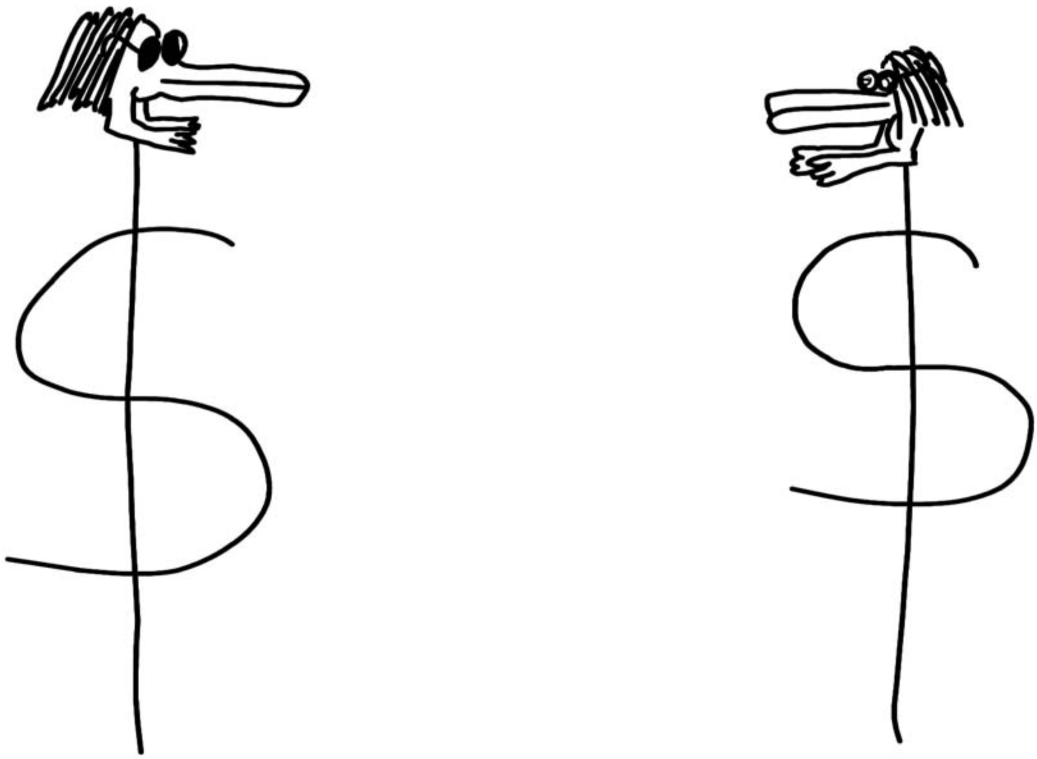
Chapter 9

Getting Started with LaTeX



The purpose of this chapter is to help you begin using LaTeX (the most popular version of TeX), a mathematical typesetting system in which you can create, edit, typeset, and view documents containing mathematics. LaTeX is easy to learn and use. You can quickly start producing output and learn more as you go. The discussion and exercises contain many examples for you to try.

9.1 What is TeX?

TeX is a typesetting program that was invented by Donald Knuth "for the creation of beautiful books—and especially for books that contain a lot of mathematics" [29]. TeX is especially useful for typesetting mathematical documents and other documents containing complicated symbols and formatting.

TeX is a markup language. In a markup language, the user specifies how the document is to look via commands. The typesetting software interprets the commands in order to produce the pages.

Thus, TeX is both a language and a program. As a whole, it is a typesetting system.

9.2 What is LaTeX?

LaTeX is an easy-to-use version of TeX designed by Leslie Lamport. "Think of LaTeX as a house built with the lumber and nails provided by TeX" [33].

In this chapter, all examples are written in LaTeX.

> LaTeX is now extremely popular in the scientific and academic communities, and it is used extensively in industry. It has become a *lingua franca* of the scientific world; scientists send their papers electronically to colleagues around the world in the form of LaTeX input. [33]

9.3 How to create LaTeX files

LaTeX files are text files, which you should edit with simple text editors such as Notepad or Notepad++. Although they look similar, it's important to not use word processors such as Microsoft Word to create LaTeX files.

Many people use a dedicated editor for their LaTeX files. For Windows, PCTeX comes with a built-in editor, and TeXnicCenter is a popular free alternative (that works in coordination with MiKTeX). Mac users may prefer TeXShop, and Linux users might use Vi or Emacs (popular editors in the free software culture).

9.4 How to create and typeset a simple LaTeX document

A minimal LaTeX file consists of a `\documentclass` command, a `\begin{document}` command, an `\end{document}` command, and perhaps a line of text.

Note. Commands are preceded by the `\` (backslash) symbol. Commands are case-sensitive.

Example 9.1. Here is a minimal LaTeX file.

```
\documentclass{article}
\begin{document}
This is      some           sample text.

Here is a new paragraph.
\end{document}
```

Note. The `article` document class is appropriate for many writing projects.

Note. LaTeX ignores most extra spaces in a file. A blank line (or lines) tells LaTeX to start a new paragraph.

Save the file as, say, `myfile.tex`. Then typeset the file. Programs like PCTeX, TeXnicCenter, and TeXShop will have a button that runs LaTeX to typeset your documents. Linux users will probably type a command, like `latex myfile.tex`, to do the same thing.

LaTeX will usually produce a file called `myfile.dvi` (`dvi` stands for "device independent"), although the TeXShop creates `pdf` files by default. The `dvi` or `pdf` file is displayed by a built-in viewer. Linux users will often invoke a viewer manually, perhaps with the command `xdvi myfile.dvi`.

Your displayed document should look like this:

> This is some sample text.
>
> Here is a new paragraph.

Congratulations! You have created and typeset your first document.

9.5 How to add basic information to your document

In this section you will learn how to add title, author, and date information to your documents. You will also learn how to create a numbered list (for example, a list of problem solutions in a homework paper), and you will learn how to emphasize text.

Title, author, and date commands

You can add a title, author, and date to your document with `\title`, `\author`, `\date`, and `\maketitle` commands. Try the following example.

Example 9.2. This file adds a title, author, and date to the file in Example 9.1.

```
\documentclass{article}
\title{My Document}
\author{A. Student}
\date{January 1, 2011}
\begin{document}
\maketitle
This is      some           sample text.

Here is a new paragraph.
\end{document}
```

Note. The part of the file preceding the `\begin{document}` command is called the *preamble*.

After saving the file and typesetting it, the output should look like this:

My Document

A. Student

January 1, 2011

This is some sample text.

Here is a new paragraph.

The enumerate environment

If your document is a homework assignment, you may want it to contain a numbered list of problem solutions. This can be accomplished with the `enumerate` environment.

The `enumerate` environment is opened with a `\begin{enumerate}` command and closed with an `\end{enumerate}` command. Within the environment, each item to be enumerated is preceded by an `\item` command. LaTeX will compute the item numbers for you, adjusting automatically as you add or remove items from the list.

Note. If you use several environments at once, they must be nested.

Highlighted text

Boldfaced text is produced with a `\textbf` command. Emphasized text, such as appears in a mathematical definition or a reference to a book title, is produced with an `\emph` command. The way that text is emphasized depends on context. Usually it will be italicized, but in italicized sections emphasis will in fact be non-italicized as appropriate to convey the meaning you want. Remember that LaTeX is a markup language; we tell LaTeX what we want to emphasize and it decides how to do that.

Example 9.3. This file creates a document containing a list of problem solutions.

```
\documentclass{article}
\title{My Document}
\author{A. Student}
\date{January 1, 2011}
\begin{document}
\maketitle

\begin{enumerate}

\item Here is some \textbf{boldfaced} text.

\item Here is some \emph{emphasized} text.

\end{enumerate}

\end{document}
```

After saving the file and typesetting it, the output should look like this:

My Document

A. Student

January 1, 2011

1. Here is some **boldfaced** text.
2. Here is some *emphasized* text.

Note. You can control the size of text with commands such as `\Large`, `\large`, `\normalsize`, `\small`, and `\tiny`. The most common way to use a size command is to place it at the beginning of a section of text enclosed in braces (which indicate grouping in LaTeX), e.g., `{\Large Hello!}`.

9.6 How to do elementary mathematical typesetting

In this section you will learn how to include simple mathematical expressions in your LaTeX documents. Once you are familiar with using LaTeX, you will find that you can typeset math about as easily as you can read it from left to right.

Math mode

There are several ways to put mathematical expressions into a document. The most common method is to enter *math mode* by inserting a math expression between a pair of `$` signs. Thus, the input `$x=3$` will produce the output $x = 3$. Observe that the font for a math x is different from the font for a normal x. It is also slightly different from the font for an italicized *x*.

The symbols `\(` and `\)` may be used instead of a pair of `$` signs. In fact, you may prefer to use `\(` and `\)` since you may spot mistakes more quickly if you use the balanced version of the delimiters, particularly if you accidently leave one out or put an extra one in.

