow`	↓	`\Downarrow`	⇓
`\updownarrow`	↕	`\Updownarrow`	⇕

If you type '`.`' after `\left` or `\right`, instead of specifying one of the basic delimiters, you get the so-called *null delimiter* (which is blank):

```
\begin{equation*}
\left(\frac{a}{b}\right.
\end{equation*}
```

$$\left(\frac{a}{b}\right.$$

This is actually very helpful, as we will see soon.

Another use of the commands `\left` and `\right`:

```
$\left|-x\right|=\left|+x\right|$.
```

$\left|-x\right|=\left|+x\right|$.

If you leave out the `\left` and `\right`, what you get is $|-x| = |+x|$. The reason is that TEX does not really understand mathematics. It thinks that you are subtracting 'x' from '|' and adding 'x' to '|', resulting in the extra spaces.

However, the mechanism of `\left` and `\right` does not always work as well as you hope!

```
\begin{equation*}
\left(a+(a+b)\right)
\end{equation*}
```

$$\left(a+(a+b)\right)$$

Well, try this:

```
\begin{equation*}
\bigl(a+(a+b)\bigr)
\end{equation*}
```

$$\bigl(a+(a+b)\bigr)$$

The `\big` delimiters are just enough bigger than ordinary ones so that the difference can be perceived, yet small enough to be used in the text of a paragraph. Here are all of them, in the ordinary size and in the `\big` size:

$$() [] \{\} \lfloor\rfloor \lceil\rceil \langle\rangle / \backslash | \| \uparrow\Uparrow\downarrow\Downarrow\updownarrow\Updownarrow$$
$$\big(\big) \big[\big] \big\{ \big\} \big\lfloor \big\rfloor \big\lceil \big\rceil \big\langle \big\rangle \big/ \big\backslash \big| \big\| \big\uparrow \big\Uparrow \big\downarrow \big\Downarrow \big\updownarrow \big\Updownarrow$$

You can also type `\Bigl` or `\Bigr` to get larger symbols suitable for displays:

$$\Big(\Big) \Big[\Big] \Big\{ \Big\} \Big\lfloor \Big\rfloor \Big\lceil \Big\rceil \Big\langle \Big\rangle \Big/ \Big\backslash \Big| \Big\| \Big\uparrow \Big\Uparrow \Big\downarrow \Big\Downarrow \Big\updownarrow \Big\Updownarrow$$

There are `\biggl` and `\biggr` versions that are 50% taller than their `\big` counterparts:

$$\biggl(\biggr)\biggl[\biggr]\biggl\{\biggr\}\biggl\lfloor\biggr\rfloor\biggl\lceil\biggr\rceil\biggl\langle\biggr\rangle\biggl/\biggr\backslash\biggl\|\biggr\|\biggl\uparrow\biggr\uparrow\biggl\downarrow\biggr\downarrow\biggl\updownarrow\biggr\updownarrow$$

Finally, there are `\Biggl` and `\Biggr` versions, 2.5 times as tall as the `\bigl` and `\bigr` delimiters:

$$\Biggl(\Biggr)\Biggl[\Biggr]\Biggl\{\Biggr\}\Biggl\lfloor\Biggr\rfloor\Biggl\lceil\Biggr\rceil\Biggl\langle\Biggr\rangle\Biggl/\Biggr\backslash\Biggl\|\Biggr\|\Biggl\uparrow\Biggr\uparrow\Biggl\downarrow\Biggr\downarrow\Biggl\updownarrow\Biggr\updownarrow$$

Any '`\...l`' delimiter is an opening, and any '`\...r`' is a closing. There are also '`\...m`' for use in the middle of formulas.

```
\begin{equation*}
\left\{\,x\in\mathbb{R}\biggm|
0<\left|x\right|<\frac{5}{3}\,\right\},
\quad\frac{a+1}{b}\bigg/\frac{c+1}{d}
\end{equation*}
```

$$\left\{\,x\in\mathbb{R}\,\middle|\,0<\left|x\right|<\frac{5}{3}\,\right\},\quad\frac{a+1}{b}\bigg/\frac{c+1}{d}$$

The '`\,`' in the first example is for the purpose of fine tuning. They add thin spaces.

Also note that in the second example, we use `\bigg`, not `\biggm`. That's because there's no need to put extra space around '/'. Compare:

```
\begin{equation}
\frac{a+1}{b}\biggm/\frac{c+1}{d}
\end{equation}
```

$$\frac{a+1}{b}\bigg/\frac{c+1}{d}\tag{2.4}$$

Sometimes `\left` and `\right` choose a larger delimiter than you want. You can tune them:

```
\begin{equation*}
\left(\sum_{k=1}^n A_k \right)\quad
\biggl(\sum_{k=1}^n A_{k}\biggr)
\end{equation*}
```

$$\left(\sum_{k=1}^{n}A_k\right)\quad\biggl(\sum_{k=1}^{n}A_k\biggr)$$

However, most of the time, `\left` and `\right` are the clear choice. The values of them are: (1) They are "automatic"—that's why they produce problems sometimes; and (2) They can produce arbitrarily large delimiters—much bigger than `\bigggggg`!

POST

You probably realized I type $\big(a+(a+b)\big)$ instead of the conventional $[a+(a+b)]$. The reason is that $[a]$ and $\{a\}$ has their unique meanings and can cause ambiguity. For example, some people might argue that $[(1+2.5)+3]=[6.3]=6$ because $[a]$ can be used to denote the "integer part" of the number a (see section 2.6). Here is an excerpt from *Mathematical Writing*:

> In some cases your audience may expect nested parentheses. In this case (or in any other case when you feel you must have them), should the outer pair be changed to brackets (or curly-braces)? This was once the prevailing convention, but it is now not only obsolete but potentially dangerous; brackets and curly braces have semantic content for many scientific professionals. ("The world is short of delimiters," says Don.) Typographers help by using slightly larger parentheses for the outer pair in a nested set.

✎ Exercise

1. Type the following excerpt from *Calculus* by Hughes-Hallett:

   Theorem If $f(x)$ is differentiable at a point $x = a$, then $f(x)$ is continuous at $x = a$.
   Proof We assume f is differentiable at $x = a$. Then we know that $f'(a)$ exists where

   $$f'(a) = \lim_{x \to a} \frac{f(x) - f(a)}{x - a}.$$

   To show that f is continuous at $x = a$, we want to show that $\lim_{x \to a} f(x) = f(a)$. We calculate $\lim_{x \to a} \big(f(x) - f(a)\big)$, hoping to get 0. By algebra, we know that for $x \neq a$,

   $$f(x) - f(a) = (x - a) \cdot \frac{f(x) - f(a)}{x - a}.$$

   Taking the limits, we have

   $$\lim_{x \to a} \big(f(x) - f(a)\big) = \lim_{x \to a} \left((x - a) \cdot \frac{f(x) - f(a)}{x - a}\right) = 0 \cdot f'(a) = 0.$$

   Thus we know that $\lim_{x \to a} f(x) = f(a)$, which means $f(x)$ is continuous at $x = a$.

2. How do you get the following?
   (a) $\big(x - s(x)\big)\big(y - s(y)\big)$
   (b) $\big||x| - |y|\big|$
3. What's the professional way to type $\big(x + f(x)\big)/\big(x - f(x)\big)$?
4. Do you still remember the notation $[\![x]\!]$? How can you produce it?
5. Some people use the "colon" notation to represent sets, e.g., $\{\, x : x > 5 \,\}$. What are the advantages and disadvantages of this notation? [Hint: What do you think of $\{\, x \in \mathbb{R} \mid |x - 3| < 3 \,\}$?]

2.8 Changing fonts

At the very beginning of this chapter, I mentioned that all variables should be set in italic type, and that all numerals should be set in an upright font. LaTeX is capable of doing this automatically. But it cannot recognize a vector automatically. So you do have to learn some font-switching commands.

Mathematical constants are usually set in an upright fonts, e.g., $\mathrm{i}^2 = -1$. Do it like this:

```
$\mathrm{i}^2=-1$
```

$\mathrm{i}^2 = -1$

Although we *write* $\vec{a}$, vectors in printed documents are set in boldface. Some people like upright bold, some prefer italic bold. Both are acceptable, and here's how to produce them:

```
$\mathbf{a}$, $\bm{a}$.
```

$\mathbf{a}$, $\boldsymbol{a}$.

(To use the command \bm, you have to load the bm package first.) You'll see the rules adopted by Coleen's Workgroup in a short while.

As you've seen in section 2.7, sets of numbers are set in a special fonts, e.g., $\mathbb{R}$ denotes the set of real numbers. Let's take a look how to produce this:

```
Load the \pcg{amsfonts} package!\\
$\mathbb{R}$: the set of real numbers.\\
$\mathbb{N}$: the set of natural
              numbers.
```

Load the amsfonts package!
$\mathbb{R}$: the set of real numbers.
$\mathbb{N}$: the set of natural numbers.

Table 2.3 gives different font switching commands.

Table 2.3: Math fonts

Command	Example
default	$ABCDEFGHIJKLMNOPQRSTUVWXYZ$
	$abcdefghijklmnopqrstuvwxyz$
\mathit	$\mathit{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
	$\mathit{abcdefghijklmnopqrstuvwxyz}$
\mathrm	$\mathrm{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
	$\mathrm{abcdefghijklmnopqrstuvwxyz}$
\mathbf	$\mathbf{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
	$\mathbf{abcdefghijklmnopqrstuvwxyz}$
\mathsf	$\mathsf{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
	$\mathsf{abcdefghijklmnopqrstuvwxyz}$
\mathtt	$\mathtt{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
	$\mathtt{abcdefghijklmnopqrstuvwxyz}$
\bm[a]	$\bm{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
	$\bm{abcdefghijklmnopqrstuvwxyz}$
\mathfrak[b]	$\mathfrak{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
	$\mathfrak{abcdefghijklmnopqrstuvwxyz}$
\mathcal	$\mathcal{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$
\mathbb[c]	$\mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$

[a] Needs package bm.
[b] Needs package amsfonts.
[c] Needs package amsfonts.

Note that the default italic and the italic produced by \mathit are different:

```
$different \neq \mathit{different}$.
```

$different \neq \mathit{different}$.

If you use the upright 'd', 'e', and 'i' a lot, you should define them in your manuscript:

```
\newcommand\rd{\mathrm{d}}
\newcommand\re{\mathrm{e}}
\newcommand\ri{\mathrm{i}}
```

Now, you can get the upright 'd', 'e', and 'i' by simply typing '\rd', '\re', and '\ri':

```
$\rd x$, $\ri^2=-1$, $\re=2.718\,28\ldots$
```

$\rd x$, $\ri^2=-1$, $\re=2.718\,28\ldots$

Now think of this problem: how do you get

$$P_{r-j} = 0 \text{ if } r - j \text{ is odd.}$$

You might be thinking of doing this, which doesn't work well:

```
\[P_{r-j}=0 \mathrm{if} r-j
        \mathrm{is odd}.\]
```

$$P_{r-j} = 0 \mathrm{if} r - j \mathrm{is odd}.$$

That's because blank spaces are ignored in math mode. Now try this:

```
\[P_{r-j}=0\ \mathrm{if}\ r-j\
        \mathrm{is\ odd}.\]
```

$$P_{r-j} = 0\ \mathrm{if}\ r - j\ \mathrm{is\ odd}.$$

However, the amsmath package provides a `\text` command which is really helpful:

```
\[P_{r-j}=0\text{ if
$r-j$ is odd.}\]
```

$$P_{r-j} = 0 \text{ if } r - j \text{ is odd.}$$

POST

Coleen's Workgroup is currently adopting the typographic rules in *ANSI/IEEE Std 260.3-1993* [4] (with some minor modifications). Here's the full text from *CWS 260.3-2005* [6]:

> Symbols for physical quantities, physical constants, mathematical variables, running numbers, general functions, and geometric elements should be printed in italic type:
>
> | | |
> |---|---|
> | A | area |
> | m | mass |
> | N_A | Avogadro constant |
> | g | acceleration due to gravity |
> | $x^2 = y^2 + z^2$ | variables x, y, and z |
> | $y = \sum_{i=1}^{m} x_i z_i$ | running numbers i |
> | $f(x)$ | f function of x |
> | $\overrightarrow{AB}$ | directed line segment from point A to point B |
>
> Symbols used for chemical elements, physical units, mathematical constants (except for ratio of perimeter to diameter of a circle), specific mathematical functions, operators, and all numerals are printed in roman (upright) type, e.g.,
>
> | | |
> |---|---|
> | O | oxygen |
> | µm | micrometer |
> | i | imaginary unit, $\mathrm{i}^2 = -1$ |
> | $\sin 2\theta$ | sine of the angle $2 \times \theta$ |
> | $a \times b + c$ | a times b plus c |
> | $\mathrm{d}x$ | differential of x |
>
> All punctuation, including grouping symbols, such as parentheses, brackets, and braces, are also printed in roman type, e.g.,
>
> | | |
> |---|---|
> | $c \times (a + b)$ | c times the sum of a and b |

$[\boldsymbol{abc}]$	triple scalar product of vectors
$\{a, b\}$	set with elements a and b
$n!$	factorial n

Subscripts and superscripts are governed by the above principles. Those that are symbols for physical quantities, mathematical variables, or for indices are printed in italic type, whereas others are printed in roman type, e.g.,

$\sin^p x$	pth power of $\sin x$
a_{ij}, a_{45}	matrix elements
I_i, I_o	input, output currents
$\mathrm{B}_x(\alpha, \beta)$	incomplete beta function
πr^2	area enclosed by a circle

To indicate the vector character of a quantity, boldface type is used, italic for general vectors, roman for unit vectors and symbols for special vector functions, e.g.,

$\boldsymbol{F}_i$	force on the ith element
$\mathbf{grad}\, f$	gradient of the scalar, f
$\mathrm{div}\, \boldsymbol{F}$	divergence of the vector, $\boldsymbol{F}$
$\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$	orthogonal unit vectors

The gradient symbol is boldface because its operation results is a vector, but "div" is not, as its operation results in a scalar. Ordinary italic type may be used to represent the magnitude of a general vector quantity.