Centered mathematical expressions set off from the text are called “displayed” expressions, and they are generated with the `\[` and `\]` symbols. Thus, the input

```
\[y=5\]
```

produces the output

$$y = 5$$

(this is displayed).

Note. In math mode, spaces don't matter at all; they are ignored. If you want extra space, use `\,` (thin space), `\:` (medium space), or `\;` (thick space), commands.

Math alphabets

Greek letters are invoked by commands such as `\alpha`, `\beta`, `\gamma`, etc., for lowercase letters, and `\Alpha`, `\Delta`, `\Theta`, etc., for uppercase letters.

$\emptyset$	`\emptyset`	$\bigcup$	`\bigcup`	$\leftarrow$	`\leftarrow`	$\div$	`\div`
$\in$	`\in`	$\bigcap$	`\bigcap`	$\rightarrow$	`\rightarrow`	$\times$	`\times`
$\notin$	`\notin`	$\setminus$	`\setminus`	$\Leftarrow$	`\Leftarrow`	$\pm$	`\pm`
$\subset$	`\subset`	$\approx$	`\approx`	$\Rightarrow$	`\Rightarrow`	$\cdot$	`\cdot`
$\subseteq$	`\subseteq`	$\doteq$	`\doteq`	$\leftrightarrow$	`\leftrightarrow`	$\circ$	`\circ`
$\supset$	`\supset`	$\neq$	`\neq`	$\Leftrightarrow$	`\Leftrightarrow`	$\ast$	`\ast`
$\supseteq$	`\supseteq`	$\geq$	`\geq`	$\cong$	`\cong`	$\cdots$	`\cdots`
$\cup$	`\cup`	$\leq$	`\leq`	∞	`\infty`	$\ldots$	`\ldots`
$\cap$	`\cap`	$\sim$	`\sim`	∂	`\partial`	$\prime$	`\prime`

TABLE 9.1: Mathematical symbols.

Boldfaced mathematical text is produced with a `\mathbf` command.

Math symbols

The following symbols are ordinary keyboard symbols and do not require special commands when used in math mode:

$$+ \quad - \quad / \quad (\quad) \quad [\quad] \quad < \quad > \quad = \quad .$$

The symbols `{` and `}` are produced with the commands `\{` and `\}`.

Table 9.1 shows some frequently used mathematical symbols and the LaTeX commands that generate them.

Note. Certain sets are usually indicated with boldface. For example, the set of real numbers is denoted **R** (produced by the command `$\mathbf{R}$`).

Math functions

Many mathematical expressions are invoked by simple LaTeX commands. For example, use `\sin` for sin, `\log` for log, `\ln` for ln, `\lim` for lim, etc.

Math structures

Some useful mathematical structures are listed in Table 9.2. For example, to produce $\lim_{x\rightarrow\infty}$, use the command `$\lim_{x \rightarrow \infty}$`.

Example 9.4. Here is a file containing some simple mathematical expressions.

```
\documentclass{article}
\title{My Document}
\author{A. Student}
\date{January 1, 2011}
\begin{document}
\maketitle

\begin{enumerate}

\item Suppose that $x=137$.
```

Structure	Command	Example Input	Example Output
subscript	`_{}`	`x_{10}`	x_{10}
superscript	`^{}`	`3^{20}`	3^{20}
fraction	`\frac{}{}`	`\frac{a+b}{x+y}`	$\frac{a+b}{x+y}$
square root	`\sqrt{}`	`\sqrt{x+y}`	$\sqrt{x+y}$
nth root	`\sqrt[n]{}`	`\sqrt[3]{x+y}`	$\sqrt[3]{x+y}$
sum	`\sum_{}^{}`	`\sum_{i=1}^{10}i^2`	$\sum_{i=1}^{10} i^2$
product	`\prod_{}^{}`	`\prod_{i=1}^{10}i^2`	$\prod_{i=1}^{10} i^2$
integral	`\int_{}^{}`	`\int_{0}^{\infty}x\,dx`	$\int_0^\infty x\,dx$

TABLE 9.2: Mathematical structures.

```
\item Let $n=3$.  Then $n^2+1=10$.

\item The curve $y= \sqrt{x}$, where $x \geq 0$, is concave downward.

\item If $\sin \theta = 0$ and $0 \leq \theta < 2 \pi$,
then $\theta=0$ or $\theta=\pi$.

\item It is not always true that
\[\frac{a+b}{c+d}=\frac{a}{c}+\frac{b}{d}.\]

\end{enumerate}

\end{document}
```

After saving the file and typesetting it, the output should look like this:

My Document

A. Student

January 1, 2011

1. Suppose that $x = 137$.
2. Let $n = 3$. Then $n^2 + 1 = 10$.
3. The curve $y = \sqrt{x}$, where $x \geq 0$, is concave downward.
4. If $\sin\theta = 0$ and $0 \leq \theta < 2\pi$, then $\theta = 0$ or $\theta = \pi$.
5. It is not always true that
$$\frac{a+b}{c+d} = \frac{a}{c} + \frac{b}{d}.$$

<table>
<tr><th>Delimiter</th><th>Command</th><th>Delimiter</th><th>Command</th></tr>
<tr><td>$($</td><td><code>\left(</code></td><td>$)$</td><td><code>\right)</code></td></tr>
<tr><td>$[$</td><td><code>\left[</code></td><td>$]$</td><td><code>\right]</code></td></tr>
<tr><td>$\{$</td><td><code>\left\{</code></td><td>$\}$</td><td><code>\right\}</code></td></tr>
<tr><td>$|$</td><td><code>\left|</code> or <code>\right|</code></td><td>$\|$</td><td><code>\left\|</code> or <code>\right\|</code></td></tr>
<tr><td>$\lfloor$</td><td><code>\lfloor</code></td><td>$\rfloor$</td><td><code>\rfloor</code></td></tr>
<tr><td>$\lceil$</td><td><code>\lceil</code></td><td>$\rceil$</td><td><code>\rceil</code></td></tr>
</table>

TABLE 9.3: Delimiters.

9.7 How to do advanced mathematical typesetting

In this section you will learn how to include complex mathematical expressions in your documents.

Delimiters

Sometimes you need parentheses that are larger than normal, such as those in the expression

$$\left(\frac{a+b}{x+y}\right)^{1/3}.$$

Large delimiters (parentheses, square brackets, curly brackets, etc.) are produced with `\left` and `\right` commands. For example, the expression above is produced with the input `\[\left(\frac{a+b}{x+y}\right)^{1/3}\]`.

Table 9.3 lists the available delimiters and the commands that produce them.

Note. Delimiters must occur in left–right pairs, but the delimiters in a pair may be of different types. If you want only one delimiter, then pair it with a period. For example, to typeset a piece-wise defined function, a left brace `\left\{` would be paired with a `\right.`, which won't show up in the display.

Arrays

Arrays (used in constructing matrices, for example) are created in the `array` environment. The `array` environment is opened with a `\begin{array}` command and closed with an `\end{array}` command. Attached to the `\begin{array}` command is a string of letters indicating the number of columns in the array and the type of alignment in each column. For example, the string `{llrr}` declares that the array contains four columns, the first and second columns aligned on the left and the third and fourth columns aligned on the right.

Line breaks (for all but the last line of the array) are indicated by two backslashes, `\\`. For example, to produce the matrix

$$\left[\begin{array}{ccc} a & b & c \\ d & e & f \\ g & h & i \end{array}\right]$$

use the following commands.

```
\[
\left[
\begin{array}{ccc}
a & b & c\\
d & e & f\\
g & h & i
\end{array}
\right]
\]
```

Multi-line expressions

Multi-line expressions (e.g., in a chain of equalities) are created in the `\eqnarray*` environment. The environment is opened with a `\begin{eqnarray*}` command and closed with an `\end{eqnarray*}` command. Within the environment, each line may consist of three parts: two expressions and a relational symbol (e.g., $=$ or $\leq$). The parts are separated by & symbols. As with arrays, each line (except the last) ends with a `\\` command.

For example, to produce the output

$$\begin{eqnarray*} e^x & = & \frac{x^0}{0!}+\frac{x^1}{1!}+\frac{x^2}{2!}+\frac{x^3}{3!}+\cdots\\ e^{-1} & = & \frac{{(-1)}^0}{0!}+\frac{{(-1)}^1}{1!}+\frac{{(-1)}^2}{2!}+\frac{{(-1)}^3}{3!}+\cdots\\ & = & \frac{1}{0!}-\frac{1}{1!}+\frac{1}{2!}-\frac{1}{3!}+\cdots \end{eqnarray*}$$

use the following commands.

```
\begin{eqnarray*}
e^x & = & \frac{x^0}{0!}+\frac{x^1}{1!}
  +\frac{x^2}{2!}+\frac{x^3}{3!}+\cdots\\
e^{-1} & = & \frac{{(-1)}^0}{0!}+\frac{{(-1)}^1}{1!}
  +\frac{{(-1)}^2}{2!}+\frac{{(-1)}^3}{3!}+\cdots\\
& = & \frac{1}{0!}-\frac{1}{1!}+\frac{1}{2!}
  -\frac{1}{3!}+\cdots
\end{eqnarray*}
```

Note. The `*` in the `eqnarray*` environment causes the equations produced to be unnumbered. If you want numbered equations, use `eqnarray` instead of `eqnarray*`.