Also noteworthy is the typeface of matrices. The symbol $\binom{a}{b}$ can be a row vector or a 2×1 matrix. Considering matrices and vectors are so closely related, denoting matrices in the same way as we denote vectors is a good idea, e.g.,

$\det \boldsymbol{A}$	determinant of matrix $\boldsymbol{A}$
$\mathbf{I}$	identity matrix: $\mathbf{I}\boldsymbol{A} = \boldsymbol{A}\mathbf{I} = \boldsymbol{A}$

You might have a question: why don't we typeset "pi" in upright type? That's because LaTeX doesn't have built-in upright lowercase Greek letters—we use the italic π because it's easier.

But it doesn't mean that you cannot get upright lowercase Greek letters. Here are a few approaches you might want to try:

- Load the package upgreek. Then you can access upright lowercase Greek letters by typing `\up...`, e.g., `\uppi`. The bad news is that the letters produced in this method don't look that good (for me).
- Load the package txfonts. Then you can access lowercase Greek letters by typing `\...up`, e.g., `\piup`. But the txfonts packages affects the typeface of the whole documents. An alternative is not to load the package and put the following codes in your preamble:

```
\DeclareSymbolFont{lettersA}{U}{txmia}{m}{it}
\DeclareMathSymbol{\piup}{\mathord}{lettersA}{25}
\DeclareMathSymbol{\muup}{\mathord}{lettersA}{22}
\DeclareMathSymbol{\deltaup}{\mathord}{lettersA}{14}
......
```

- You can also use the PostScript symbols. Try the following codes:

```
\usepackage{ifthen}
\makeatletter
\newcommand{\allmodesymb}[2]{\relax\ifmmode{\mathchoice
{\mbox{\fontsize{\tf@size}{\tf@size}#1{#2}}}
{\mbox{\fontsize{\tf@size}{\tf@size}#1{#2}}}
{\mbox{\fontsize{\sf@size}{\sf@size}#1{#2}}}
{\mbox{\fontsize{\ssf@size}{\ssf@size}#1{#2}}}}
\else
\mbox{#1{#2}}\fi}
\makeatother
\newcommand{\greeksym}[1]{\usefont{U}{psy}{m}{n}#1}
\newcommand{\ualpha}{\allmodesymb{\greeksym}{a}}
\newcommand{\udelta}{\allmodesymb{\greeksym}{d}}
\newcommand{\upi}{\allmodesymb{\greeksym}{p}}
......
```

- The lowercase Greek letters provided by GreekTEX is rather good. But there are (at least) two problems: (1) It might cause compatibility issues; (2) If you put the letters in subscripts or superscripts, their size remain the same as in text style. Anyway, here are the codes:

```
\input{greektex}
\newcommand\uppi{\text{\gr p}}
\newcommand\upd{\text{\gr d}}
......
```

POST

I have a few more words on sets:

1. We've already seen that $\mathbb{R}$ represents the set of real numbers and that $\mathbb{N}$ represents the set of natural numbers. There are more: $\mathbb{Q}$ for the set of rational numbers; $\mathbb{C}$ is the set of complex numbers; $\mathbb{Z}$ is the set of integers; $\mathbb{P}$ is the set of prime numbers.

2. Does the set of natural numbers include 0? This is a question of much controversy. Coleen's Workgroup takes the advice from *ISO 31-11:1992* [5]: Yes, it does! So $\mathbb{N} = \{0, 1, 2, \ldots, n\}$, $\mathbb{N}^* = \{1, 2, \ldots, n\}$.

3. In European countries, a subscript $+$ is frequently used to represent the set of nonnegative numbers. This is not adopted by Coleen's Workgroup. We use a superscript $+/-$ to indicate a set of positive/negative numbers. We also use a subscript 0 to indicate a set including zero. So $\mathbb{R}^+ = (0, \infty)$, $\mathbb{R}_0^+ = [0, \infty)$.

4. What's the proper notation to indicate that the set A is properly contained in B? There are two systems that are widely adopted:

	A is contained in B	A is properly contained in B
(1)	$A \subseteq B$	$A \subset B$
(2)	$A \subset B$	$A \subsetneqq B$

   Coleen's Workgroup currently use a mixture of these: we use $A \subseteq B$ to indicate A is contained in B, and $A \subsetneqq B$ to indicate A is properly contained in B.

5. Some people use $\emptyset$ (`\emptyset`) to indicate an empty set. I prefer $\varnothing$ (`\varnothing`).

6. There are many different ways of indicating the complement of B: B', $\overline{B}$, $\sim B$, B^C, etc. Coleen's Workgroup prefers the notation given in *ISO 31-11:1992* [5]: we use $\complement_A B$ to indicate the complement of B in A.

POST

If the differential operator 'd' is upright, so should the partial operator. The default partial operator produced by `\partial` is italic, like this '∂'. We can define the upright version ourselves:

```
\font\ursymbol=psyr at 10pt  % You also use other font sizes.
\def\urpartial{\mbox{\ursymbol\char"B6}}
```

Now the code '`\frac{\urpartial f}{\urpartial x}`' gives the following output:

$$\frac{\partial f}{\partial x},$$

which is perfect!

✎ Exercise

1. How will you typeset the following?

$$\binom{n}{m} = \mathrm{C}_n^m = \frac{\mathrm{P}_n^m}{m!} = \frac{n!}{(n-m)!\,m!}$$

2. What's the professional way of typesetting $\vec{a} = a_x\hat{\imath} + a_y\hat{\jmath} + a_z\hat{k}$?
3. How do you get the following?

The curl of $\boldsymbol{F}$ is represented by $\mathbf{rot}\,\boldsymbol{F}$, defined as follows:

$$\nabla \cdot \boldsymbol{F} = \left(\frac{\partial F_z}{\partial y} - \frac{\partial F_z}{\partial z}\right)\mathbf{i} + \left(\frac{\partial F_x}{\partial z} - \frac{\partial F_y}{\partial x}\right)\mathbf{j} + \left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y}\right)\mathbf{k}.$$

4. Do some research on the Internet and find out the typeface used to indicate tensors.

2.9 Spacing

We've seen the command '`\,`' which produces a thin space a couple of times. Here's another application:

```
123\,456
```

123 456

Table 2.4: Spaces in math mode

Positive Space	Example	Negative Space	Example
`$ab$`	ab		
`$a b$`	ab		
`$a\ b$`	$a\ b$		
`$a\,b$` (`a\thinspace b`)	$a\,b$	`a\!b`	$a\!b$
`$a\:b$` (`a\medspace b`)	$a\:b$	`a\negmedspace b`	$a\negmedspace b$
`$a\;b$` (`a\thickspace b`)	$a\;b$	`a\negthickspace b`	$a\negthickspace b$
`$a\quad b$`	$a\quad b$		
`$a\text{---}b$`	$a\text{---}b$		
`$a\qquad b$`	$a\qquad b$		
`$a\hspace{0.5cm}b$`	$a\hspace{0.5cm}b$	`$a\hspace{-0.5cm}b$`	$a\hspace{-0.5cm}b$
`$a\phantom{xx}b$`	ab		
`$axxb$`	$axxb$		

But this is by no means the end of the story. LaTeX provides quite a few commands for producing horizontal spaces, as are listed in table 2.4.

A few words about the command `\phantom` in the table. By using the `\phantom` command, you can reserve space for characters that do not show up in the final output:

```
\begin{equation*}
{}^{12}_{\phantom{1}6}\textrm{C}
\qquad \textrm{versus} \qquad
{}^{12}_{6}\textrm{C}
\end{equation*}
```

$$ {}^{12}_{6}\textrm{C} \qquad \textrm{versus} \qquad {}^{12}_{6}\textrm{C} $$

POST

Most of the time, TeX can producing the desired spacing. But there are a few occasions that require your attention:

- A thin space should be added before back subscripts, e.g., a_0x_1bh is obtained from `$a\,{}_0x_1bh$`.
- A thin space should be added before and after ds, dp, dx, and similar combinations of d and another symbol following, e.g.,

$$ \int f(x)\,\mathrm{d}x \qquad \iiint \mathrm{d}r\,\mathrm{d}\theta\,\mathrm{d}r $$

- A thin space should be added between a number and a unit, e.g., $1\,\mathrm{m} = 100\,\mathrm{cm}$.
- A thick space should be used before a mathematical condition in text, e.g., t_n $(n = 1, 2, \ldots)$
- An em quad should be used between a symbolic statement and a verbal expression in displayed expressions:

$$ E_n(t) \to \mathrm{e}^{-t} \quad \text{as } t \to \infty. $$

- An em quad should be used around conjunctions:

$$ x(a+b) \quad \text{or} \quad y(a-b) $$

- A two-em quad should be used between two separate formulas in the same line of a display

$$x^2 + y^2 = a, \qquad x + y = b.$$

  However, it is generally accepted that symbols in different formulas must be separated by words, e.g., instead of saying "consider S_q, $q < p$," write "consider S_q, where $q < p$."

- A two-em quad should be used between a symbolic statement and a condition on the statement.

$$x^n - y^n - z^n = A \qquad (n = 0, 1, \ldots, p)$$

The TEXbook [9] provides the following fine-tuning examples:

```
$\sqrt{2}\,x$\\
$\sqrt{\,\lg x}$\\
$O\bigl(1/\sqrt{n}\,\bigr)$\\
$[\,0,1)$\\
$\lg n\,(\lg\lg n)^2$\\
$x^2\!/2$\\
$n/\!\lg n$\\
$\Gamma_{\!2}+\Delta^{\!2}$\\
$R_i{}^j{}_{\!kl}$\\
$\int_0^x\!\int_0^y\rd F(u,v)$\\
$(2n)!/\bigl(n!\,(n+1)!\bigr)$
```

$\sqrt{2}\,x$
$\sqrt{\,\lg x}$
$O\bigl(1/\sqrt{n}\,\bigr)$
$[\,0,1)$
$\lg n\,(\lg\lg n)^2$
$x^2\!/2$
$n/\!\lg n$
$\Gamma_{\!2}+\Delta^{\!2}$
$R_i{}^j{}_{\!kl}$
$\int_0^x\!\int_0^y\rd F(u,v)$
$(2n)!/\bigl(n!\,(n+1)!\bigr)$

POST

Here are some spacing rules that TEX handles automatically, for your reference:

A **thin space** is normally used in the following cases:

— Before and after symbols used as verbs like =, e.g., $0 < y \le x$.

— Before and after symbols used as conjunctions like +, e.g., $a - b + c \times d$.

— Before but not after +, −, ±, and ∓ when used as adjectives: $-x(a + b) = -2$.

— After the commas in sets of symbols and coordinates of points, e.g., $x_1, x_2, \ldots, x_n$, $(1, 2)$.

— Before and after functions set in roman type, e.g., $a \sin x$. Exceptions: If any of these functions are preceded or followed by parentheses, braces, brackets, or bars, then the space is eliminated: $\exp(ab)$, $(a + b) \sin a$.

— Before and after vertical rules appearing singly rather than in pairs; the same rule holds for a colon that is used as a mathematical symbol rather than as punctuation: $a \mid b$, $3 : 4$.

Exercise

1. How do you get the following?

$$\sum_{p \text{ prime}} f(p) = \int_{t>1} f(t)\,\mathrm{d}\pi(t).$$

There are some more commands controlling the vertical space in and around displays.

The vertical spaces before and after each display environment are controlled by the following rubber lengths, where the values in parentheses are those for `\normalsize` with the (default) `10pt` option in the standard LaTeX classes:

- `\abovedisplayskip`, `\belowdisplayskip`
  The normal vertical space added above and below a mathematical display (default `10pt plus 2pt minus 5pt`).

- `\abovedisplayshortskip`, `\belowdisplayshortskip`
  The (usually smaller) vertical space added above and below a "short display" (`0pt plus 3pt` and `6pt plus 3pt minus 3pt`, respectively). A *short display* is one that starts to the right of where the preceding text line ends.

Here's an example demonstrating the use of these commands:

```
\small
\abovedisplayshortskip=5pt
\belowdisplayshortskip=5pt
\abovedisplayskip=15pt
\belowdisplayskip=15pt
\noindent
Before \begin{equation}
f(x) = \int\frac{\sin x}{x}dx
\end{equation}
\noindent The line doesn't
end before the formula.
\begin{equation}
f(x) = \int\frac{\sin x}{x}dx
\end{equation}
\noindent And the next line starts as
usual with some text\dots.
```

Before

$$f(x) = \int \frac{\sin x}{x} dx \tag{2.5}$$

The line doesn't end before the formula.

$$f(x) = \int \frac{\sin x}{x} dx \tag{2.6}$$

And the next line starts as usual with some text....

✎ Exercise

1. I cannot think of any good exercise right now. So carry on, or pause here....

2.10 Punctuation

In math mode, commas and semicolons are treated as punctuation marks, so TeX puts some extra spaces after them. For example,

```
$f(x,y;z)$
```

$f(x,y;z)$

This is a good mechanism, but it can cause problems. In the U.S., numbers are grouped by using commas, e.g., '123,456'. If you type `$123,456$`, what you get is '123, 456'. To get the correct spacing, type `$123{,}456$` which produces '123,456'. This approach is not adopted in Coleen's Workgroup. In Coleen's Workgroup, we group numbers with a thin space, e.g., 123 456, as you've seen before.

Interestingly, a period is not treated as a punctuation mark, so `$123.456$` does produce the correct '123.456'.

Colon is also treated as a special punctuation in TEX—representing "ratio," e.g., '3 : 4'. But it's wrong to type something like $f : A \to B$. Instead, try the following:

```
$f\mathpunct{:}A\to B$
```

$f\mathpunct{:}A\to B$

As a matter of act, if you are using standard LATEX (without loading the amsmath package), you can type `$f\colon A\to B$`. However, the amsmath package makes unfortunate major changes to the spacing produced by the command `\colon`.

Now, let's talk about something more general. When a formula is followed by a period, comma, semicolon, question mark, exclamation point, etc., put the punctuation after the `$`, when the formula is in the text; but put the punctuation before the end of a display math environment. For example,

```
If $x<0$, we have shown that
\begin{equation*} y=f(x). \end{equation*}
```

If $x<0$, we have shown that

$$y=f(x).$$

Similarly, don't ever type anything like

```
for $x=a,b$, or $c$.
```

for $x=a,b$, or c.