Example 9.5. Here is a file containing several complex mathematical expressions.

Note. When typesetting long, complicated documents, it's a good idea to input the file in small blocks, typesetting at each stage. This way, if your document contains an error, you'll be able to find and correct it easily.

```
\documentclass{article}
\title{My Document}
\author{A. Student}
\date{January 1, 2011}
\begin{document}
\maketitle
```

```
\begin{enumerate}

\item Let $\mathbf{x}=(x_1,\ldots,x_n)$,
where the $x_i$ are nonnegative real numbers.
Set
\[
M_r(\mathbf{x}) = \left(\frac{x_1^r+x_2^r
+\cdots+x_n^r}{n}\right)^{1/r},
\; \; r \in \mathbf{R} \setminus \{0\},
\]
and
\[
M_0(\mathbf{x})=\left( x_1 x_2 \ldots x_n \right)^{1/n}.
\]
We call $M_r(\mathbf{x})$ the \emph{$r$th power mean}
of $\mathbf{x}$.

Claim:
\[
\lim_{r \rightarrow 0} M_r(\mathbf{x}) =
M_0(\mathbf{x}).
\]

\item Define
\[
V_n=
\left[
\begin{array}{ccccc}
1 & 1 & 1 & \ldots & 1\\
x_1 & x_2 & x_3 & \ldots & x_n\\
x_1^2 & x_2^2 & x_3^2 & \ldots & x_n^2\\
\vdots & \vdots & \vdots & \ddots & \vdots\\
x_1^{n-1} & x_2^{n-1} & x_3^{n-1} & \ldots & x_n^{n-1}
\end{array}
\right].
\]
We call $V_n$ the \emph{Vandermonde matrix} of order $n$.

Claim:
\[
\det V_n = \prod_{1 \leq i < j \leq n}(x_j-x_i).
\]

\end{enumerate}

\end{document}
```

After saving the file and typesetting it, the output should look like this:

My Document

A. Student

January 1, 2011

1. Let $\mathbf{x} = (x_1, \ldots, x_n)$, where the x_i are nonnegative real numbers. Set

$$M_r(\mathbf{x}) = \left(\frac{x_1^r + x_2^r + \cdots + x_n^r}{n} \right)^{1/r}, \quad r \in \mathbf{R} \setminus \{0\},$$

   and

$$M_0(\mathbf{x}) = (x_1 x_2 \ldots x_n)^{1/n}.$$

   We call $M_r(\mathbf{x})$ the *rth power mean* of $\mathbf{x}$.

   Claim:

$$\lim_{r \to 0} M_r(\mathbf{x}) = M_0(\mathbf{x}).$$

2. Define

$$V_n = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ x_1 & x_2 & x_3 & \ldots & x_n \\ x_1^2 & x_2^2 & x_3^2 & \ldots & x_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & x_3^{n-1} & \ldots & x_n^{n-1} \end{bmatrix}.$$

   We call V_n the *Vandermonde matrix* of order n.

   Claim:

$$\det V_n = \prod_{1 \leq i < j \leq n} (x_j - x_i).$$

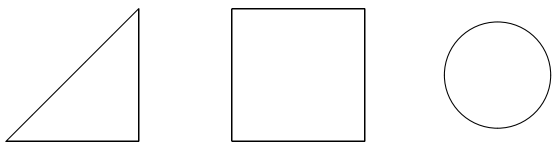
9.8 How to use graphics

There are many options for including graphics in your LaTeX documents. The simplest method is to use LaTeX's `picture` environment. For more complex pictures, you may need to use PSTricks (see Chapter 10), a package for creating images within LaTeX. Beyond that, you can include graphics created elsewhere, such as in a computer algebra system (see, for example, Chapters 12 and 13), or in applications such as Graphviz (Graph Visualization Software) and gnuplot (a graphing utility). You can also make precise mathematical images using PostScript (Chapter 17).

Example 9.6. You can use LaTeX's `picture` environment to create (somewhat limited) pictures. Here is a picture of a triangle, a square, and a circle.

```
\begin{picture}(250,75)
% draw triangle
\put(15,10){\line(1,0){50}}
\put(65,10){\line(0,1){50}}
```

```
\put(65,60){\line(-1,-1){50}}
% draw square
\put(100,10){\line(1,0){50}}
\put(150,10){\line(0,1){50}}
\put(150,60){\line(-1,0){50}}
\put(100,60){\line(0,-1){50}}
% draw circle
\put(200,35){\circle{40}}
\end{picture}
```



The commands are fairly self-explanatory. The argument `(250,75)` sets up space for a picture 250×75 units. Coordinates refer to a rectangular coordinate system with the origin at the lower-left corner. The command `\line(1,0){50}` produces a line of length 50 in the direction given by the vector $(1, 0)$. The `\circle{40}` command produces a circle of diameter 40. The `\put` commands tell where these graphical elements are placed.

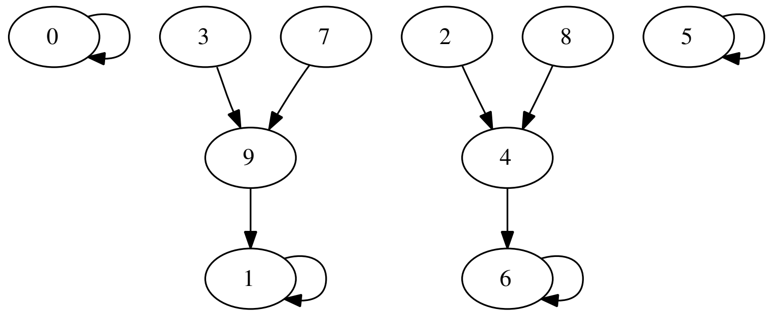
If you want to include a graphics file created outside your LaTeX document, put a `\usepackage{graphicx}` in your document's preamble. Use an `\includegraphics` command where you want an image to appear. For example, the command

```
\includegraphics[width=3in,height=3in]{5-12-13triangle.eps}
```

inserts the Encapsulated PostScript graphics file called "5-12-13triangle.eps," scaled as indicated. LaTeX can understand several types of graphics files, but PDF or PNG images generally make sense for LaTeX documents that are converted to PDF (Portable Document Format), whereas images in PS (PostScript) or EPS (Encapsulated PostScript) make sense for documents that are made into PostScript.

Example 9.7. Graphviz is a good program for quickly making graph network diagrams (directed or undirected) to use with `\includegraphics`. The following Graphviz input makes a directed graph depicting the squaring map on the integers modulo 10.

```
digraph {
"0" -> "0"
"1" -> "1"
"2" -> "4"
"3" -> "9"
"4" -> "6"
"5" -> "5"
"6" -> "6"
"7" -> "9"
"8" -> "4"
"9" -> "1"
}
```



Graphviz doesn't allow the use of LaTeX commands as PSTricks does. The Graphviz Web site is at:

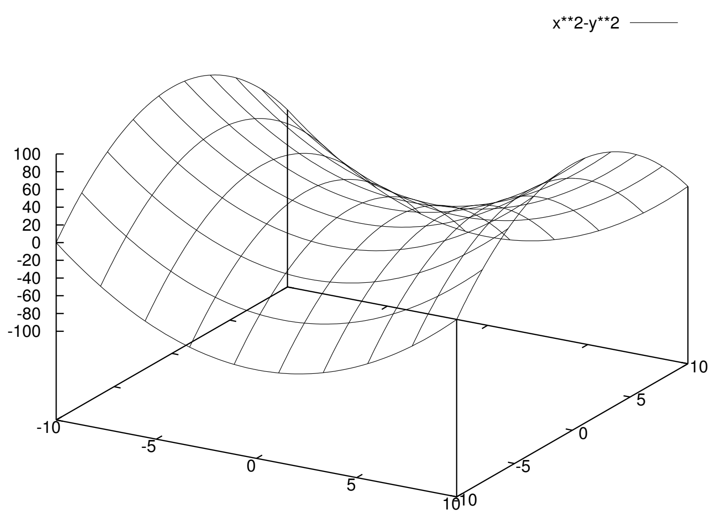
www.graphviz.org

For producing pictures of function plots, gnuplot is an easy-to-use tool.

Example 9.8. The gnuplot input

```
splot x**2-y**2
```

produces the saddle surface $z = x^2 - y^2$.



Maxima (Chapter 12) and Octave (Chapter 13) use gnuplot as their plotting engine. The gnuplot Web site is at:

www.gnuplot.info

Using PostScript, you can create precise mathematical illustrations using drawing commands within a coordinate system. See Chapter 17 for an introduction.

9.9 How to learn more

There are many aspects of LaTeX not discussed in this introduction, such as user-defined environments and packages. Here are some resources for you to investigate to learn more.