It should be

```
for $x=a$, $b$, or $c$.
```

for $x=a$, b, or c.

(Better yet, use a tie: '`or~$c$`'.) The reason is that TEX will typeset expression '`$x=a,b$`' as a single formula, so it will put a "thin space" between the comma and the b. This space will not be the same as the space that TEX puts after the comma following the b, since spaces between words are always bigger than thin spaces.

Another reason for not typing `$x=a,b$` is that it inhibits breaking lines in a paragraph: TEX will never break at the space between the comma and the `b` because breaks after commas in formulas are usually wrong. For example, in the equation, we certainly don't want to break something like $P(1,2)$.

✎ Exercise

1. In inline mode, which of the following is the best way to get "a_1, a_2, …, a_n"? ('…' is produced by `\dots`.)
   (a) `$a_1, a_2, \ldots, a_n$`;
   (b) `$a_1$, $a_2$, \ldots, $a_n$`;
   (c) `$a_1$, $a_2$,~\ldots, $a_n$`.

[Hint: In professional typesetting, it's a bad idea to put an ellipsis at the beginning of a line unless it does start a paragraph, which is rather rare.]

We've already seen that we can produce ellipsis "..." by typing `\ldots`. Now let's talk about it more comprehensively, starting from ellipsis in text mode. Let's start the discussion from ellipsis outside the math mode. Generally speaking, the three-or-four dot method is adopted in Coleen's Workgroup:

> Three dots indicate an omission within a quoted sentence. Four mark the omission of one or more sentences. When three are used, space occurs both before the first dot and after the final dot. When four are used, the first dot is a true period—that is, there is no space between it and the preceding word. What precedes and, normally, what follows the four dots should be grammatically complete sentences as quoted, even if part of either sentence has been omitted.

If you use the default LaTeX command, problems might arise. Try typing '`H \ldots\ H`, what you get is 'H ... H'—the spaces before and after the ellipsis are not even. LaTeX does this because we frequently put other punctuation like commas after ellipsis, and '...,' looks odd. We can fix this problem by putting '...' in math mode. For example,

```
The spirit of our American radicalism is
destructive and aimless\ldots. On the
other side, the conservative party
$\ldots$ is timid, and merely
defensive\ldots. It does not build, nor
write, nor cherish the arts, nor foster
religion, nor establish schools.
```

The spirit of our American radicalism is destructive and aimless.... On the other side, the conservative party ... is timid, and merely defensive.... It does not build, nor write, nor cherish the arts, nor foster religion, nor establish schools.

Now let's get into the math mode. Since we realize that `\ldots` does not produce a thin space after it when used in math mode, it might cause some problems. Look at the following examples:

```
$a_1,a_2,a_3,\ldots$.\\
$a_1,a_2,a_3,\ldots.$
```

$a_1, a_2, a_3, \ldots.$
$a_1, a_2, a_3, \ldots.$

The first one is logical, but the output is "awesome." The second one looks OK, but the space after the period would be distorted since it is not really considered an ending-a-sentence period. The correct way is to type this:

```
$a_1,a_2,a_3,\ldots\,$.
```

$a_1, a_2, a_3, \ldots.$

But the best way is actually '`$a_1$, $a_2$, $a_3$,~\ldots.`'.

The position of ellipsis is also an art. It is generally correct to produce formulas like

```
$x_1+\cdots+x_n$ and $(x_1,\ldots,x_n)$,
```

$x_1 + \cdots + x_n$ and $(x_1, \ldots, x_n)$,

but wrong to produce formulas like

```
$x_1+\ldots+x_n$ and $(x_1,\cdots,x_n)$.
```

$x_1 + \ldots + x_n$ and $(x_1, \cdots, x_n)$.

If you've loaded the amsmath package, try the following:

```
$x_1+\dots+x_n$ and $(x_1,\dots,x_n)$.
```

$x_1 + \cdots + x_n$ and $(x_1, \ldots, x_n)$.

The amsmath package decides the position of the ellipsis according to what kind of symbol follows `\dots`. If the next symbol is a plus sign, the dots will be centered; if it is a comma, they will be on the baseline. If the dots fall at the end of a mathematical formulas, the next object will be something like `\end` or `$`, etc., which does not give any information about how to place the dots. In such a case, you must help by using `\dotsc` for "dots with commas," `\dotsb` for "dots with binary operator/relation symbols," `\dotsm` for "multiplication dots," `\dotsi` for "dots with integrals," or even `\dotso` for "none of the above." For example,

```
A series $H_1, H_2, \dotsc\,$, a sum
$H_1+H_2+\dotsb\,$, an orthogonal product
$H_1\times H_2\times\dotsm\,$, and an
infinite integral: \[\int_{H_1}\!
   \int_{H_2}\dotsi\;
{-\Gamma}\,\rd\Theta.\]
```

A series $H_1, H_2, \ldots$, a sum $H_1+H_2+\cdots$, an orthogonal product $H_1\times H_2\times\cdots$, and an infinite integral:

$$\int_{H_1}\int_{H_2}\cdots\,-\Gamma\,\mathrm{d}\Theta.$$

I also adapt the following examples from *The TEXbook* [9] to illustrate the proper use of ellipses:

```
$x_1+\dots+x_n$\\
$x_1=\dots=x_n$\\
$A_1\times\dots\times A_n$\\
$f(x_1,\dots,x_n)$\\
$x_1x_2\dots x_n$\\
$(1-x)(1-x^2)\dots(1-x^n)$\\
$n(n-1)\dots(1)$\\
$x_1\cdot x_2\cdot\ldots\cdot x_n$
```

$x_1 + \cdots + x_n$
$x_1 = \cdots = x_n$
$A_1 \times \cdots \times A_n$
$f(x_1, \ldots, x_n)$
$x_1x_2 \ldots x_n$
$(1-x)(1-x^2)\ldots(1-x^n)$
$n(n-1)\ldots(1)$
$x_1 \cdot x_2 \cdot \ldots \cdot x_n$

Caution: the example on the last line is *not* a typo! But it does look odd, so try to avoid it.

✎ Exercise

1. How do you type the following correctly?
   Prove that $(1-x)^{-1} = 1 + x + x^2 + \cdots$. (Where do you put the period and how?)
   Clearly $a_i < b_i$ for $i = 1, 2, \ldots, n$. (Is it correct to type `$i=1,2,\dots,n$`?)
   The coefficients $c_0, c_1, \ldots, c_n$ are positive. (Is it correct to type `$c_0,c_1,\dots,c_n$`?)
2. How do you obtain $\int_0^1 \cdots \int_0^1 f(x_1, \ldots, x_n)\,\mathrm{d}x_1 \ldots \mathrm{d}x_n$?
3. How do you obtain
$$(1 + x_1z + x_1^2z^2 + \cdots)\ldots(1 + x_nz + x_n^2z^2 + \cdots) = \frac{1}{(1 - x_1z)\ldots(1 - x_nz)}?$$
4. How do you obtain
$$\prod_{j\ge0}\Bigl(\sum_{k\ge0} a_{jk}z^k\Bigr) = \sum_{n\ge0} z^n \Biggl(\sum_{\substack{k_0,k_1,\ldots\ge0\\ k_0+k_1+\cdots=n}} a_{0k_0}a_{1k_1}\ldots\Biggr)?$$

5. What about

$$\frac{(n_1+n_2+\cdots+n_m)!}{n_1!\,n_2!\ldots n_m!} = \binom{n_1+n_2}{n_2}\binom{n_1+n_2+n_3}{n_3}\cdots\binom{n_1+n_2+\cdots+n_m}{n_m}?$$

2.11 More about Displayed Equations

You might want to put two equations on two individual lines, but

```
\begin{equation*}
(a+b)^2=a^2+2ab+b^2\\
\sin^2\eta+\cos^2\eta=1
\end{equation*}
```

is not valid, as no line breaks are allowed in an `equation*` environment. What about

```
\begin{equation*}
(a+b)^2=a^2+2ab+b^2
\end{equation*}
\begin{equation*}
\sin^2\eta+\cos^2\eta=1
\end{equation*}
```

$$(a+b)^2 = a^2+2ab+b^2$$
$$\sin^2\eta+\cos^2\eta=1$$

You'll find that there's too much space between the two equations.
OK, here comes the solution: you can try the `gather` or `gather*` environment:

```
\begin{gather}
(a+b)^2=a^2+2ab+b^2\\
\sin^2\eta+\cos^2\eta=1
\end{gather}
```

$$(a+b)^2 = a^2+2ab+b^2 \tag{2.7}$$
$$\sin^2\eta+\cos^2\eta=1 \tag{2.8}$$

If you do not want the equation number, just use the starred form. What if you want to number the first equation but not the second one?

```
\begin{gather}
(a+b)^2=a^2+2ab+b^2\\
\sin^2\eta+\cos^2\eta=1\notag
\end{gather}
```

$$(a+b)^2 = a^2+2ab+b^2 \tag{2.9}$$
$$\sin^2\eta+\cos^2\eta=1$$

The `gather` environment is perfect for putting two or more equations on individual lines, centered. But sometimes, we want to "align" them at a relation symbol. We can use the `align` or `align*` environment.

```
\begin{align}
x^2+y^2 &= z^2\\
x^3+y^3 &< z^3+\cdots
\end{align}
```

$$x^2+y^2 = z^2 \tag{2.10}$$
$$x^3+y^3 < z^3+\cdots \tag{2.11}$$

Again, if you do not want the equation numbers, use the starred form.

Another challenge, what if we want to "group" these equations and "center" the equation number vertically? The answer is to use the `...ed` variant of the environments above.

```
\begin{equation}
\begin{gathered}
(a+b)^2=a^2+2ab+b^2\\
\sin^2\eta+\cos^2\eta=1
\end{gathered}
\end{equation}
```

$$\begin{gathered}(a+b)^2=a^2+2ab+b^2\\ \sin^2\eta+\cos^2\eta=1\end{gathered} \tag{2.12}$$

Another example:

```
\begin{equation*}
\begin{aligned}
x^2+y^2 &= z^2\\
x^3+y^3 &< z^3+\cdots
\end{aligned}
\end{equation*}
```

$$\begin{aligned}x^2+y^2 &= z^2\\ x^3+y^3 &< z^3+\cdots\end{aligned}$$

(What happened to the equation number?)

And another one!

```
\begin{equation}
\begin{split}
x^2+y^2 &= z^2\\
x^3+y^3 &< z^3+\cdots
\end{split}
\end{equation}
```

$$\begin{aligned}x^2+y^2 &= z^2\\ x^3+y^3 &< z^3+\cdots\end{aligned} \tag{2.13}$$

Remember how I talked about the "null delimiter"? Let's take a look at its application:

```
\begin{equation*}
\left.
\begin{aligned}
\bm{B}'&=-c\nabla\times\bm{E}\\
\bm{E}'&=c\nabla\times\bm{B}-4\pi\bm{J}
\end{aligned}\right\}
\text{Maxwell's}
\end{equation*}
```

$$\left.\begin{aligned}\boldsymbol{B}' &= -c\nabla\times\boldsymbol{E}\\ \boldsymbol{E}' &= c\nabla\times\boldsymbol{B}-4\pi\boldsymbol{J}\end{aligned}\right\}\text{Maxwell's}$$

One more command to introduce: `\intertext`.

```
\begin{align}
A_1&=N_0(\lambda;\Omega')-
     \phi(\lambda;\Omega')\\
A_2&=\phi(\lambda;\Omega')
     \phi(\lambda;\Omega)
\intertext{and finally} A_3&=
\mathcal{N}(\lambda;\omega)
\end{align}
```

$$A_1 = N_0(\lambda;\Omega') - \phi(\lambda;\Omega') \tag{2.14}$$
$$A_2 = \phi(\lambda;\Omega')\phi(\lambda;\Omega) \tag{2.15}$$

and finally

$$A_3 = \mathcal{N}(\lambda;\omega) \tag{2.16}$$

✎ Exercise

1. I can't think of any good problems for you. So just try out all the examples in this section.

2.12 Breaking an Inline Equation

When you have formulas in a paragraph, TEX may have to break them between lines. This is a necessary evil, something like the hyphenation of words; we want to avoid it unless the alternative is worse.

POST

Some really nice rules for breaking inline equations are given in *Mathematics into Type* [8]. I made some *reasonable* changes to them and here they are (Read at least the ones marked '∗'):

— Equations may be broken after an operator that is a verb (or after a comma or semicolon) that *does not* occur between fences (Why after? Because it informs the readers to go on to the next line/page.):

$$a + b = \checkmark x(c - d)$$

∗— An equation may be broken at any thick space:

$$y = 4n - 1\checkmark\ (n = 0, 1, 2)$$

— After a collective sign, no break is allowed until an operator occurs outside of fences:

$$\sum(a - b) + \checkmark abc$$

— After an integral sign, no break is allowed until a 'd' occurs; then break after the next punctuation or at a verb.

$$\int a\,\mathrm{d}x, \checkmark \int a(a + b)(xy - w)\,\mathrm{d}x = \checkmark$$

Exception: $\int \mathrm{d}x + \mathrm{d}y$, no break is allowed.

∗— When a set of fences is followed directly by another set of fences, the equation may be broken between them, provided a times sign is inserted:

$$(a + b)\checkmark(c + d)$$

(Break at ✓ and insert a times sign [some people insert a center dot, I hate it], as appropriate, before the second set of fences.)

Exceptions: (1) This rule does not apply if the fences are preceded by a sigma-class symbol. (2) Do not break at the slash (/) in a slashed fraction.

∗— Equation may be broken before or after an operator (or after a comma or semicolon) that occurs between fences if both of the following conditions are present: (1) if one of the opening fences (that has not closed) is *not* preceded directly by a noun, a fence, or sigma-class symbol; and (2) if the subsequent closing fences are *not* followed directly by a noun or a fence. It is good policy, however, to avoid breaking equations between fences whenever possible.

$$\big(a \lg x - \checkmark b \sin(x/y)\big), \qquad \big((\sin a - \cos b) + \checkmark \tan c - \checkmark(xy \cos b)\big)$$

Most of the time, TEX can handle these rules automatically. (I kind of think Donald is a big fan of *Mathematics into Type*.) A formula will be broken only *after* a relation symbol like = or < or →, or after a binary operation symbol like + or − or ×, where the relation or binary operation is on the "outer level" of the formula (i.e., not enclosed in `{...}`). For example,

```
equationbreakanequation
 $f(x,y)=x^2-y^2=(x+y)(x-y)$.
```

equationbreakanequation $f(x,y) = x^2 - y^2 = (x+y)(x-y)$.

There's a chance that TEX will break after either of the = signs (it prefers this) or after the - or + or - (in an emergency). But there won't be a break after the comma in any case.

If you don't want to permit breaking in this example except after the = signs, you could type

```
equationbreakanequation
 $f(x,y)={x^2-y^2}={(x+y)(x-y)}$.
```

equationbreakanequation $f(x,y) = x^2 - y^2 = (x+y)(x-y)$.

On the other hand, if you do want to break after the comma, try this:

```
breakanequationbreakanequation
 $f(x,\allowbreak y)=x^2-y^2=(x+y)(x-y)$.
```

breakanequationbreakanequation $f(x, y) = x^2 - y^2 = (x+y)(x-y)$.

This is not a good example. But sometimes, you might want to break something like $(x_1, \ldots, x_m, y_1, \ldots, y_n)$.

Another interesting example:

```
anequation
 $f(x,y)=x^2-y^2=(x+y)(x-y)$\\
anequation
 $f(x,y)=x^2-y^2=(x+y)\*(x-y)$\\
equation
 $f(x,y)=x^2-y^2=(x+y)\*(x-y)$
```

anequation $f(x,y) = x^2 - y^2 = (x+y)(x-y)$
anequation $f(x,y) = x^2 - y^2 = (x+y) \times (x-y)$
equation $f(x,y) = x^2 - y^2 = (x+y) \times (x-y)$

The command `\*` acts like `\-`. However, instead of inserting a hyphen, a × sign is inserted in text size.

✎ Exercise

1. Review the rules for breaking an inline equation.

2.13 Breaking a Displayed Equation

Breaking a displayed equation is mostly not preferable, but sometimes you just have to. Two things that you have to keep in heart when breaking a displayed equation: (1) It is stated in *The TEXbook* [9] that "Although formulas within a paragraph always break *after* binary operations and relations,

displayed formulas break *before* binary operations and relations." (2) Retain the logic of your math equations! I formatted the equations in this section rather carefully, in the hope that you can figure out the rules lying behind the surface.

The first environment to introduce is the `multline` environment, which does the following:

```
\begin{multline}
\text{First line of a multline}\\
\text{Centered Middle line}\\
\shoveright{\text{A right Middle}}\\
\text{Another centered Middle}\\
\text{Yet another centered Middle}\\
\shoveleft{\text{A left Middle}}\\
\text{Last line of the multline}
\end{multline}
```

$$\begin{multline}
\text{First line of a multline}\\
\text{Centered Middle line}\\
\shoveright{\text{A right Middle}}\\
\text{Another centered Middle}\\
\text{Yet another centered Middle}\\
\shoveleft{\text{A left Middle}}\\
\text{Last line of the multline}
\end{multline} \tag{2.17}$$

Let's look at a serious example:

```
\begin{multline}
A=\lim_{n\to\infty}\Delta x\Bigl(a^2+
        \bigl(a^2+2a\Delta x
         +(\Delta x)^2\bigr)\\
  +\bigl(a^2+2\times2a\Delta x+
           2^2(\Delta x)^2\bigr)\\
  +\bigl(a^2+2\times3a\Delta x+
           3^2(\Delta x)^2\bigr)\\
  +\cdots\\
  +\bigl(a^2+2\cdot(n-1)a\Delta x+
    (n-1)^2(\Delta x)^2\bigr)\Bigr)\\
  =\tfrac{1}{3}(b^3-a^3).
\end{multline}
```

$$\begin{multline}
A=\lim_{n\to\infty}\Delta x\Bigl(a^2+\bigl(a^2+2a\Delta x+(\Delta x)^2\bigr)\\
+\bigl(a^2+2\times2a\Delta x+2^2(\Delta x)^2\bigr)\\
+\bigl(a^2+2\times3a\Delta x+3^2(\Delta x)^2\bigr)\\
+\cdots\\
+\bigl(a^2+2\cdot(n-1)a\Delta x+(n-1)^2(\Delta x)^2\bigr)\Bigr)\\
=\tfrac{1}{3}(b^3-a^3).
\end{multline} \tag{2.18}$$

Now let's try the `align` environment.

```
\begin{align*}
(a+b)^3&=(a+b)(a+b)^2\\
       &=(a+b)(a^2+2ab+b^2)\\
       &=a^3+3a^2b+3ab^2+b^3
\end{align*}
```

$$\begin{aligned}
(a+b)^3&=(a+b)(a+b)^2\\
&=(a+b)(a^2+2ab+b^2)\\
&=a^3+3a^2b+3ab^2+b^3
\end{aligned}$$

You can achieve the same effect with the `split` environment.

```
\begin{equation*}
\begin{split}
(a+b)^3&=(a+b)(a+b)^2\\
       &=(a+b)(a^2+2ab+b^2)\\
       &=a^3+3a^2b+3ab^2+b^3
\end{split}
\end{equation*}
```

$$\begin{equation*}
\begin{split}
(a+b)^3&=(a+b)(a+b)^2\\
&=(a+b)(a^2+2ab+b^2)\\
&=a^3+3a^2b+3ab^2+b^3
\end{split}
\end{equation*}$$

Another example. With the following codes (Pay attention to the use of the command `\quad`):

```
\begin{verbatim}
\begin{align*}
x_nu_1+\cdots+x_{n+t-1}u_t&=
  x_nu_1+(ax_n+c)u_2+\cdots\\
  &\quad+\bigl(a^{t-1}x_n+
     c(a^{t-2}+\cdots+1)\bigr)u_t\\
  &=(u_1+au_2+\cdots+a^{t-1}u_t)x_n
     +h(u_1,\ldots,u_t).
\end{align*}
\end{verbatim}
```

we can get the following output:

$$\begin{aligned} x_nu_1+\cdots+x_{n+t-1}u_t &= x_nu_1+(ax_n+c)u_2+\cdots \\ &\quad +\bigl(a^{t-1}x_n+c(a^{t-2}+\cdots+1)\bigr)u_t \\ &= (u_1+au_2+\cdots+a^{t-1}u_t)x_n+h(u_1,\ldots,u_t). \end{aligned}$$

Now we're going to take a look at a *very* complicated example.

$$\begin{aligned} f_{h,\varepsilon}(x,y) &= \varepsilon\mathbf{E}_{x,y}\int_0^{t_\varepsilon} L_{x,y_{\varphi(\varepsilon u)}}\varphi(x)\,\mathrm{d}u \\ &= h\int L_{x,z}\varphi(x)+\rho_x(\mathrm{d}z) \\ &\quad +h\biggl(\frac{1}{t_\varepsilon}\biggl(\mathbf{E}_y\int_0^{t_\varepsilon} L_{x,y^x(s)}\varphi(x)\,\mathrm{d}s\ -t_\varepsilon\int L_{x,z}\varphi(x)\rho_x(\mathrm{d}z)\biggr) \\ &\qquad +\frac{1}{t_\varepsilon}\biggl(\mathbf{E}_y\int_0^{t_\varepsilon} L_{x,y^x(s)}\varphi(x)\,\mathrm{d}s-\mathbf{E}_{x,y}\int_0^{t_\varepsilon} L_{x,y_{\varphi(\varepsilon s)}}\varphi(x)\,\mathrm{d}s\biggr)\biggr) \end{aligned} \tag{2.19}$$

Here are the codes to typeset the equation above.

```
\begin{verbatim}
\newcommand\ve{\varepsilon}   \newcommand\tve{t_{\varepsilon}}
\newcommand\vf{\varphi}       \newcommand\yvf{y_{\varphi}}
\newcommand\bfE{\mathbf{E}}
\newcommand\relphantom[1]{\mathrel{\phantom{#1}}}
\begin{equation}
\begin{split}
f_{h,\ve}(x,y)&=\ve\bfE_{x,y}\int_0^{\tve} L_{x,\yvf(\ve u)}\vf(x)\,\rd u\\
    &=h\int L_{x,z}\vf(x)+\rho_x(\rd z)\\
    &\relphantom{=}{}+h\Biggl(\frac{1}{\tve}\biggl(
        \bfE_y \int_0^{\tve} L_{x,y^x(s)}\vf(x)\,\rd s\
            -\tve \int L_{x,z}\vf(x)\rho_x(\rd z)\biggr)\\
     &\relphantom{=}\phantom{{}+h\Biggl(}+
     \frac{1}{\tve}
     \biggl(\bfE_y\int_0^{\tve} L_{x,y^x(s)}\vf(x)\,\rd s
     -\bfE_{x,y}\int_0^{\tve} L_{x,\yvf(\ve s)}\vf(x)\,\rd s\biggr)\Biggr)
\end{split}
\end{equation}
\end{verbatim}
```

Standard LaTeX also provides the `eqnarray` environment for typesetting equations that will spread onto a few lines. I hardly use it. But it is introduced below FYI.

```
\setlength\arraycolsep{2pt}
\begin{eqnarray}
y & = & a+b+c+d\nonumber\\
  &   & +e+f+g\nonumber\\
  &   & {}+h+i+j\nonumber\\
  & \geq & {}-k-l-m
\end{eqnarray}
```

$$\begin{aligned} y &= a+b+c+d \\ &\quad +e+f+g \\ &\quad +h+i+j \\ &\geq -k-l-m \end{aligned} \tag{2.20}$$

I have two comments: (1) Notice the use of the `{}`. (Can you explain what happens right here?) (2) Setting `\arraycolsep` to 2 pt could give better output. (It controls the space before and after the sign enclosed between `&`'s.)

By the way, by default LaTeX does not allow any page break within a displayed equation. If you do want to allow page breaks, put `\allowpagebreak` in the preamble of your document.

✎ Exercise

1. Typeset the following equation:

$$\begin{aligned} \int_U \delta(I)\mu(I) \leq \sum_D \sum_{D_{I'}} \Bigg(& \int_J \alpha(J')\mu(J') - \alpha(J)\mu(J) \\ & - \int_J \frac{s(\alpha\eta)(J')}{\eta(J')} \cdot \mu(J') - \frac{s(\alpha\eta)(J)}{\eta(J)} \cdot \mu(J) \Bigg) \\ & + \left(\sum_D \sum_{D_{I'}} \left| \alpha(J) - \frac{s(\alpha\eta)(J)}{\eta(J)} \right| \cdot \mu(J) \right) \\ & \times \left(\sum_D \sum_{D_{I'}} \left| \alpha(J) - \frac{s(\alpha\eta)(J)}{\eta(J)} \right| \cdot \eta(J) \right). \end{aligned}$$

2. What is the output of the following codes? Now comes the second example:

```
\begin{eqnarray}
\lefteqn{w+x+y+z=}\\
&& a+b+c+d+e+f+\\
&& g+h+i+j+k+l
\end{eqnarray}
```

2.14 Array

You probably know that we can construct tables with the `tabular` environment. However, it cannot be used in the math mode. Arrays, in mathematics, are produced with the `array` environment. It has a single argument that specifies the number of columns and the alignment of items within the columns. For each column in the array, there is a single letter in the argument that specifies how items in the column should be positioned: `c` for centered, `l` for flush left, or `r` for flush right. Within the body of the environment, adjacent rows are separated by a `\\` command and adjacent items within a row are separated by an `&` character. For example,

```
\[
\begin{array}{clcr}
a+b+c & uv  & x-y & 27\\
a+b   & u+v & z   & 134\\
a     & 3u+uw & xyz & 2\,978
\end{array}
\]
```

$$\begin{array}{clcr} a+b+c & uv & x-y & 27 \\ a+b & u+v & z & 134 \\ a & 3u+uw & xyz & 2\,978 \end{array}$$

You can do a lot of amazing things with this structure:

```
\begin{equation*}
P_{r-j}=\left\{\!\!\!\!
\begin{array}{ll}
0 & \text{if $r-j$ is odd,}\\
r!\,(-1)^{(r-j)/2} &
   \text{if $r-j$ is even.}
\end{array}\right.
\end{equation*}
```

$$P_{r-j} = \begin{cases} 0 & \text{if } r-j \text{ is odd,} \\ r!\,(-1)^{(r-j)/2} & \text{if } r-j \text{ is even.} \end{cases}$$

The amsmath package provides an alternative:

```
\begin{equation*}
P_{r-j}=
\begin{cases}
0 & \text{if $r-j$ is odd,}\\
r!\,(-1)^{(r-j)/2} &
   \text{if $r-j$ is even.}
\end{cases}
\end{equation*}
```

$$P_{r-j} = \begin{cases} 0 & \text{if } r-j \text{ is odd,} \\ r!\,(-1)^{(r-j)/2} & \text{if } r-j \text{ is even.} \end{cases}$$

If you look closely at the two outputs, you'll find that they are actually *slightly* different. Coleen's Workgroup prefers the former one, though it is more difficult to enter.