The definitive book about TeX is [29]. A good beginning book is [53].

In order to learn more about LaTeX, an excellent place to start is [33]. Other good introductory books are [12], [20], and [21]. To go further, you may want to consult [17], [19], and [18], which are all advanced books.

Some useful Web sites about TeX and LaTeX are www.ctan.org ("The Comprehensive TeX Archive Network"),

www.tug.org/interest.html

("The TeX Users Group"),

www.emerson.emory.edu/services/latex/latex2e/latex2e_toc.html

("LaTeX Help"), and

http://en.wikibooks.org/wiki/LaTeX

(a LaTeX Wikibook).

Information about PCTeX (a user-friendly LaTeX system) is available at www.pctex.com ("PCTeX HomePage").

The project page for TeXnicCenter is www.texniccenter.org. Since TeXnicCenter is only a "front end" for LaTeX, you may want to first get MiKTeX from miktex.org. You may also want GSview for viewing PostScript files; it is available from:

pages.cs.wisc.edu/~ghost/gsview

TeXShop is available from texshop.org and bundles a complete LaTeX system for Mac OS X.

The TeX Live project is hosted at www.tug.org/texlive. This project provides the LaTeX that is most commonly installed on Linux and Unix systems.

General references about mathematical writing and typesetting are [31], [30], and [55].

Exercises

1. What is wrong with the following LaTeX input? What is the correct way to do it?

```
If m=1 and n=2, then m+n=3.
```

2. What is wrong with the following input? What is the right way to do it?

```
If $theta = pi$, then $sin theta = 0$.
```

3. What is illogical about the following LaTeX input? What is a better way to do it?

```
If $x=3,$ then $3x=9.$
```

4. Make the following equations.

$$3^3 + 4^3 + 5^3 = 6^3$$

$$\sqrt{100} = 10$$

$$(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$$

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$$

$$\frac{\pi}{4} = \frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \cdots$$

$$\cos\theta = \sin(90^\circ - \theta)$$

$$e^{i\theta} = \cos\theta + i\sin\theta$$

$$\lim_{\theta\to 0} \frac{\sin\theta}{\theta} = 1$$

$$\lim_{x\to\infty} \frac{\pi(x)}{x/\log x} = 1$$

$$\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi}$$

5. Typeset the following sentences.

Positive numbers a, b, and c are the side lengths of a triangle if and only if $a+b>c$, $b+c>a$, and $c+a>b$.

The area of a triangle with side lengths a, b, c is given by *Heron's formula*:

$$A = \sqrt{s(s-a)(s-b)(s-c)},$$

where s is the semiperimeter $(a+b+c)/2$.

The volume of a regular tetrahedron of edge length 1 is $\sqrt{2}/12$.

The quadratic equation $ax^2 + bx + c = 0$ has roots

$$r_1, r_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

The *derivative* of a function f, denoted f', is defined by

$$f'(x) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h}.$$

A real-valued function f is *convex* on an interval I if

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y),$$

for all $x, y \in I$ and $0 \leq \lambda \leq 1$.

The general solution to the differential equation

$$y'' - 3y' + 2y = 0$$

is

$$y = C_1 e^x + C_2 e^{2x}.$$

The *Fermat number* F_n is defined as

$$F_n = 2^{2^n}, \quad n \geq 0.$$

6. Make the following equations. Notice the large delimiters.

$$\frac{d}{dx}\left(\frac{x}{x+1}\right) = \frac{1}{(x+1)^2}$$

$$\lim_{n\to\infty}\left(1 + \frac{1}{n}\right)^n = e$$

$$\left|\begin{array}{cc} a & b \\ c & d \end{array}\right| = ad - bc$$

$$R_\theta = \left[\begin{array}{cc} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{array}\right]$$

$$\left|\begin{array}{ccc} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{array}\right| = \left|\begin{array}{cc} a_2 & a_3 \\ b_2 & b_3 \end{array}\right|\mathbf{i} - \left|\begin{array}{cc} a_1 & a_3 \\ b_1 & b_3 \end{array}\right|\mathbf{j} + \left|\begin{array}{cc} a_1 & a_2 \\ b_1 & b_2 \end{array}\right|\mathbf{k}$$

$$\left[\begin{array}{cc} a_{11} & a_{12} \\ a_{21} & a_{22} \end{array}\right]\left[\begin{array}{cc} b_{11} & b_{12} \\ b_{21} & b_{22} \end{array}\right] = \left[\begin{array}{cc} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{array}\right]$$

$$f(x) = \left\{\begin{array}{ll} -x^2, & x < 0 \\ x^2, & 0 \leq x \leq 2 \\ 4, & x > 2 \end{array}\right.$$

7. Make the following multi-line equations.

$$\begin{aligned}
1+2 &= 3\\
4+5+6 &= 7+8\\
9+10+11+12 &= 13+14+15\\
16+17+18+19+20 &= 21+22+23+24\\
25+26+27+28+29+30 &= 31+32+33+34+35
\end{aligned}$$

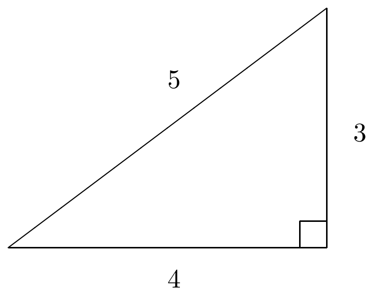
$$\begin{aligned}
(a+b)^2 &= (a+b)(a+b)\\
&= (a+b)a+(a+b)b\\
&= a(a+b)+b(a+b)\\
&= a^2+ab+ba+b^2\\
&= a^2+ab+ab+b^2\\
&= a^2+2ab+b^2
\end{aligned}$$

$$\begin{aligned}
\tan(\alpha+\beta+\gamma) &= \frac{\tan(\alpha+\beta)+\tan\gamma}{1-\tan(\alpha+\beta)\tan\gamma}\\
&= \frac{\frac{\tan\alpha+\tan\beta}{1-\tan\alpha\tan\beta}+\tan\gamma}{1-\left(\frac{\tan\alpha+\tan\beta}{1-\tan\alpha\tan\beta}\right)\tan\gamma}\\
&= \frac{\tan\alpha+\tan\beta+(1-\tan\alpha\tan\beta)\tan\gamma}{1-\tan\alpha\tan\beta-(\tan\alpha+\tan\beta)\tan\gamma}\\
&= \frac{\tan\alpha+\tan\beta+\tan\gamma-\tan\alpha\tan\beta\tan\gamma}{1-\tan\alpha\tan\beta-\tan\alpha\tan\gamma-\tan\beta\tan\gamma}
\end{aligned}$$

$$\begin{aligned}
\prod_p\left(1-\frac{1}{p^2}\right) &= \prod_p\frac{1}{1+\frac{1}{p^2}+\frac{1}{p^4}+\cdots}\\
&= \left(\prod_p\left(1+\frac{1}{p^2}+\frac{1}{p^4}+\cdots\right)\right)^{-1}\\
&= \left(1+\frac{1}{2^2}+\frac{1}{3^2}+\frac{1}{4^2}+\cdots\right)^{-1}\\
&= \frac{6}{\pi^2}
\end{aligned}$$

8. You can center text or graphics using the `center` environment. (Open the environment with `\begin{center}` and close it with `\end{center}`.) Use the `center` environment and an `\includegraphics` command to put a mathematical picture in a document. You can create the picture yourself or take it from another source. Remember to put a `\usepackage{graphicx}` command in the preamble.

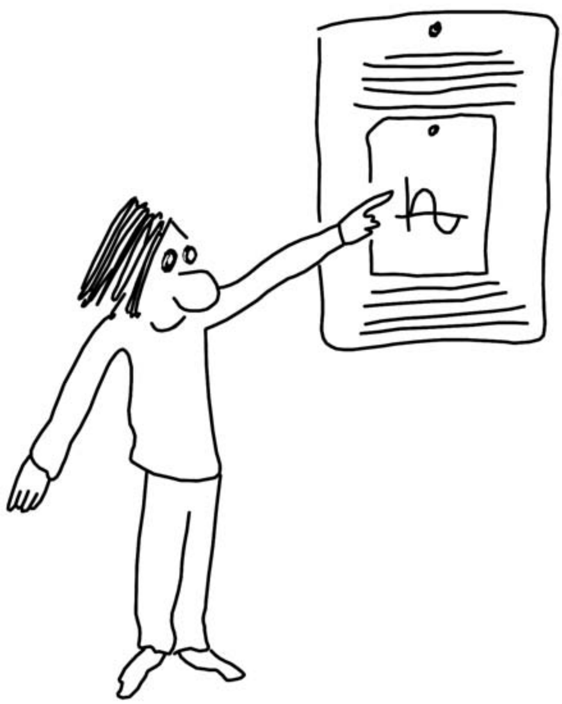
9. Use LaTeX's `picture` environment to make a picture of a 3–4–5 Pythagorean triangle, as below.



10. Add the inscribed circle of the triangle to the picture of the previous exercise.

Chapter 10

Getting Started with PSTricks



Now that you have some experience with LaTeX (see Chapter 9), you may want to spice up your LaTeX documents with some great graphics. Because of its flexibility and ease of use, the PSTricks package is a good choice for making images. The best way to learn PSTricks is by playing with examples.

10.1 What is PSTricks?

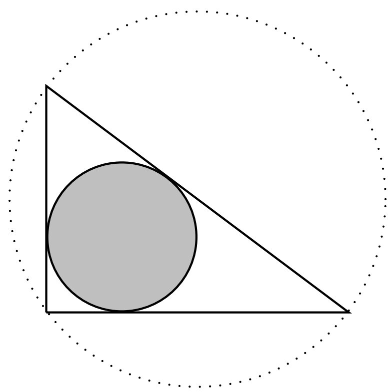
PSTricks, written by Timothy van Zandt, is a package that can be included in LaTeX documents. With PSTricks, you can use the power of PostScript's image creation language (see Chapter 17) within your LaTeX code. Many built-in commands make PSTricks easy to use.

10.2 How to make simple pictures

To use PSTricks, include a `\usepackage{pstricks}` command in the preamble of your LaTeX document.

Example 10.1. Let's start with a little figure displaying lines and circles.

```
\begin{pspicture}(5,5)
\psline(1,1)(5,1)(1,4)(1,1)
\pscircle[linestyle=dotted](3,2.5){2.5}
\pscircle[fillstyle=solid,fillcolor=lightgray](2,2){1}
\end{pspicture}
```



Here are the commands used to make the figure.

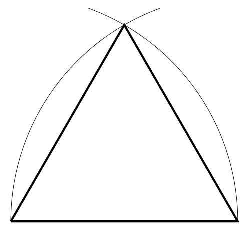
- The command `\begin{pspicture}(5,5)` starts the picture and sets aside space in a coordinate system with lower-left corner $(0, 0)$ and upper-right corner $(5, 5)$. The default units are equal to 1 cm.
- The command `\psline(1,1)(5,1)(1,4)(1,1)` draws a line path from $(1, 1)$ to $(5, 1)$ to $(1, 4)$ to $(1, 1)$, i.e., a triangle.
- The command `\pscircle(3,2.5){2.5}` draws a circle with center $(3, 2.5)$ and radius 2.5. Setting the parameter `linestyle` to `dotted` causes the circumference of the circle to be shown with a dotted line.
- The command `\pscircle(2,2){1}` draws a circle with center $(2, 2)$ and radius 1. Setting the parameters `fillstyle` and `fillcolor` to `solid` and `lightgray`, respectively, causes the interior of the circle to be shaded light gray.
- The command `\end{pspicture}` ends the picture.

Let's do another simple example.

Example 10.2. We draw a figure illustrating the straightedge and compass construction of an equilateral triangle.

```
\begin{pspicture}(5,5)
\psset{unit=1.5}
```

```
\psline(1,1)(3,1)(2,2.732)(1,1)
\psarc[linewidth=0.1pt](1,1){2}{0}{70}
\psarc[linewidth=0.1pt](3,1){2}{110}{180}
\end{pspicture}
```

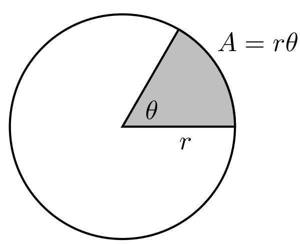


- The command `\psset{unit=1.5}` changes units from 1 cm to 1.5 cm.
- The command `\psarc(1,1){2}{0}{70}` draws an arc of a circle of radius 2 centered at the point $(1,1)$, going from the reference angle $0°$ to the reference angle $70°$. Setting the parameter `linewidth` to `0.1pt` defines the line width to be 0.1 points (72 points equals one inch), making the thin construction lines.

An important feature of PSTricks is that it allows the use of math mode within a picture.

Example 10.3. We draw a picture of a circle with a shaded sector, and some symbols and an equation in math mode.

```
\begin{pspicture}(4,4)
\pscircle(2,2){1.5}
\pswedge[fillstyle=solid,fillcolor=lightgray](2,2){1.5}{0}{60}
\put(2.75,1.7){$r$}
\put(2.3,2.1){$\theta$}
\put(3.25,3){$A=r\theta$}
\end{pspicture}
```

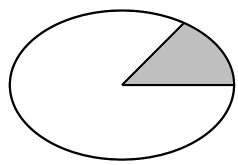


- The `\put` commands are from LaTeX's `picture` environment.
- The command `\pswedge(2,2){1.5}{0}{60}` creates a wedge (a sector of a circle), centered at $(2,2)$, with radius 1.5, going from reference angle $0°$ to reference angle $60°$. The parameters cause the wedge to be filled light gray.

If we want to draw a shaded sector of an ellipse, then we need to use a different method, because `\pswedge` only works for circles. One way to achieve the effect we want is to use a clipping path, which restricts further graphics to a specified region. Think of a clipping path as laying a window on the picture. After we have defined a clipping path, we can draw anywhere in the picture, but only the parts in the window will show.

Example 10.4. We draw an ellipse with a shaded sector.

```
\begin{pspicture}(4,4)
\psclip{\psellipse(2,2)(1.5,1)}
\psline[fillstyle=solid,fillcolor=lightgray](2,2)(4,2)(4,4)(2,2)
\endpsclip
\psellipse(2,2)(1.5,1)
\end{pspicture}
```



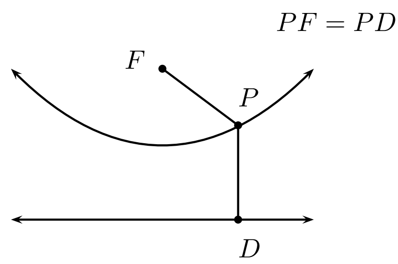
- The command `\psclip` sets up a clipping path defined by what is drawn in the braces that follow (the ellipse).
- The command `\psellipse(2,2)(1.5,1)` draws an ellipse with center $(2,2)$, horizontal radius 1.5, and vertical radius 1.
- The command `\psline` draws a filled gray triangle, but only the part within the ellipse shows (because of the clipping path).
- The command `\endpsclip` removes the clipping path, so subsequent commands can draw anywhere.

We need to draw the ellipse a second time in order to ensure that its black line boundary is not wiped out by the gray sector.

Example 10.5. We create a picture of a parabola together with its focus and directrix. We add some labels and an equation using math mode.

```
\begin{pspicture}(-2,-2)(2,2)
\parabola{<->}(-2,1)(0,0)
\psline{<->}(-2,-1)(2,-1)
\put(0,1){\circle*{0.1}}
\put(1,0.25){\circle*{0.1}}
\put(1,-1){\circle*{0.1}}
\psline(0,1)(1,0.25)
\psline(1,0.25)(1,-1)
\put(-0.5,1){$F$}
\put(1,0.5){$P$}
```

```
\put(1,-1.5){$D$}
\put(1.5,1.5){$PF=PD$}
\end{pspicture}
```

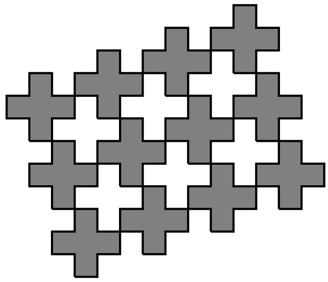


- The command `\begin{pspicture}(-2,-2)(2,2)` starts the picture and sets aside space in a coordinate system with lower-left corner $(-2,-2)$ and upper-right corner $(2,2)$.
- The command `\parabola(-2,1)(0,0)` draws a parabola that starts at $(-2,1)$ and has its extreme value (a minimum) at $(0,0)$.
- Including the parameter `{<->}` in the `\parabola` and `\psline` commands puts arrows at both ends of these curves.
- The `\circle*` commands draw filled circles that represent points in the picture.

It's sometimes handy to define a drawing procedure and use it several times. We do this with a `\def` command.

Example 10.6. We make a tessellation pattern of crosses.

```
\begin{pspicture}(5,4)
\psset{unit=0.3}
\def\cross
{\psline[fillstyle=solid,fillcolor=gray]%
  (0,0)(1,0)(1,1)(2,1)(2,2)(1,2)(1,3)(0,3)%
  (0,2)(-1,2)(-1,1)(0,1)(0,0)}
\put(3,1){\cross}
\put(6,2){\cross}
\put(9,3){\cross}
\put(12,4){\cross}
\put(2,4){\cross}
\put(5,5){\cross}
\put(8,6){\cross}
\put(11,7){\cross}
\put(1,7){\cross}
\put(4,8){\cross}
\put(7,9){\cross}
\put(10,10){\cross}
\end{pspicture}
```



- The `\def` command defines the procedure `\cross`. Everything in the immediately following braces will run each time we use the `\cross` command.

- The percentage sign (%) at the end of a line tells PSTricks that more of the declaration of a command is coming on the next line.