Matrices are produced in the similar way:

```
\begin{equation*}
\left(\!\!\!\!\begin{array}{cc} 0& -1\\
1 & 0\end{array}\!\!\!\!\right)
\end{equation*}
```

$$\left(\!\!\!\!\begin{array}{cc} 0& -1\\ 1 & 0\end{array}\!\!\!\!\right)$$

Again, the amsmath package provides some simpler solutions:

```
\begin{gather*}
\begin{matrix}0&1\\ 1&0\end{matrix}\quad
\begin{pmatrix}0&1\\ 1&0\end{pmatrix}\\
\begin{bmatrix}0&1\\ 1&0\end{bmatrix}\quad
\begin{Bmatrix}0&1\\ 1&0\end{Bmatrix}\\
\begin{vmatrix}0&1\\ 1&0\end{vmatrix}\quad
\begin{Vmatrix}0&1\\ 1&0\end{Vmatrix}
\end{gather*}
```

$$\begin{matrix}0&1\\ 1&0\end{matrix}\quad \begin{pmatrix}0&1\\ 1&0\end{pmatrix}$$
$$\begin{bmatrix}0&1\\ 1&0\end{bmatrix}\quad \begin{Bmatrix}0&1\\ 1&0\end{Bmatrix}$$
$$\begin{vmatrix}0&1\\ 1&0\end{vmatrix}\quad \begin{Vmatrix}0&1\\ 1&0\end{Vmatrix}$$

It is generally speaking not preferable to put a matrix in inline mode. However, if it is a small matrix, you can do it with the `smallmatrix` environment provided by the amsmath package:

```
To show the effect of the matrix on
surrounding lines in side a paragraph,
we put it here:
$\left(\begin{smallmatrix}
1&0\\0&-1
\end{smallmatrix}\right)$ and follow
it with enough text to ensure that there
is at least one full line below the matrix.
```

To show the effect of the matrix on surrounding lines in side a paragraph, we put it here: $\left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$ and follow it with enough text to ensure that there is at least one full line below the matrix.

There's also a command provided by TEX that produces a special kind of matrix:

```
\[\bordermatrix{
  & 0 & 1 & 2\cr
0 & A & B & C\cr
1 & d & e & f\cr
2 & 1 & 2 & 3}\]
```

$$\bordermatrix{ & 0 & 1 & 2\cr 0 & A & B & C\cr 1 & d & e & f\cr 2 & 1 & 2 & 3}$$

A final trick. Some people (including me) feel the braces are too big when used with arrays. The following example might give you some insight:[1]

```
\[f(x)= \left\{%
\vphantom{\begin{array}{c} a\\[13ex]
\end{array}}\right.\kern-7pt
\begin{array}{ll}
4, & \text{if} \quad x\in (4,\infty), \\
3, & \text{if} \quad x\in (3,4], \\
2, & \text{if} \quad x \in (2,3], \\
1, & \text{if} \quad x \in (1,2], \\
0, & \text{if} \quad x \in (-\infty,1].
\end{array}\]
```

$$f(x)= \left\{% \vphantom{\begin{array}{c} a\\[13ex] \end{array}}\right.\kern-7pt \begin{array}{ll} 4, & \text{if} \quad x\in (4,\infty), \\ 3, & \text{if} \quad x\in (3,4], \\ 2, & \text{if} \quad x \in (2,3], \\ 1, & \text{if} \quad x \in (1,2], \\ 0, & \text{if} \quad x \in (-\infty,1]. \end{array}$$

Exercise

1. The following is a neatly typeset matrix example, and demonstrates the style of matrix adopted by Coleen's Workgroup:

$$\det \boldsymbol{A} = \det \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{11} \\ a_{21} & a_{22} & \cdots & a_{21} \\ a_{31} & a_{32} & \cdots & a_{31} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{m1} \end{pmatrix} = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{11} \\ a_{21} & a_{22} & \cdots & a_{21} \\ a_{31} & a_{32} & \cdots & a_{31} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{m1} \end{vmatrix}.$$

[1] Provided by Neals of the CTEX Community.

2.15 Dress Your Letters!

Sometimes, you might want to put a "hat" on your letter, e.g., $\hat{a}$. All the related commands are given in table 2.5:

Table 2.5: Accents in math mode

\acute	$\acute{a}$	\bar	$\bar{a}$	\breve	$\breve{a}$
\check	$\check{a}$	\dot	$\dot{a}$	\ddot	$\ddot{a}$
\dddot	$\dddot{a}$	\grave	$\grave{a}$	\hat	$\hat{a}$
\mathring	$\mathring{a}$	\tilde	$\tilde{a}$	\vec	$\vec{a}$
\underbar	$\underline{\text{a}}$				
\overline	$\overline{AB}$	\underline	$\underline{AB}$	\overrightarrow	$\overrightarrow{AB}$
\overleftarrow	$\overleftarrow{AB}$	\overleftrightarrow	$\overleftrightarrow{AB}$	\underleftarrow	$\underleftarrow{AB}$
\underrightarrow	$\underrightarrow{AB}$	\underleftrightarrow	$\underleftrightarrow{AB}$	\widetilde	$\widetilde{AB}$
\widehat	$\widehat{AB}$				

A few comments: (1) The notation $\vec{a}$ and $\hat{a}$ are only used in handwritten documents. In professional typesetting, use $\boldsymbol{a}$ and $\mathbf{a}$ instead. (2) If you do want to use the arrow notation, use `\imath` and `\jmath` instead of `i` and `j`, e.g., $\vec{\imath}$, $\hat{\jmath}$. (3) However, we do use $\overrightarrow{AB}$ instead of $\boldsymbol{AB}$. (4) Instead of $\widetilde{A+B}$, you might consider an alternative: $(A+B)^{\sim}$, which can be produced by typing `$(A+B)\sptilde$` provided by the amsxtra package. Here are more examples:

```
$(xyz)\spdddot$\quad$(xyz)\spddot$\\
$(xyz)\spdot$  \quad$(xyz)\spbreve$\\
$(xyz)\spcheck$\quad$(xyz)\sphat$\\
$(xyz)\sptilde$
```

$(xyz)^{\cdots}$ $(xyz)^{\cdot\cdot}$
$(xyz)^{\cdot}$ $(xyz)^{\breve{}}$
$(xyz)^{\vee}$ $(xyz)^{\hat{}}$
$(xyz)^{\sim}$

There are two more commands there are really useful: `\underbrace` and `\overbrace`.

```
\begin{align*}
y&=x^2+bx+c\\
 &=x^2+2\cdot\frac{b}{2}x+c\\
 &=\underbrace{x^2+2\cdot\frac{b}{2}x+
    \left(\frac{b}{2}\right)^2}_{\left(x+
    (b/2)\right)^2}
      -\left(\frac{b}{2}\right)^2+c
\end{align*}
```

$$\begin{aligned} y &= x^2 + bx + c \\ &= x^2 + 2 \cdot \frac{b}{2} x + c \\ &= \underbrace{x^2 + 2 \cdot \frac{b}{2} x + \left(\frac{b}{2}\right)^2}_{(x+(b/2))^2} - \left(\frac{b}{2}\right)^2 + c \end{aligned}$$

✎ Exercise

1. This is a bad exercise, but do it anyway. How do you produce $\vec{a}+\vec{b}+\vec{c}$ and $\overrightarrow{a+b+c}$ respectively? Which one is better?

CHAPTER 3

A bit further ...

3.1 Constructing New Symbols

In *ISO 31-11:1992* [5], the symbol '$a \stackrel{\text{def}}{=} b$' is used to denote "$a$ is by definition equal to b." You can easily define it with the '`\stackrel`' command provided by the amsmath package:

```
\newcommand\eqdef{%
\stackrel{\mathrm{def}}{=}}
$a\eqdef b$
```

$$a \stackrel{\text{def}}{=} b$$

Exercise

1. We can use the symbol '$p \stackrel{\wedge}{=} q$' for "p corresponds to q." How do you produce it?

3.2 Extensible arrows

The commands `\xleftarrow` and `\xrightarrow` produce horizontal relation arrows; they are intended to have textual decorations above and/or below the arrow and the length of the arrow is chosen automatically to accommodate the text. These arrows are normally available in only one size. Thus, they will probably not be suited for use in fractions, subscripts, or superscripts. For example.

```
\[
0\xleftarrow[\zeta]{} F\times
  \Delta(n-1)
   \xrightarrow{\partial_0\alpha(b)}
    E^{\partial_0 b}\]
```

$$0 \underset{\zeta}{\leftarrow} F \times \Delta(n-1) \xrightarrow{\partial_0 \alpha(b)} E^{\partial_0 b}$$

3.3 Framed Math[1]

You can use the '`\fbox`' command to get any inline equation framed. For example:

```
Let's frame \fbox{$f(x)=\sqrt{x}$}!
```

Let's frame $f(x)=\sqrt{x}$!

Now let's do a more complex example:

```
\fboxsep=1mm \fboxrule=1mm
Let's frame \fbox{$f(x)=\sqrt{x}$}!
```

Let's frame $f(x)=\sqrt{x}$!

After loading the color package, we can even frame an inline math formula in a colored box:

```
\colorbox{yellow}{ $f(x)=\sqrt{x}$}!
```

$f(x)=\sqrt{x}$!

Now let's do the same thing with displayed equation. Some good news for you: the `\fbox` command still works.

```
\fbox{\parbox{0.9\linewidth}{%
\begin{equation}
f(x)=\sqrt{x}\end{equation}}}
```

$$f(x)=\sqrt{x} \tag{3.1}$$

And `\colorbox` works as well:

```
\colorbox{yellow}{\parbox{0.9\linewidth}{%
\begin{equation}
f(x)=\sqrt{x}\end{equation}}}
```

$$f(x)=\sqrt{x} \tag{3.2}$$

If you don't want to frame the equation number, try the `\boxed` command provided by the amsmath package:

```
\begin{equation}
\boxed{W_t-F\subseteq V(P_i)\subseteq W_t}
\end{equation}
```

$$\boxed{W_t - F \subseteq V(P_i) \subseteq W_t} \tag{3.3}$$

What if you want the box to be colored as well? We can try the empheq package. It supports different frames for math environments of the amsmath package.

[1] Chapters 2 and 3 are intended to introduce standard LaTeX commands and the amsmath package only. But this section is an exception. It includes also the color and empheq packages. I think this way of organization will help you better grasp the techniques of framing a math equation.

```
\begin{empheq}[box=\fbox]{align}
f(x)=\int_1^\infty \frac{1}{x^2}\,\rd t=1
\end{empheq}
\begin{empheq}[box={\fboxsep=10pt
 \colorbox{yellow}}]{align}
f(x)=\int_1^\infty \frac{1}{x^2}\,\rd t=1
\end{empheq}
\begin{subequations}
\begin{empheq}[box={
\fboxsep=1pt\colorbox{cyan}}]{align}
f(x)&=\int_1^\infty
\frac{1}{x^2}\,\rd
   t=1\\
f(x)&=\int_2^\infty \frac{1}{x^2}\,\rd
   t=0.25
\end{empheq}
\end{subequations}
```

$$f(x) = \int_1^\infty \frac{1}{x^2}\,\mathrm{d}t = 1 \tag{3.4}$$

$$f(x) = \int_1^\infty \frac{1}{x^2}\,\mathrm{d}t = 1 \tag{3.5}$$

$$f(x) = \int_1^\infty \frac{1}{x^2}\,\mathrm{d}t = 1 \tag{3.6a}$$

$$f(x) = \int_2^\infty \frac{1}{x^2}\,\mathrm{d}t = 0.25 \tag{3.6b}$$

Exercise

1. No new exercises. Hooray! Just try out the cool examples.

3.4 Aligning Your Equations

Let's now turn to a variant of the `align` environment:

```
\begin{flalign}
x&=y & X&=Y & a&=b+c\\
x'&=y' & X'&=Y' & a'&=b
\end{flalign}
```

$$x = y \qquad X = Y \qquad a = b + c \tag{3.7}$$
$$x' = y' \qquad X' = Y' \qquad a' = b \tag{3.8}$$

Isn't that smart? You can set the space between "column-pairs" by changing `\minalignsep`, whose default value is `10pt`.

```
\renewcommand\minalignsep{25pt}
\begin{flalign}
x&=y & X&=Y & a&=b+c\\
x'&=y' & X'&=Y' & a'&=b
\end{flalign}
```

$$x = y \qquad X = Y \qquad a = b + c \tag{3.9}$$
$$x' = y' \qquad X' = Y' \qquad a' = b \tag{3.10}$$

OK, now you're ready for the following:

```
\begin{flalign}
x&=y && \text{by hypothesis}\tag{1}\\
x'&=y'&&\text{by definition}\tag{*}\\
x+x'&=y+y'&&\text{by Axiom 1}\tag{$*$}
\end{flalign}
```

$$x = y \qquad \text{by hypothesis} \quad (1)$$
$$x' = y' \qquad \text{by definition} \quad (*)$$
$$x + x' = y + y' \qquad \text{by Axiom 1} \quad (*)$$

Notice the use of `\tag`.

✎ Exercise

1. Try to do some research yourself: how do you use the `alignat` environment?

3.5 Footnotes in Math Mode

Let's do something eccentric. If you want to add a footnote to a displayed equation, you'll find that the command `\footnote` works in a very *mysterious* way. The correct solution is to use the `\footnotemark` and `\footnotetext` commands.

For example, the following code:

```
\begin{verbatim}
\begin{displaymath}
a+b=c+d\footnotemark
\end{displaymath}
\footnotetext{Here comes the footnote
 in math mode. Hooray!!!}
\end{verbatim}
```

should give the following glorious output:

$$a + b = c + d^{2}$$

3.6 Equation Numbers

The first "crazy" thing you might want to do is to put the equation number on the left side of your document. To do this is easy—you just turn "on" the `leqno` option of your document class, e.g., `\documentclass[leqno]{article}`. The premise is that the document class in question provides the `leqno` option. If not, you can achieve this by giving options to your `amsmath` package. To place equation numbers on the left, say '`\usepackage[leqno]{amsmath}`'. To place equation numbers on the right, say '`\usepackage[reqno]{amsmath}`', which is the default value.

Going on, let's talk about the style of equation numbers. This book is prepared with the standard LaTeX `book` class. Equation numbers take the form "chapter number + equation number within the chapter." You can change this by redefining the `\theequation` command. For example,

```
\renewcommand\theequation{
\thesection-\roman{equation}}
\begin{equation}
a+b=c+d
\end{equation}
```

$$a + b = c + d \tag{3.6-xi}$$

[2] Here comes the footnote in math mode. Hooray!!!

If you want to make the equation numbers to go like "chapter number + equation number within section," the amsmath package provides a useful command:

`\numberwithin{equation}{section}.`

Another topic: sub-equations. The amsmath package provides some useful commands:

```
\begin{subequations}
\begin{align}
y&=d\\
y&=cx+d\\
y&=bx^2+cx+d\\
y&=ax^3+bx^2+cx+d
\end{align}
\end{subequations}
```

$$y = d \tag{3.12a}$$
$$y = cx + d \tag{3.12b}$$
$$y = bx^2 + cx + d \tag{3.12c}$$
$$y = ax^3 + bx^2 + cx + d \tag{3.12d}$$

OK, now let's try modifying the equation numbers of the sub-equations:

```
\renewcommand\theequation{%
\theparentequation{}-\arabic{equation}}
\begin{subequations}
\begin{align}
y&=d\\
y&=cx+d\\
y&=bx^2+cx+d\\
y&=ax^3+bx^2+cx+d
\end{align}
\end{subequations}
```

$$y = d \tag{12-13a}$$
$$y = cx + d \tag{12-13b}$$
$$y = bx^2 + cx + d \tag{12-13c}$$
$$y = ax^3 + bx^2 + cx + d \tag{12-13d}$$

✎ Exercise

1. How do you get the following output?

$$f = g \tag{3.14i}$$
$$f' = g' \tag{3.14ii}$$

3.7 Prime Equation Numbers

```
First an equation.
\begin{equation}\label{e:previous}
A=B
\end{equation}
That was equation \eqref{e:previous}.

Then the same, with a prime on the number.
\begin{equation}
\tag{\ref{e:previous}$'$}
\label{e:prevprime}
C=D
\end{equation}
And that was equation \eqref{e:prevprime}.
```

First an equation.

$$A = B \tag{3.15}$$

That was equation (3.15).

Then the same, with a prime on the number.

$$C = D \tag{3.15′}$$

And that was equation (3.15′).

Notice, by the way, that when a `\ref` occurs inside a `\tag`, and that `\tag` is then `\label`'d, a `\ref` for the the second `\label` requires *three* runs of LaTeX in order to get the proper value. (If you run through the logic of LaTeX's cross-referencing mechanisms as they apply in this case, you will see that this is necessary.) Note the use of `\eqref`: instead of simply giving the "number," it also enclose the equation number in parentheses.

3.8 Equation Numbers on Both Sides

I don't know why anyone wants to do this, but here is the solution just in case.[3]

```
\makeatletter
\def\xlabel#1#2{%
{\@bsphack\protected@write\@auxout{}%
{\string\newlabel{#2}{{#1}{\thepage}}}%
\@esphack}{\mathrm(#1)}}
\makeatother

\begin{flalign}
\xlabel{H1}{eq:refL}&&x=y+z&&
                \label{eq:ee1}\\
\xlabel{H2}{eq:xxy}&&a=b^2+c^2-a&&
                \label{eq:ee2}
\end{flalign}
```

$$(H1) \qquad x = y + z \qquad (1)$$
$$(H2) \qquad a = b^2 + c^2 - a \qquad (2)$$

3.9 A List of Options of the amsmath Package

The amsmath package has the following options:

- `centertags`
  (default) Place equation numbers vertically centered on the total height of the equation when using the `split` environment.

[3] Provided by mytex of the CTeX Community.

- `tbtags`
  If the equation numbers are on the right, place equation numbers level with the last line. If the equation numbers are on the left, place equation numbers level with the first line.

- `sumlimimts`
  (default) Place the subscripts and superscripts of summation symbols above and below, in displayed equations. It also affects other symbols of the same type, e.g., $\prod$, $\bigotimes$. However, it doesn't affect integrals.

- `nosumlimits`
  Place the subscripts and superscripts of summation-type symbols to the side, even in displayed equations.

- `intlimits`
  It is just like `sumlimits`, but it works for integral symbols.

- `nointlimits`
  (default) Opposite of `intlimits`.

- `namelimits`
  (default) It is just like `sumlimits`, but it works for functions, e.g., det, lim, etc., which traditionally have subscripts placed underneath when they occur in a displayed equation.

- `nonamelimits`
  You can guess its function, can't you?

3.10 MathType into LaTeX

If you're still experiencing transition from Microsoft Word to LaTeX, MathType could be of great help for typesetting math equations.

Launch MathType, go to Preference, and click on Translators. Now select Translation to other language (text), pull down the Translator menu, and choose TeX – AMS-LaTeX. If you wish, you could also deselect Include translator name in translation.

Now enter the equation $a^2 + b^2 = c^2$ as you normally would do. Select it and click on Copy. Then go to your TeX editor, and click on Paste. You should get the following:

```
\begin{verbatim}
\[
a^2  + b^2  = c^2
\]
\end{verbatim}
```

CHAPTER 4

Further, and Further, and Further

4.1 Revisiting: Dressing Your Letters

4.1.1 More Accents: The accents Package

We've talked about accents in section 2.15. But how do we produce stuff like $\overset{*}{d}$. The accents packages can help. Here are a few examples:

```
$\accentset{*}{d}$\\
$\accentset{*}{h}$
```

$\overset{*}{d}$
$\overset{*}{h}$

If you look at the examples very carefully, you'll find that the accents package even takes the skewness of letters into consideration.

The accents package also allows you to dress your letters with "shoes":

```
$\underaccent{\bar}{x}$\\
$\underaccent{*}{x}$
```

$\underline{x}$
$\underset{*}{x}$

Refer to http://texcatalogue.sarovar.org/entries/accents.html for more details.

4.1.2 "$\imath$" in Different Fonts—The dotlessi package

I mentioned in section 2.15 that if you want to "dress" the letter "i" or "j," you should first remove the dot by using the commands `\imath` and `\jmath`. Question: how can we make the dotless i and j upright? You might want to try `$\mathrm{\imath}$`, which still gives '$\imath$'. The solution is provided by the dotlessi package. After loading the package, you can do the following:

```
$\vec{\mathsf{\dotlessj}}$\\
$\tilde{\bm{\dotlessi}}$
```

$\vec{\mathsf{\jmath}}$
$\tilde{\boldsymbol{\imath}}$

4.1.3 The undertilde Package

The undertilde package provides the `\utilde` command, which behaves more or less like TEX's `\widetilde`, except that the resulting accent is placed under the letter.

```
$\utilde{a} \neq \widetilde{a}$
```

$$\underset{\sim}{a} \neq \widetilde{a}$$

4.2 Revisiting: Case Structures—The cases Package

We already know that case structures can be constructed either with the `array` environment or the `cases` environment provided by the amsmath package. Here is another really useful package—cases. Its general syntax goes like this:

```
\begin{numcases}{left_side}
 case_1 & explanation_1 \\
 case_2 & explanation_2 \\
 ...
 case_n & explanation_n
\end{numcases}
```

Let's take a look at an example:

```
\begin{numcases}{|x|=}
  x, & for $x \geq 0$\\
  -x, & for $x < 0$
\end{numcases}
```

$$|x| = \begin{cases} x, & \text{for } x \geq 0 \quad (4.1) \\ -x, & \text{for } x < 0 \quad (4.2) \end{cases}$$

And here is a more complex one:

```
\begin{verbatim}
 \begin{subnumcases}{\label{w} w\equiv}
  0    & $c = d = 0$\label{wzero}\\
 \sqrt{|c|}\,\sqrt{\frac{1 + \sqrt{1+(d/c)^2}}{2}}   & $|c| \geq |d|$ \\
 \sqrt{|d|}\,\sqrt{\frac{|c/d| + \sqrt{1+(c/d)^2}}{2}}   & $|c| < |d|$
 \end{subnumcases}
 Then, using $w$ from equation~(\ref{w}), the square root is
 \begin{subnumcases}{\sqrt{c+\ri d}=}
 0                    & $w=0$ (case \ref{wzero})\\
 w+\ri\frac{d}{2w}      & $w \neq 0$, $c \geq 0$ \\
 \frac{|d|}{2w} + \ri w  & $w \neq 0$, $c < 0$, $d \geq 0$ \\
 \frac{|d|}{2w} - \ri w  & $w \neq 0$, $c < 0$, $d < 0$
 \end{subnumcases}
\end{verbatim}
```

The above codes generates the following output:

$$w \equiv \begin{cases} 0 & c = d = 0 \quad (4.3a) \\ \sqrt{|c|}\sqrt{\dfrac{1+\sqrt{1+(d/c)^2}}{2}} & |c| \geq |d| \quad (4.3b) \\ \sqrt{|d|}\sqrt{\dfrac{|c/d|+\sqrt{1+(c/d)^2}}{2}} & |c| < |d| \quad (4.3c) \end{cases}$$

Then, using w from equation (4.3), the square root is

$$\sqrt{c+\mathrm{i}d} = \begin{cases} 0 & w = 0 \text{ (case 4.3a)} & \text{(4.4a)} \\ w + \mathrm{i}\dfrac{d}{2w} & w \neq 0,\ c \geq 0 & \text{(4.4b)} \\ \dfrac{|d|}{2w} + \mathrm{i}w & w \neq 0,\ c < 0,\ d \geq 0 & \text{(4.4c)} \\ \dfrac{|d|}{2w} - \mathrm{i}w & w \neq 0,\ c < 0,\ d < 0 & \text{(4.4d)} \end{cases}$$

4.3 Revisiting: Vectors and Tensors

4.3.1 Arrows over Letters—The esvect Package

We've already introduced the command `\vec` to indicate a vector (though a boldface letter is preferred). The esvect package gives you more flexibility. Instead of using `\vec`, we now use the command `\vv` to dress the letter:

```
$\vv{E}$, $\vv{\imath}$, $\vv{AB}$
```

$\vec{E}$, $\vec{\imath}$, $\overrightarrow{AB}$

Notice that the arrow produced by `\vv` is actually extendable.

There are also starred version of the command, for properly placing subscripts following a vector. Compare the following:

```
$\vec{e}_r$, $\vv{e}_r$, $\vv*{e}{r}$
```

$\vec{e}_r$, $\vec{e}_r$, $\vec{e}_r$

The choice of arrow is done by specifying an option for the package, i.e., `\usepackage[...]{esvect}`. The '`...`' can be letters `a` to `h`. The corresponding shapes are shown below:

option	a	b	c	d	e	f	g	h
flèche	→	→	→	→	→	→	→	→

4.3.2 The vector Package

The vector package provides some commands to ease the typesetting of vectors. Keep in mind that the designer of the package might not necessarily abide by the rules I proposed. Anyway, let me demonstrate some useful commands:

```
\begin{gather*}
\uvec{p}=\left(\irvec{q}\right)\\
\uuvec{q}=\left(\!\!\icvec{q}\!\!\right)\\
\buvec{r}=\{\rvec{r}{1}{6}\}\\
\bvec{s}=\left[
   \!\!\cvec{s}{0}{2}\!\!\right]
\end{gather*}
```

$$\underline{p} = (q_1, \ldots, q_n)$$
$$\hat{\underline{q}} = \begin{pmatrix} q_1 \\ \vdots \\ q_n \end{pmatrix}$$
$$\hat{\mathbf{r}} = \{r_1, r_2, r_3, r_4, r_5, r_6\}$$
$$\mathbf{s} = \begin{bmatrix} s_0 \\ s_1 \\ s_2 \end{bmatrix}$$

4.3.3 The tensor Package

We've already known that $R_i{}^j{}_{kl}$ can be obtained from `$R_i{}^j{}_{\!kl}$`. This is rather hard to enter! The tensor package provides an easier solution. Here's how:

```
$R\indices{_i^j_{\!kl}}$
```

$R_i{}^j{}_{kl}$

You can even do some very complex things with this package:

```
$\tensor[^a_b^c_d]{M}{^a_b^c_d}$
```

${}^{a\ c}_{b\ d}M^{a\ c}_{b\ d}$

The two commands mentioned above also have "starred" forms, which can collapse the spacing. This can be quite useful! For example,

```
$\tensor*[^{14}_6]{\text{C}}{}$
```

${}^{14}_{6}\mathrm{C}$

Exercise

1. How do you get the following: ${}^{*}_{*}\prod^{*}_{*}$?
2. You might want to try out the tensind and mattens packages. Try the website http://www.ctan.org for help!

4.4 Revisiting: Extensible Arrows

4.4.1 The extarrows Package

The amsmath package provides a few simple extendable arrows. The extarrows package is pretty much a supplement. It follows the same syntax of amsmath:

```
\arrowname[subscript]{superscript}
```

Examples are listed in table 4.1.

4.4.2 The harpoon Package

Let's talk a bit more about the notation for directed line segments. Some people do not like the notation $\overrightarrow{AB}$ and want a change. The harpoon package is a good choice.

```
$\overrightharp{AB}$, $\overleftharp{AB}$,
$\overrightharpdown{AB}$,
$\overleftharpdown{AB}$,
$\underrightharp{AB}$,
$\underleftharp{AB}$,
$\underrightharpdown{AB}$,
$\underleftharpdown{AB}$
```

$\overrightharp{AB}$, $\overleftharp{AB}$, $\overrightharpdown{AB}$, $\overleftharpdown{AB}$, $\underrightharp{AB}$, $\underleftharp{AB}$, $\underrightharpdown{AB}$, $\underleftharpdown{AB}$

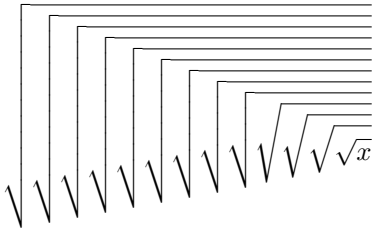
`\xlongequal:`	$A \xlongequal[sub]{\text{we love to love}} Z$
`\xLongleftarrow:`	$A \xLongleftarrow[sub]{\text{we love to love}} Z$
`\xLongrightarrow:`	$A \xLongrightarrow[sub]{\text{we love to love}} Z$
`\xLongleftrightarrow:`	$A \xLongleftrightarrow[sub]{\text{we love to love}} Z$
`\xLeftrightarrow:`	$A \xLeftrightarrow[sub]{\text{we love to love}} Z$
`\xlongleftrightarrow:`	$A \xlongleftrightarrow[sub]{\text{we love to love}} Z$
`\xlongrightarrow:`	$A \xlongrightarrow[sub]{\text{we love to love}} Z$
`\xleftrightarrow:`	$A \xleftrightarrow[sub]{\text{we love to love}} Z$
`\xlongleftarrow:`	$A \xlongleftarrow[sub]{\text{we love to love}} Z$
(*amsmath*) `\xleftarrow:`	$A \xleftarrow[sub]{\text{we love to love}} Z$
(*amsmath*) `\xrightarrow:`	$A \xrightarrow[sub]{\text{we love to love}} Z$
`\xlongleftarrow:`	$A \longleftarrow Z$
`\xlongrightarrow:`	$A \longrightarrow Z$
(*amsmath*) `\xleftarrow:`	$A \leftarrow Z$
(*amsmath*) `\xrightarrow:`	$A \rightarrow Z$

Table 4.1: Extendible arrows of the extarrows package

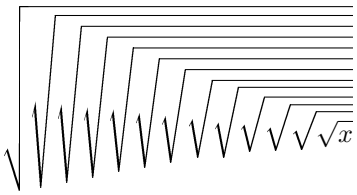
4.5 Revisiting: Delimiters

4.5.1 Largggggge Delimiters—The yhmath Package

An old saying goes, "Even TEX becomes dumb sometimes." (Well, maybe not that old.) And rightly so! Here is the default output of a series of root signs:



What's the word that comes up to your mind when you see the output? "Ugly," I suppose. This is what happens here: Only a few root signs were defined in TEX. When they are used out, TEX will "construct" new root signs—that's how the vertical ones come into being. However, if you load the yhmath package, the output would be very different and better:



An important feature of the yhmath package is that it provides a set of large delimiters. That is to say, virtually all large delimiters will be different from the original output of TEX. (I hardly ever use this package because although it does provide really neat root signs, the parentheses are way beyond my sense of aesthetic.)