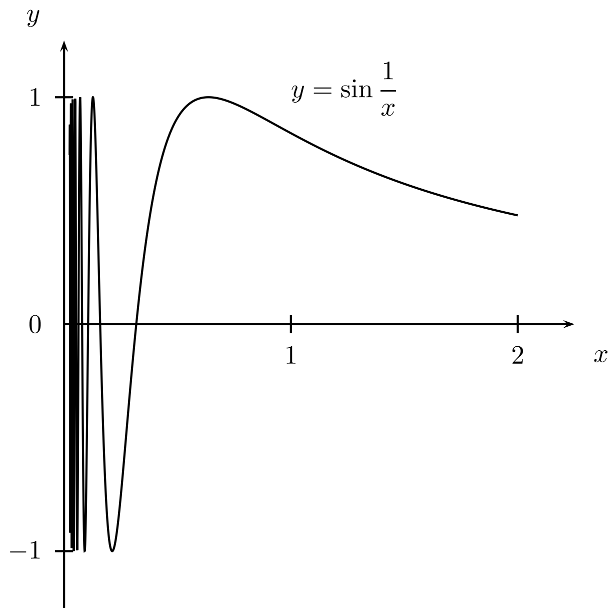
Note. You may wonder whether the process of drawing elements repeatedly (e.g., the crosses in the previous example) can be done in a loop. Indeed, loops are possible in PSTricks, but in our experience, figures requiring programming constructs (loops, arrays, etc.) are created more easily using PostScript (Chapter 17).

10.3 How to plot functions

To plot a function using PSTricks, we need the `pst-plot` package. The examples in this section assume that we have included a `\usepackage{pst-plot}` command in the preamble of our document.

Example 10.7. We graph the wildly oscillating function $y = \sin(1/x)$.

```
\begin{pspicture}(-0.25,-4.25)(7.5,4.25)
\psset{xunit=3cm,yunit=3cm}
\psaxes{->}(0,0)(0,-1.25)(2.25,1.25)
\psplot[plotpoints=2500]{0.025}{2}{1 x div RadtoDeg sin}
\put(7,-0.5){$x$}
\put(-0.5,4){$y$}
\put(3,3){$y=\sin\dfrac{1}{x}$}
\end{pspicture}
```



- The command `\psset{xunit=3cm,yunit=3cm}` defines units for the x and y axes.
- The `\psaxes` command creates the coordinate axes. The numbers

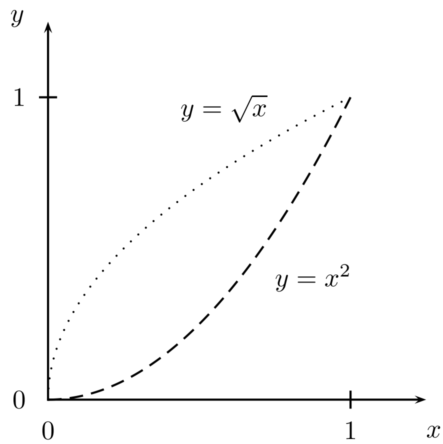
  ```
  (0,0)(0,-1.25)(2.25,1.25)
  ```

  determine that the coordinate axes intersect at $(0, 0)$, the lower-left corner of the coordinate system is $(0, -1.25)$, and the upper-right corner is $(2.25, 1.25)$.
- The command `\psplot` graphs the function. Setting `plotpoints=2500` specifies the number of points in the graph. The variable x goes from 0.025 to 2. (We can't let $x = 0$ because of the nature of the function.)
- Functions are defined in postfix notation, i.e., with arithmetic operators coming after the numbers they operate on rather than between them (see Chapter 17). The function definition `1 x div RadtoDeg sin` first computes $1/x$ (notice the `div` comes after the 1 and the x), then converts this real number to a degree measure, and finally computes the sine of the result.

Example 10.8. You can plot several curves together. We show a square root function and a squaring function. The square root function is depicted as a dotted curve and the square function as a dashed curve.

```
\begin{pspicture}(-0.5,-0.5)(5,5.5)
\psset{xunit=4cm,yunit=4cm}
\psaxes{->}(0,0)(1.25,1.25)
\psset{plotpoints=500}
\psplot[linestyle=dotted]{0}{1}{x sqrt}
\psplot[linestyle=dashed]{0}{1}{x 2 exp}
\put(5,-0.5){$x$}
\put(-0.5,5){$y$}
```

```
\put(1.75,3.75){$y=\sqrt{x}$}
\put(3,1.5){$y=x^2$}
\end{pspicture}
```

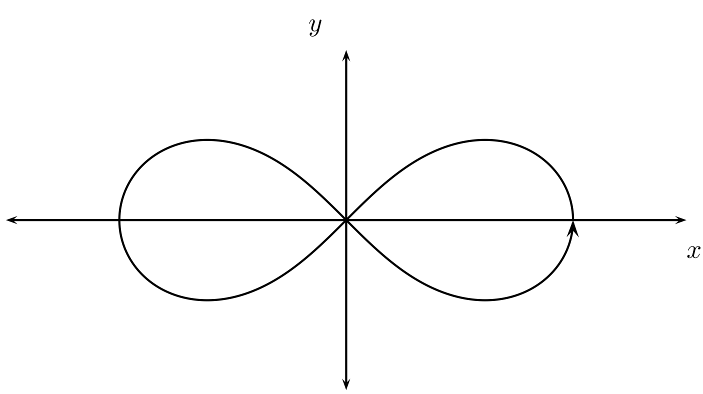


- The `\psset` command specifies the number of points plotted in both functions. Notice that we don't need as many points as in the previous example because these functions don't oscillate.

- In this example and the previous one, we use x as the independent variable. The `psplot` command expects x, so we can't use any variable that we dream up. In the next example, we do a parametric plot, so the variable is t.

Example 10.9. We make a parametric plot of a lemniscate.

```
\begin{pspicture}(-2.5,-2.5)(2.5,2.5)
\psset{xunit=3cm,yunit=3cm}
\psaxes[ticks=none,labels=none]{<->}(0,0)(-1.5,-0.75)(1.5,0.75)
\parametricplot[plotpoints=500,arrows=->,arrowscale=1.5]{0}{360}
{t cos 1 t sin 2 exp add div
t sin t cos mul 1 t sin 2 exp add div}
\put(4.5,-0.5){$x$}
\put(-0.5,2.5){$y$}
\end{pspicture}
```



The parametric equations for the lemniscate are

$$x = \frac{\cos t}{1+\sin^2 t}, \quad y = \frac{\sin t \cos t}{1+\sin^2 t}, \quad 0^\circ \le t < 360^\circ.$$

- In the `\psaxes` command, setting `ticks=none` and `labels=none` suppresses the ticks and numbers on the axes.

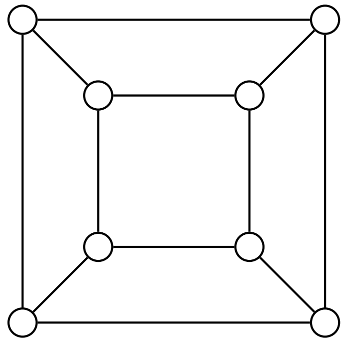
- The command `\parametricplot` sets up a parametric plot. Its parameters tell the number of points to plot, with t going from 0° to 360° and put a little arrow in the positive t direction along the curve. The x and y functions are defined in postfix notation.

10.4 How to make pictures with nodes

To make diagrams with nodes and connections, we need the `pst-node` package, so to do the examples in this section we would put a `\usepackage{pst-node}` command in the preamble.

Example 10.10. We make the cube graph, consisting of eight vertices and twelve edges.

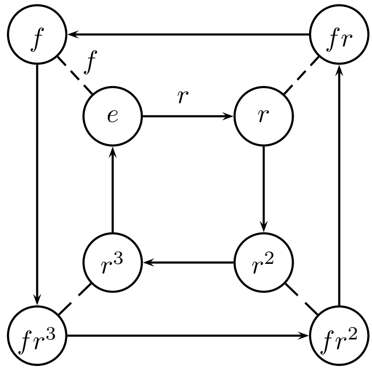
```
\begin{pspicture}(-3,-3)(3,3)
\psset{radius=0.2}
% draw inner square of vertices and edges
\put(-1,1){\Circlenode{1}}
\put(1,1){\Circlenode{2}}
\put(1,-1){\Circlenode{3}}
\put(-1,-1){\Circlenode{4}}
\ncline{1}{2}
\ncline{2}{3}
\ncline{3}{4}
\ncline{4}{1}
% draw outer square of vertices and edges
\put(-2,2){\Circlenode{5}}
\put(2,2){\Circlenode{6}}
\put(2,-2){\Circlenode{7}}
\put(-2,-2){\Circlenode{8}}
\ncline{5}{6}
\ncline{6}{7}
\ncline{7}{8}
\ncline{8}{5}
% draw edges between inner and outer square
\ncline{1}{5}
\ncline{2}{6}
\ncline{3}{7}
\ncline{4}{8}
\end{pspicture}
```



- The command `\psset{radius=0.25}` sets the radii of all nodes to 0.25.
- The command `\Circlenode{1}` defines a node in the shape of a circle, with the name "1".
- The command `\ncline{1}{2}` (which stands for 'node connection line') draws a line between nodes "1" and "2".