Anyway, here are a few other features of the package. It also offers some wide accents. You might remember that there is a limit to TEX's commands such as `\widetilde`, e.g., `\widetilde{ABCDEFG}` would become $\widetilde{ABCDEFG}$, which is awful. But after loading the yhmath package, the output becomes:

$$\widetilde{ABCDEFG}$$

Yet, I still insist that $(ABCDEFG)^{\sim}$ is a better solution. Hopefully you would agree with me.

The yhmath package also provides the `amatrix` environment which is used the same as amsmath's `pmatrix`, but instead of parenthesis, angles are used. For example, you can easily construct the following:

$$\left\langle \begin{matrix} a_1 & a_2 \\ a_3 & a_4 \end{matrix} \right\rangle$$

I listed here some important features of yhmath which I think are most likely to be used. But it has other functions. Please refer to http://texcatalogue.sarovar.org/entries/yhmath.html.

4.5.2 The delarray Package

The delarray package is a useful general extension to the array package that allows you to specify opening and closing extensible delimiters to surround a mathematical `array` environment.

```
\[\bm{Q}=
\begin{array}[t]({cc}) X&Y \end{array}
\begin{array}[t][{cc}]A&B\\ C&D\end{array}
\begin{array}[b]\lgroup{cc}\rgroup
 L\\M\end{array}\]
```

$$\boldsymbol{Q} = \begin{pmatrix} X & Y \end{pmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{pmatrix} L \\ M \end{pmatrix}$$

4.6 Revisiting: Matrix—The pmat Package

The pmat package is designed for typesetting partitioned matrices. The `\pmat` macro takes three arguments. The first one is a left delimiter (the thing you put immediately after a `\left` command). The last one is a right delimiter (the thing you put immediately after a `\right` command). As usual, a delimiter may be omitted by using a dot (`.`). The middle argument specifies the dashed vertical lines that are to be placed between columns of the matrix. This argument must contain exactly $n-1$ characters, where n is the number of columns of the matrix. If a character is a `|` then a dashed vertical line will be placed between the appropriate columns. Otherwise, no dashed line will be placed between those columns (we recommend the use of the character dot (.) in these cases). The format of the entries of the partitioned matrix follows the conventions of plain TEX, i.e., entries are separated by a `&` (just like in LATEX), but lines are separated by a `\cr` (instead of the `\\` used in LATEX). All entries are typeset in math mode (in `\textstyle`). For technical reasons, a `\cr` must also be placed at the

end of the last line. The placement of horizontal dashed lines is done with the command `\-`, which must be placed immediately after the command `\cr`. For example,

```
\[
\begin{pmat}({.|})
a_{11} & a_{12} & b_{11} \cr
a_{21} & a_{22} & b_{21} \cr\-
c_{11} & c_{12} & d_{11} \cr
\end{pmat}
\]
```

$$\left(\begin{array}{cc:c} a_{11} & a_{12} & b_{11} \\ a_{21} & a_{22} & b_{21} \\ \hdashline c_{11} & c_{12} & d_{11} \end{array}\right)$$

A lot of parameters may be changed to modify the appearance. For more information, refer to ftp://ibiblio.org/pub/packages/TeX/macros/generic/pmat/pmat.pdf.

4.7 Revisiting: Equation Numbers—The subeqnarray Package

The subeqnarray package defines the `subeqnarray` and `subeqnarray*` environments, which behave like the equivalent `eqnarray` and `eqnarray*` environments, except that the individual lines are numbered like 1a, 1b, 1c, etc. Here's an application:

```
\begin{subeqnarray}
\label{eqw} \slabel{eq0}
 x & = & a \times b \\
\slabel{eq1}
 & = & z + t\\
\slabel{eq2}
 & = & z + t
\end{subeqnarray}
The first equation is number~\eqref{eq0},
the last is~\eqref{eq2}. The
equation as a whole can be referred to as
equation~\eqref{eqw}.
```

$$\begin{aligned} x &= a \times b && \text{(4.5a)} \\ &= z + t && \text{(4.5b)} \\ &= z + t && \text{(4.5c)} \end{aligned}$$

The first equation is number (4.5a), the last is (4.5c). The equation as a whole can be referred to as equation (4.5).

Exercise

1. There is a more powerful package called subeqn. Study it!

4.8 Revisiting: Sets—The braket Package

The braket package can greatly reduce your effort in producing sets:

```
\begin{equation*}
\Set{x\in\mathbb{R}|
  0<\left|x\right|<\frac{5}{3}}\\
\end{equation*}
```

$$\left\{ x \in \mathbb{R} \,\middle|\, 0 < |x| < \frac{5}{3} \right\}$$

4.9 Commutative Diagrams—The amscd Package

Some commands for producing simple commutative diagrams based on arrays are available in the amscd package. It provides some useful shorthand forms for specifying the decorated arrows and other connectors.

In the CD environment the notations @>>>, @<<<, @VVV, and @AAA give right, left, down, and up arrows, respectively. For example,

```
\[\begin{CD}
\cov(L) @>>> \non(K) @>>> \cf(K)\\
@VVV @AAA @AAA \\
\add(L) @>>> \add(K) @>>> \cov(K)\\
\end{CD}\]
```

$$\begin{array}{ccccc} \mathrm{cov}(L) & \longrightarrow & \mathrm{non}(K) & \longrightarrow & \mathrm{cf}(K) \\ \downarrow & & \uparrow & & \uparrow \\ \mathrm{add}(L) & \longrightarrow & \mathrm{add}(K) & \longrightarrow & \mathrm{cov}(K) \end{array}$$

Decorations on the arrows are specified as follows. For the horizontal arrows, material between the first and second > or < symbols will be typeset as a superscript, and material between the second and third will be typeset as a subscript. Similarly, material between the first and second, or second and third, As or Vs of vertical arrows will be typeset as left or right "side-scripts".

The notations @= and @| give horizontal and vertical double lines.

A "null arrow" (produced by @) can be used instead of a visible arrow to fill out an array where needed.

```
\[\begin{CD}
S^{W_\Lambda}\otimes T @>j>>
    T \\
@VVV @VV{\mathop{\rm End} P}V\\
(S\otimes T)/I @= (Z\otimes T)/J
\end{CD}\]
```

$$\begin{array}{ccc} S^{W_\Lambda}\otimes T & \xrightarrow{\ j\ } & T \\ \downarrow & & \Big\downarrow{\scriptstyle \mathrm{End}\,P} \\ (S\otimes T)/I & = & (Z\otimes T)/J \end{array}$$

4.10 Coloring Your Math—The color Package

There is no difference in producing colored text and colored math expression. With the color package, you can do this:

```
\begin{equation}
\textcolor{blue}{f(x)}=\int_1^\infty
  \textcolor{red}{\frac{1}{x^2}}\,\rd x=1
\end{equation}
```

$$\textcolor{blue}{f(x)} = \int_1^\infty \textcolor{red}{\frac{1}{x^2}}\,\mathrm{d}x = 1 \tag{4.6}$$

4.11 Packages Smarter Than Me

4.11.1 The polynom package

Here comes a package that is better at math than I am. An example should shed some light on its usage:

```
\polylongdiv{x^3+x^2-1}{x-1}
```

$$
\begin{array}{rl}
 & x^2+2x+2 \\
\cline{2-2}
x-1 \big) & x^3+x^2-1 \\
 & \underline{-x^3+x^2} \\
 & \,2x^2 \\
 & \underline{-2x^2+2x} \\
 & \,2x-1 \\
 & \underline{-2x+2} \\
 & \,1
\end{array}
$$

Here are more examples:

```
\polyhornerscheme[x=1]{x^3+x^2-1}
```

$$
\begin{array}{r|cccc}
 & 1 & 1 & 0 & -1 \\
1 & & 1 & 2 & 2 \\
\cline{2-5}
 & 1 & 2 & 2 & 1
\end{array}
$$

```
\polyfactorize{2x^3+x^2-7x+3}
```

$$2\left(x-\tfrac{1}{2}\right)\left(x+\tfrac{1}{2}+\tfrac{\sqrt{13}}{2}\right)\left(x+\tfrac{1}{2}-\tfrac{\sqrt{13}}{2}\right)$$

Also, the output of `\polylonggcd{(x-1)(x-1)(x^2+1)}{(x-1)(x+1)(x+1)}` is:

$$
\begin{aligned}
x^4-2x^3+2x^2-2x+1 &= \left(x^3+x^2-x-1\right)\cdot\left(x-3\right)+\left(6x^2-4x-2\right)\\
x^3+x^2-x-1 &= \left(6x^2-4x-2\right)\cdot\left(\tfrac{1}{6}x+\tfrac{5}{18}\right)+\left(\tfrac{4}{9}x-\tfrac{4}{9}\right)\\
6x^2-4x-2 &= \left(\tfrac{4}{9}x-\tfrac{4}{9}\right)\ \cdot\ \left(\tfrac{27}{2}x+\tfrac{9}{2}\right)\ +0
\end{aligned}
$$

4.11.2 The longdiv package

The longdiv is actually a TEX package. So you should load it with '`\input longdiv.tex`'. Now take a look at what you can do with it:

```
\longdiv{31415}{2}
```

$$
\begin{array}{r}
15707 \\
\cline{1-1}
2\,\big)\,31415 \\
\underline{20000} \\
11415 \\
\underline{10000} \\
1415 \\
\underline{1400} \\
15 \\
\underline{14} \\
1
\end{array}
$$

4.12 The mathlig Package

The mathlig is a TEX package and should be loaded by '`\input mathlig.tex`'. It can produce special "math ligatures," like these:

```
\mathlig{->}{\rightarrow}
\mathlig{<-}{\leftarrow}
\mathlig{<->}{\leftrightarrow}
$->$, $<-$, $<->$
```

→, ←, ↔

4.13 Miscellaneous

4.13.1 Canceling out—The cancel Package

Another short section. (Happy?) After loading the cancel package, you can do this:

```
\[f(x)=\frac{(x^2+1)\cancel{(x-1)}}
{\cancel{(x-1)}(x+1)}\]
```

$$f(x) = \frac{(x^2+1)\cancel{(x-1)}}{\cancel{(x-1)}(x+1)}$$

4.13.2 The units and nicefrac Packages

About the loading of the packages: (1) When you load the units package, the nicefrac package is loaded automatically; (2) The units package itself has two options, `tight` and `loose`. The default value, `tight`, indicates a thin space will be added between the number and the unit. The option `loose` will add a normal word spacing between the number and the unit. You should remember that I have said that a thin space is preferable! (3) The nicefrac package has two options, `nice` and `ugly`. We'll talk a little bit about them in a short while. (4) Options specified for the units package will be passed on to the nicefrac package. (5) The nicefrac package can be used independently.

Let's now take a look how this package can be used. Suppose no options are specified; i.e., the options `tight` and `nice` are used, this is what you are going to get:

```
\unit[20]{cm}\\
\unitfrac[20]{m}{s}\\
\nicefrac[\textsf]{m}{s}
```

20 cm
20 m/s
m/s

However, if the `\ugly` option is specified, the command `\unitfrac[20]{m}{s}` will produce 20 m/s.

My recommendation is to use `\usepackage[ugly]{units}` which would produce the output I've been proposing in this book.

4.13.3 Math in Titles—The maybemath Package

The maybemath package provides a set of commands for adjusting math mode typesetting to match the context of the surrounding paragraph.