Example 10.11. We expand on the cube graph of Example 10.10 to make a diagram of a Cayley graph of the dihedral group D_4 (of order eight).

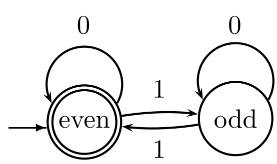
```
\begin{pspicture}(-3,-3)(3,3)
\psset{radius=0.4}
\put(-1,1){\Circlenode{E}{$e$}}
\put(1,1){\Circlenode{R}{$r$}}
\put(1,-1){\Circlenode{R2}{$r^2$}}
\put(-1,-1){\Circlenode{R3}{$r^3$}}
\ncline{->}{E}{R}
\ncline{->}{R}{R2}
\ncline{->}{R2}{R3}
\ncline{->}{R3}{E}
\put(-2,2){\Circlenode{F}{$f$}}
\put(2,2){\Circlenode{FR}{$fr$}}
\put(2,-2){\Circlenode{FR2}{$fr^2$}}
\put(-2,-2){\Circlenode{FR3}{$fr^3$}}
\ncline{->}{F}{FR3}
\ncline{->}{FR3}{FR2}
\ncline{->}{FR2}{FR}
\ncline{->}{FR}{F}
\ncline[linestyle=dashed]{E}{F}
\ncline[linestyle=dashed]{R}{FR}
\ncline[linestyle=dashed]{R2}{FR2}
\ncline[linestyle=dashed]{R3}{FR3}
\put(0.25,1.25){$r$}
\put(-1,1.7){$f$}
\end{pspicture}
```



- The parameter `{->}` in an `\ncline` command puts an arrowhead on the end of the line.

Example 10.12. We make a picture of a finite state automaton that accepts binary strings with an even number of 1s.

```
\begin{pspicture}(6,3)
\psset{radius=0.5}
\put(2,1){\Circlenode[doubleline=true]{EVEN}{even}}
\put(4,1){\Circlenode{ODD}{odd}}
\ncarc{->}{EVEN}{ODD}
\ncarc{->}{ODD}{EVEN}
\nccircle{->}{EVEN}{0.5}
\nccircle{->}{ODD}{0.5}
\put(1.5,1){\vector(1,0){0.5}}
\put(2.4,2.25){$0$}
\put(4.4,2.25){$0$}
\put(3.4,1.4){$1$}
\put(3.4,0.6){$1$}
\end{pspicture}
```



- Setting the parameter `doubleline=true` creates a double circle node.
- The command `\ncarc` draws an arc.

10.5 How to learn more

PSTricks supports the use of color. The following example depicts the five tetrominoes packed into a 3×7 box with one empty square. See Figure 1 of the color insert.

```
\begin{pspicture}(8,4)
\psline[fillstyle=solid,fillcolor=red]
(1,1)(3,1)(3,3)(1,3)(1,1)
\psline[fillstyle=solid,fillcolor=yellow]
(1,3)(5,3)(5,4)(1,4)(1,3)
\psline[fillstyle=solid,fillcolor=green]
(3,1)(5,1)(5,2)(6,2)(6,3)(4,3)(4,2)(3,2)(3,1)
\psline[fillstyle=solid,fillcolor=blue]
(5,1)(8,1)(8,2)(7,2)(7,3)(6,3)(6,2)(5,2)(5,1)
\psline[fillstyle=solid,fillcolor=lightgray]
(5,4)(8,4)(8,2)(7,2)(7,3)(5,3)(5,4)
\end{pspicture}
```

When the five tetrominoes are put in a 3×7 box, can you figure out why the empty square can't be the center square?

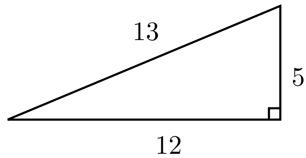
A good source for lots of examples, in black-and-white and color, is the PSTricks Web site at:

www.tug.org/PSTricks/main.cgi

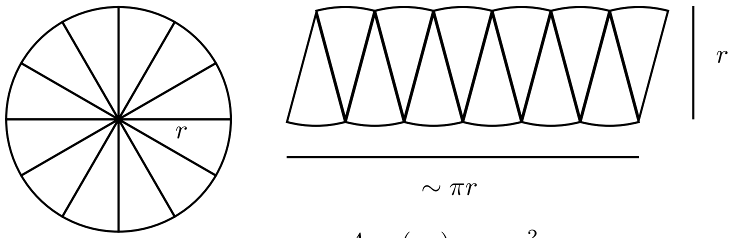
The standard reference for LaTeX graphics is [19].

Exercises

1. Make a picture of a 5–12–13 Pythagorean triangle, as below.

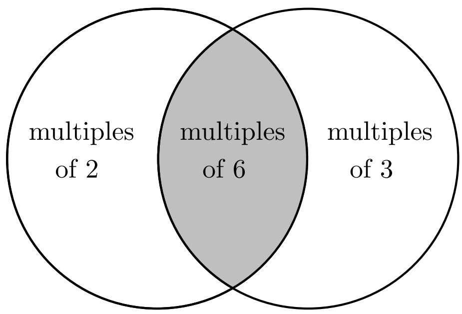


2. Make a diagram like the one that follows, illustrating Archimedes' demonstration of the formula for the area of a circle.



$$A = (\pi r)r = \pi r^2$$

3. Make a picture of the Venn diagram below.



4. Make a picture of an ellipse and its two foci F_1 and F_2, illustrating the relation $F_1P + F_2P = \text{constant}$, where P is a point on the ellipse.

5. Write each of these postfix expressions in standard form:

```
x 1 add 2 exp

x 1 add 2 exp 1 x sub div

x x sin mul

2 x sin x cos mul mul
```

6. Plot the function
$$f(x) = \begin{cases} x^2, & 0 \le x \le 2 \\ -x^2, & -2 \le x < 0. \end{cases}$$

7. Plot $y = \sin x$ and $y = \cos x$ on the same coordinate system, for $0 \le x \le 2\pi$. Show the sine function as a solid curve and the cosine function as a dotted curve.

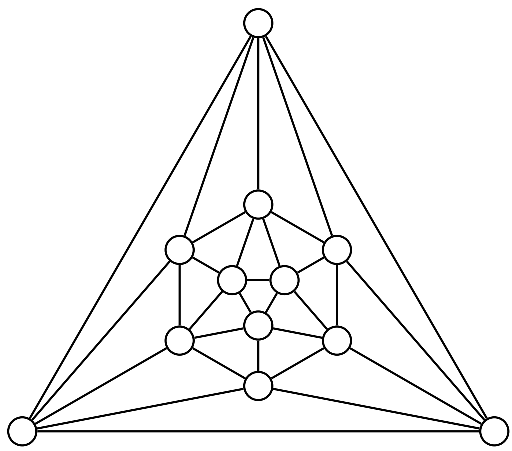
8. Plot $y = \sqrt{x}\sin(1/x)$, for $0 < x \le 2$. On the same coordinate system, plot the functions $y = \sqrt{x}$ and $y = -\sqrt{x}$, for $0 \le x \le 2$, with these functions shown as dotted curves.

9. Plot the cardioid given by the parametric equations
$$x = \cos t(1 - \cos t)$$
$$y = \sin t(1 - \cos t), \quad 0 \le t \le 2\pi.$$

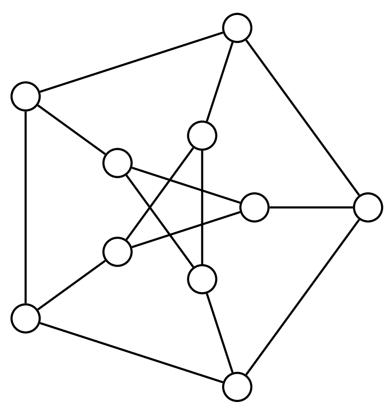
10. Draw a graph consisting of two sets of three nodes and all nine possible line connections between the two sets.

11. Draw a graph consisting of fives nodes and all possible line connections between them.

12. Make a diagram of the icosahedral graph below.



13. Make a diagram of the Petersen graph, with ten vertices and fifteen edges, as below. Color the vertices with three colors so that no connected vertices are the same color.

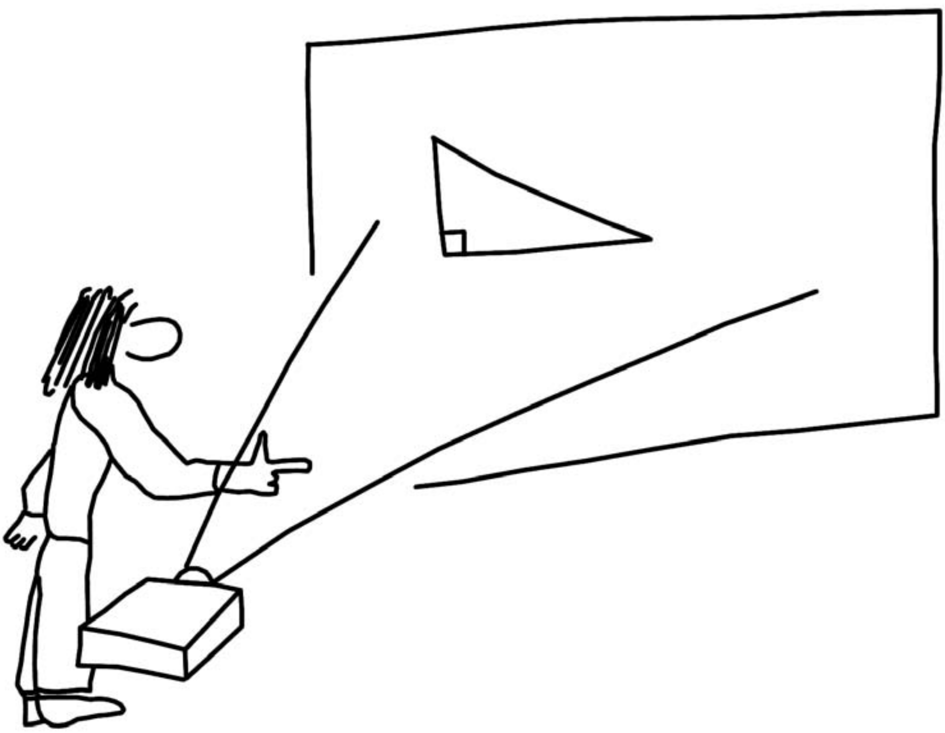


14. Make a diagram of the Cayley graph of the eight-element group $\mathbf{Z}_4 \times \mathbf{Z}_2$, with generators $(1, 0)$ and $(0, 1)$.

15. Draw a finite state automaton that accepts binary strings that do not contain two consecutive 0's. You can label the states "good string not ending in 0," "good string ending in 0," and "bad string."

Chapter 11

Getting Started with Beamer



In this chapter, we take a look at one of the most popular ways to give a math talk, using Beamer.

11.1 What is Beamer?

So, you've been asked to make a presentation on a mathematical topic. How do you project mathematical symbols and formulas for your audience to see? Beamer was designed by Till Tantau for just this purpose. Beamer is a LaTeX document class that enables you to project a slide show composed in the LaTeX mathematical typesetting system. In this chapter, we'll assume that you know the basics of LaTeX and have some material composed in LaTeX (see Chapter 9). We'll cover the essentials of Beamer.

11.2 How to think in terms of frames

Starting with your LaTeX document, Beamer produces a PDF document that you can project onto a screen. In terms of ease of use and flexibility, Beamer has much in common with Microsoft's PowerPoint. However, with Beamer you can project work composed in LaTeX. This is very convenient for math students (and professional mathematicians), because mathematical formulas and symbols typically play a large role in math presentations.

The basic unit of Beamer is a frame, which generates the contents of a single slide. (We'll learn about frames in the next section.) You will need to think about breaking up your work into frames. If you've already written a paper for an assignment, then you will want to distill the main points of your work into easily digested units to make up the frames. A little word of advice on this point: Concise is usually better than verbose. You should make each frame easy to comprehend for the slow readers in your audience. You can (and will) say more than what is written in a frame, so you have plenty of opportunities for verbal embellishment.

As you prepare your talk, keep in mind the "secrets" of good slide presentations.

The Seven Secrets of Superb Slide Shows:

- Put a small amount of material on each slide.
- Keep slides simple (no distractions).
- Keep slides consistent in style.
- Use pictures where appropriate.
- Use numbered or itemized lists where appropriate.
- Do examples.
- Show references where people can look up more information about your subject.

11.3 How to set up a Beamer document

Here are the basic elements of a beamer document. If you want to use this skeleton as a template for your work, go right ahead. You can simply change the material to make it suit your needs.

Notice that we use `beamer` (rather than `article`) as the document class in the preamble of the LaTeX document. Remember that you can use LaTeX commands for mathematical typesetting within your document. We don't do much of that in this example, but we do put `$` signs around numbers.

A Sample Beamer Document

```
\documentclass{beamer}

\title{My Wonderful Topic}
\author{Myself}
\institute{My University}

\begin{document}

\begin{frame}
\titlepage
\end{frame}

\begin{frame}
\begin{abstract}
We will discuss some properties of the number $12$.
\end{abstract}
\end{frame}

\begin{frame}
\frametitle{Facts about $12$}

\begin{itemize}
\item $12$ has six positive divisors.
\item $12$ is twice a perfect number ($6$).
\item $12$ is one of the legs of a famous
      Pythagorean triangle ($5$--$12$--$13$).
\end{itemize}
\end{frame}

\begin{frame}
\frametitle{Open Questions}

\begin{itemize}
\item How many positive integers less than $100$
      have six positive divisors?
\item Is twice a perfect number always an abundant number?
\item Are there other Pythagorean triangles with $12$
      as a side length?
\end{itemize}
\end{frame}

\end{document}
```

As you can see, the basic ingredient of a beamer document is the frame. Frames generate what you and your audience see on the screen. Similar to other LaTeX constructions (such as theorems and lists), frames are made with a `\begin{frame}...\end{frame}` construction. Our first frame (with the `\titlepage` command) produces the title page. The second frame contains an abstract for the talk. The other frames have titles given by the `\frametitle` command.

Note. After you create each new frame, rebuild your LaTeX document to check that things are working the way that you want. This will help you pinpoint the locations of errors (which we all inevitably make).

You can use ordinary LaTeX commands in the frames of your presentation. In the example above, we've used math mode for the numbers. In practice, the math can be quite complex (anything that LaTeX can handle). But remember the general rule to keep things as simple as possible for your audience.

This is a good time to reiterate some advice about giving talks (see Chapter 4). It's often a good idea to start a talk with an example. This can be a sample calculation or an easy-to-understand example of a phenomenon that you wish to describe. Too many talks start off with a laundry list of definitions and theorems, leaving the audience members in need of life preservers. In a math talk, an example is always welcome.

11.4 How to enhance a Beamer presentation

Would a picture enhance your math talk? If you are giving examples of the Pythagorean Theorem, you would want to show a picture of a right triangle. You can include images in your Beamer presentation the same way you do in other LaTeX documents. Simply put a `\usepackage{graphicx}` command in the preamble of your document, and use an `\includegraphics` command in the frame where you want the image to appear.

```
\begin{frame}
\frametitle{A famous Pythagorean triangle}

Here is a picture of the $5$--$12$--$13$ Pythagorean triangle.

\begin{center}
\includegraphics[width=3in,height=3in]{5-12-13triangle.eps}
\end{center}
\end{frame}
```

The LaTeX `picture` environment also works in Beamer documents, and it may be suitable if your pictures are simple.

Or you can use Beamer together with PSTricks (see Chapter 10). You will need a `\usepackage{pstricks}` command in the preamble of your document. Use the construction

```
\begin{frame}{PSTricks}
\end{frame}
```

for frames that contain PSTricks pictures.

You can do much more with Beamer, such as show the elements of a list one point at a time. To do this, put a `\pause` command in an item in an enumerate environment.

```
\begin{enumerate}
\item We will pause to discuss this item.  \pause
\item We will pause to discuss this item too.  \pause
\item Now we will discuss the final item.
\end{enumerate}
```

Now that you know the basics of Beamer, we encourage you to explore. For example, you can learn how to do each of the following:

- Enhance the look of your slides with "Beamer blocks."
- Jump forward and backward in your presentation, and link to Web documents.
- Generate a Table of Contents for your talk.

11.5 How to learn more

You can learn more about Beamer at:

`ctan.org/tex-archive/macros/latex/contrib/beamer/doc/beameruserguide.pdf`

There are many other good guides to Beamer on the World Wide Web. You can search for guides by entering "Beamer tutorial" in a Web browser. The LaTeX Beamer Class Homepage is located at:

`latex-beamer.sourceforge.net`

You may want to create a simple Beamer presentation, perhaps using the examples in this chapter, and then add to your presentation by following some of the more detailed advice that you can find on the Web.

Exercises

Use Beamer to give presentations on the following topics.

1. Give a presentation on the Pythagorean Theorem. Be sure to include diagrams of right triangles.
2. Give a presentation on Pascal's triangle.
3. Give a presentation on the Fundamental Theorem of Calculus. Include diagrams.
4. Give a presentation on Linear Programming optimization methods. You may want to include a discussion of the Simplex Algorithm.
5. Give a presentation on Numerical Analysis used in solving equations.
6. Give a presentation on some theorems of Graph Theory. Include diagrams.
7. Give a presentation on fast Fourier transforms.
8. Give a presentation on a mathematician chosen from [5] or [6]. Give biographical background on the person and describe some of the mathematics that the person created.
9. Give a presentation on a topic in mathematics and art, chosen from [28], [13], or [41].