For context-sensitive boldness use `\maybebm`:

```
Normal $x^2+\maybebm{x^3}+\cdots$\\
\textbf{$x^2+\maybebm{x^3}+\cdots$}
```

Normal $x^2 + x^3 + \cdots$
$x^2 + \boldsymbol{x^3} + \cdots$

For context-sensitive upright math typesetting use `\mayberm`:

```
Normal $x^2+\mayberm{x^3}+\cdots$\\
\textit{$x^2+\mayberm{x^3}+\cdots$}
```

Normal $x^2 + \mathrm{x}^3 + \cdots$
$x^2 + x^3 + \cdots$

Alternatively, to force `\mathit` in italic contexts use `\maybeit`:

```
Normal $x^2+\maybeit{x^3}+\cdots$\\
\textit{$x^2+\maybeit{x^3}+\cdots$}
```

Normal $x^2 + x^3 + \cdots$
$x^2 + x^3 + \cdots$

The functionality of both `\mayberm` and `\maybeit` is combined for convenience in the command `\maybeitrm`:

```
Normal $x^2+\maybeitrm{x^3}+\cdots$\\
\textit{$x^2+\maybeitrm{x^3}+\cdots$}
```

Normal $x^2 + \mathrm{x}^3 + \cdots$
$x^2 + x^3 + \cdots$

For context-sensitive sans-serif math typesetting use `\maybesf`:

```
Normal $x^2+\maybesf{x^3}+\cdots$\\
\textit{$x^2+\maybesf{x^3}+\cdots$}
```

Normal $x^2 + x^3 + \cdots$
$x^2 + x^3 + \cdots$

For combined bold-and-sans-serif context handling, a `\maybebmsf` command is provided:

```
Normal $x^2+\maybebmsf{x^3}+\cdots$\\
\textbf{$x^2+\maybebmsf{x^3}+\cdots$}\\
\textsf{$x^2+\maybebmsf{x^3}+\cdots$}\\
\textbf{\textsf{$x^2+\maybebmsf{x^3}+
    \cdots$}}
```

Normal $x^2 + x^3 + \cdots$
$x^2 + \boldsymbol{x^3} + \cdots$
$x^2 + \mathsf{x}^3 + \cdots$
$x^2 + \mathbf{\mathsf{x}^3} + \cdots$

The most important application of this package is to control the font in titles. If you are using the default book or article class files, type things like '`\section{... $\maybebm{...}$}`' to get the correct font.

4.13.4 The nccmath Package

The nccmath package extends the amsmath package. It also improves spacing control before display equations and fixes a bug of ignoring the `\displaybreak` in the amsmath version of the `equation` environment.

Its first feature is a modification to the `\intertext` command:

```
\begin{align*}
a+b&=c+d.
\intertext[1cm]{Therefore,}
e+f&=g+h.
\end{align*}
```

$$a + b = c + d.$$

Therefore,

$$e + f = g + h.$$

As you can see, the additional option can specify a vertical space inserted before and after the text. If it is omitted, standard TEX's skips are inserted.

It also allows you to create a series of medium-sized mathematics:

```
\[\medmath{\cfrac{1}{\sqrt 2 +
      \cfrac{1}{\sqrt 2 +\dotsb}}}
\quad \cfrac{1}{\sqrt 2 +\cfrac{1}
       {\sqrt 2 +\dotsb}}\]
```

$$\medmath{\cfrac{1}{\sqrt 2 + \cfrac{1}{\sqrt 2 +\dotsb}}} \quad \cfrac{1}{\sqrt 2 + \cfrac{1}{\sqrt 2 +\dotsb}}$$

Now medium-sized operators:

```
$\sum_{i=1}^n \medop\sum_{i=1}^n
\displaystyle\sum\nolimits_{i=1}^n$
\quad $\sum\limits_{i=1}^n
\displaystyle \medop\sum_{i=1}^n
\sum_{i=1}^n$
```

$\sum_{i=1}^n \medop\sum_{i=1}^n \displaystyle\sum\nolimits_{i=1}^n$ \quad $\sum\limits_{i=1}^n \displaystyle \medop\sum_{i=1}^n \sum_{i=1}^n$

There are also commands for producing medium-sized integral, fractions, binomial coefficient, and matrix.

```
$\int_a^b\medint\int_a^b
 \displaystyle\int_a^b$\quad
$\iint_a^b\medint\iint_a^b$
```

$\int_a^b\medint\int_a^b \displaystyle\int_a^b$\quad $\iint_a^b\medint\iint_a^b$

```
$\frac{x+y}{a-b} \mfrac{x+y}{a-b}
  \dfrac{x+y}{a-b}$\quad
$\binom{n}{k} \mbinom{n}{k}
  \dbinom{n}{k}$
```

$\frac{x+y}{a-b} \mfrac{x+y}{a-b} \dfrac{x+y}{a-b}$\quad $\binom{n}{k} \mbinom{n}{k} \dbinom{n}{k}$

```
$\bigl(\begin{smallmatrix}  a&b\\c&d
\end{smallmatrix}\bigr)$
$\Bigl(\begin{mmatrix}  a&b\\c&d
\end{mmatrix}\Bigr)$
$\begin{pmatrix}  a&b\\c&d
\end{pmatrix}$
```

$\bigl(\begin{smallmatrix} a&b\\c&d \end{smallmatrix}\bigr)$ $\Bigl(\begin{mmatrix} a&b\\c&d \end{mmatrix}\Bigr)$ $\begin{pmatrix} a&b\\c&d \end{pmatrix}$

This package actually has more features. Please refer to ftp://ftp.sunsite.utk.edu/pub/CTAN/macros/latex/contrib/ncctools/doc/nccmath.pdf.

4.14 Theorems: The amsthm Package[1].

The amsthm package provides an enhanced version of standard LATEX's `\newtheorem` command for specifying theorem-like environments.

The `\newtheorem` command has two mandatory arguments. The first is the environment name that the author would like to use for this element. The second is the heading text:

```
\newtheorem{name}{heading}
```

[1] This section is adapted from *The LATEX Companion* [1]

If \newtheorem* is used instead, no automatic numbers will be generated for the environments.

```
\newtheorem{lem}{Lemma}
\newtheorem*{ML}{Colin's Lemma}
\begin{lem}[Main]
The \LaTeX\ Mathematics Companion
complements any math mode introduction.
\end{lem}
\begin{ML}
The \LaTeX\ Mathematics Companion
contains many package descriptions.
\end{ML}
```

Lemma 1 (Main)**.** *The LaTeX Mathematics Companion complements any math mode introduction.*

Colin's Lemma. *The LaTeX Mathematics Companion contains many package descriptions.*

In addition to the two mandatory arguments, \newtheorem has two mutually exclusive optional arguments. They affect the sequencing and hierarchy of the numbering:

```
\newtheorem{name}[use-counter]{heading}
\newthoerem{name}{heading}[number-within]
```

By default, each kind of theorem-like environment is numbered independently. Thus, if you have lemmas, theorems, and some examples interspersed, they will be numbered something like this: Example 1, Lemma 1, Lemma 2, Theorem 1, Example 2, Lemma 3, Theorem 2. If, for example, you want the lemmas and theorems to share the same numbering sequence—Example 1, Lemma 1, Lemma 2, Theorem 3, Example 2, Lemma 4, Theorem 5—then you should indicate the desired relationship as follows:

```
\newtheorem{thm}{Theorem}  \newtheorem{lem}[thm]{Lemma}
```

The optional use-counter argument (value thm) in the second statement means that the lem environment should share the thm numbering sequence instead of having its own independent sequence.

To have a theorem environment numbered subordinately within a sectional unit—for example, to get exercises numbered Exercise 2.1, Exercise 2.2, and so on, in section 2—put the name of the parent counter in square brackets in the final position:

```
\newthoerem{exa}[Exercise][section]
```

The specification part of the amsthm package supports the notion of a current theorem style, which determines the formatting that will be set up by a collection of \newtheorem commands.

```
\theoremstyle{style}
```

The three theorem styles provided by the package are plain, definition, and remark; they specify different typographical treatments that give the environments a visual emphasis corresponding to their relative importance. The details of this typographical treatment may vary depending on the document class, but typically type the plain style produces italic body text and the other two styles produce Roman body text.

To create new theorem-like environments in these styles, divide your \newtheorem declarations into groups and preface each group with the appropriate \theoremstyle. If no \theoremstyle command is given, the style used will be plain. For example,

```
\theoremstyle{plain}
\newtheorem{thm}{Theorem}
\theoremstyle{definition}
\newtheorem{defn}[thm]{Definition}
\theoremstyle{remark}
\newtheorem*{rem}{Remark}
\begin{defn}
A typographical challenge is\ldots.
\end{defn}
\begin{thm}
There are no \ldots
\end{thm}
\begin{rem}
The proof is left to the reader.
\end{rem}
```

Definition 1. A typographical challenge is. . . .

Theorem 2. *There are no . . .*

Remark. The proof is left to the reader.

You can further customize your theorem-like environments:

```
\newtheoremstyle{note}      % <name>
                {3pt}       % <Space above>
                {3pt}       % <Space below>
                {}          % <Body font>
                {}          % <Indent amounti>
                {\itshape}  % <Theorem head font>
                {:}         % <Punctuation after theorem head>
                {.5em}      % <Space after theorem head>
                {} % <Theorem head spec (can be left empty, meaning ‘‘normal’’)>
```

A predefined proof environment provided by the amsthm package produces the heading "Proof" with appropriate spacing and punctuation. The proof environment is primarily intended for short proofs, no more than a page or two in length; longer proofs are usually better done as a separate `\section` or `\subsection` in your document. An optional argument of the proof environment allows you to substitute a different name for the standard "Proof." If you want the proof heading to be, say, "Proof of the Main Theorem," then write

```
\begin{proof}[Proof of the Main Theorem]
```

A QED symbol, □, is automatically appended at the end of a proof environment. To substitute a different end-of-proof symbol, use `\renewcommand` to redefine the command `\qedsymbol`. For a long proof done as a subsection or section instead of with the proof environment, you can obtain the symbol and the usual amount of preceding space by using `\qed`. Placement of the QED symbol can be problematic if the last part of a proof environment is a displayed equation or list environment or something of that nature. In that case put a `\qedhere` command at the place where the QED symbol should appear, for example,

```
\begin{proof}
\ldots
\begin{equation}
G(t)=L\gamma!\,t^{-\gamma}+
t^{-\delta}\eta(t) \qedhere
\end{equation}
\end{proof}
```

Proof. . . .

$$G(t) = L\gamma!\, t^{-\gamma} + t^{-\delta}\eta(t) \qquad (4.7)$$

□

For more details, refer to ftp://mirror.sg.depaul.edu/pub/TeX/CTAN/macros/latex/required/amslatex/classes/amsthdoc.pdf.

✎ Exercise

1. The nccthm and ntheorem packages also enhance the standard LaTeX's `theorem` environment. Try them out!

4.15 Two Powerful Packages Mentioned Merely in Passing

There are two more amazing, brilliant packages I really want to show you. However, they cause some compatibility issues which disturb the compilation of my book. What's more, they themselves come with easy-to-read and well-written documentations. Anyway, I'd like to give you an overview.

The first package to mention is the nath package. Here is an excerpt from the documentation of the package:

> Nath is a LaTeX style to separate presentation and content in mathematical typography. The style delivers a particular context-dependent presentation on the basis of a rather coarse context-independent notation. Although essentially backward compatible with LaTeX, Nath aims at producing traditional math typography even from sources devoid of aesthetic ambitions. Its name is derived from "*na*tural ma*th* notation."

This description is quite accurate: The nath package has quite some compatibility issues, but its functions are amazing! I strongly recommend that you read its marvelous documentation, which could be obtained at http://texcatalogue.sarovar.org/entries/nath.html.

Another one is the mathenv package, which can ease your work to a large extent. The documentation is available at ftp://ftp.sunsite.utk.edu/pub/CTAN/macros/latex/contrib/bosisio/mathenv.html.

CHAPTER 5

Further Directions

Just like I mentioned in the very beginning of this book, it doesn't try to cover everything. It is simply a *companion*. Here is where you look for more details:

- If you need to know about the technical details of the math mode of TEX, refer to chapters 16–18 of *The TEXbook* [9].
- If you need to learn more about TEX's math styles, refer to chapter 17 of *The TEXbook* [9].
- If you need a comprehensive list of symbols, refer to *The Comprehensive LATEX Symbol List* [10]. But this is a huge list, you might consider section 8.9 of *The LATEX Companion* [1] instead.
- If you need to learn more about fonts in math mode, refer to chapters 7–8 of *The LATEX Companion* [1]. (Section 8.8.3 of [1] describes a collection of font packages.)
- If you need to learn about other LATEX packages for math typesetting, refer to `http://texcatalogue.sarovar.org/`.
- Part III of *Math mode* [3] introduces some basic TEX commands.
- If you need to do some special and funny things, refer to parts V and VI of *Math mode* [3].
- If you need to learn more rules of mathematical typesetting, refer to *Mathematics into Type* [8].
- If you need some detailed typographic information, refer to *The Chicago Manual of Style* [7].
- If you need to learn which symbols and notations to use in math composition, refer to *ANS/IEEE Std 260.3-1993* [4], *ISO 31-11:1992* [5], and *CWS 260.3-2005* [6]. *CWS 260.3-2005* I'd recommend [6] written by me, because it combines the advantages of [4] and [5]. ☺
- This book doesn't talk about graphs and pictures at all, because there are so many well written books that I couldn't possibly excel. One that I highly recommend is *The LATEX Graphics Companion* [2].

BIBLIOGRAPHY

[1] Mittel, Frank, and Michel Goossens. *The LaTeX Companion, Second Edition*. Boston: Addison-Wesley, 2004.

[2] Goosens, Michel, Sebastian Rahtz, and Frank Mittelbach. *The LaTeX Graphics Companion*. Boston: Addison-Wesley, 1997.

[3] Voß, Herbert. *Math mode – v.2.05*, 2005, http://perce.de/LaTeX/math/mathmode/Mathmode.pdf.

[4] IEEE. *ANSI/IEEE Std 260.3-1993: Mathematical Signs and Symbols for Use in Physical Sciences and Technology*. New York: The Institute of Electrical and Electronics Engineers, Inc., 1993

[5] ISO/TC 12. *ISO 31-11: Quantities and units — Part 11 Mathematical signs and symbols for use in the physical sciences and technology*. Switzerland: ISO, 1992

[6] Gai, Helin. *CWS 260.3-2005: Mathematical Signs and Symbols for Use in Physical Sciences and Technology*, 2005, http://www.ctex.org/forums/index.php?act=Attach&type=post&id=18111

[7] The University of Chicago Press. *The Chicago Manual of Style, 15th edition*. Chicago: The University of Chicago Press, 2003

[8] Swanson, Ellen, update by Arlene O'Sean and Antoinette Schleyer, *Mathematics into Type, Updated Edition*. Provindence: American Mathematical Society, 1999

[9] Knuth, Donald. *The TeXbook*. Boston: Addison-Wesley, 1986.

[10] Pakin, Scott. *The Comprehensive LaTeX Symbol List*, 2005, http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-letter.pdf

INDEX