Chapter 2

Typesetting

2.1 Getting started with LaTeX

Computer scientists nowadays write papers in LaTeX (also written LaTeX) due to its high quality, but it takes time to become familiar with this software. This chapter gives a simple introduction to the most basic and essential features.

For more complicated features, we search the Internet for examples and/or documentation and follow or edit those to suit. The website `tex.stackexchange.com` is an online question and answer site on LaTeX (and beyond); it's effectively StackOverflow for LaTeX. It is a good place to ask questions for how to achieve something, and to find example code to modify to suit your needs.

In LaTeX, the user creates a text file, such as the following

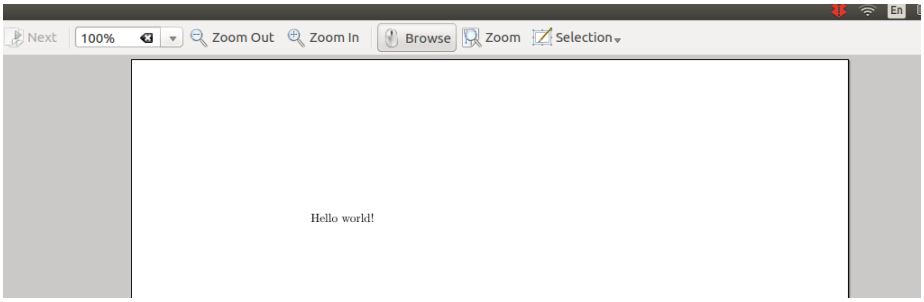
```
\documentclass{article}

\begin{document}

Hello world!

\end{document}
```

which is then compiled to give a high-quality rendering (usually in .pdf format). If we open the output in a PDF viewer, it looks like:



We can build from here in many ways.

Control commands We edit the LaTeX code using control commands:

- Section structure: we need `\section{...}` and `\subsection{...}`, and perhaps also `\subsubsection{...}`, and `\paragraph{...}`.
- Referencing: `\label{...}`, `\ref{...}`, and `\cite{...}`.
- Mathematical expressions: we write inline mathematical expressions in dollar signs, such as `$f(x)=mx+c$` and `$1,2,\ldots,100$`, and using mathematical control commands such as `\ldots`, `\pi`, and `\simeq` (see Section 2.2.2).

  There is an enormous number of mathematical symbols; Pakin [1] maintains a list (338 pages at the time of writing).

Packages Packages can be used to extend LaTeX's functionality. One example is the `color` package, which we call via `\usepackage{color}`, gives color functionality `\color{...}` and `\textcolor{...}{...}`.

Packages are generally available through the Comprehensive TeX Archive Network `https://www.ctan.org/pkg/`. Commonly used packages are available with most LaTeX software, in which case, we don't need to download them.

As a more complicated example, the LaTeX code

```
\documentclass{article}

\usepackage{amsmath}
\usepackage{color}

\begin{document}

\section{Introduction}

Hello world! I can count to one hundred:
$1,2,\ldots,100$.

In Section \ref{sec:details} we give
equation \eqref{eq:pi_approx}.

\subsection{Details}\label{sec:details}

Here's an equation:
\begin{equation}\label{eq:pi_approx}
\pi \simeq 3.14159
\end{equation}

\subsection{Colors}

\color{red} Here's \color{blue} some
\color{green} green.

\end{document}
```

compiles to the following:

1 Introduction

Hello world! I can count to one hundred: $1, 2, \ldots, 100$.
In Section 1.1 we give equation (1).

1.1 Details

Here's an equation:

$$\pi \simeq 3.14159 \tag{1}$$

1.2 Colors

Here's some green.

From here, we keep building and building.

2.1.1 LaTeX software

We edit the LaTeX file in a text editor; many text editors recognize LaTeX syntax. The LaTeX compiler is separate to the text editor, but many text editors are capable of calling the LaTeX compiler. For most applications the LaTeX compiler `pdflatex` is fine.

I primarily use two methods for writing LaTeX files:

TeX Live TeX Live (`tug.org/texlive`) is a LaTeX system for Linux (but its website says Windows too). I install `texlive-full` through the Ubuntu repository using the command `sudo apt-get install texlive-full`. The full version includes a lot of packages (it's a big download!). I use Kile as the text editor.

Overleaf Overleaf (`overleaf.com`) is a LaTeX editor and compiler that can be used from a web browser. You edit the LaTeX file, and it is compiled by the Overleaf servers. This is helpful when you have multiple authors. It's also helpful as you don't need local software (other than a web browser).

When I used Windows:

MikTeX MikTeX (`miktex.org`) is a LaTeX system for Windows. I used to use it in conjunction with the editor TeXnicCenter (`texniccenter.org`).

There are *many* LaTeX software packages around, and there is a list of (La)TeX editors at Wikipedia: `https://en.wikipedia.org/wiki/Comparison_of_TeX_editors`. Which one is "best" is usually whichever one you're most familiar with.

2.1.2 LaTeX preamble

The main text is written within `\begin{document}` and `\end{document}`. Before `\begin{document}`, we write the preamble, where we give the document class, identify which packages to load, describe macros we want to use, and so on.

Document classes

In the examples in Section 2.1, we use `\documentclass{article}` which tells the compiler that we're writing an "article". There are other document classes to choose from. Only a few are especially important:

Article `\documentclass{article}` is the the default way of writing a paper. We can change some options such as by `\documentclass[12pt]{article}` which changes the font size or `\documentclass[twocolumns]{article}` which changes to two-column format.

Beamer `\documentclass{beamer}` is great for writing talk slides in .pdf format. It takes a bit of getting used to. (If you're giving a talk at a conference, I reocmmend changing from the default beamer template—it's overused!)

Standalone `\documentclass{standalone}` is great for making high-quality .pdf figures. Sometimes a complicated figure takes a long time to compile, so it helps to compile figures separately when preparing a long document. A useful option is

```
\documentclass[crop]{standalone}
```

which gets rid of the surrounding white space, or

```
\documentclass[crop,margin=1mm]{standalone}
```

which only has a little white space.

Once the figure is compiled, we can use e.g.

```
\includegraphics[width=0.8\textwidth]{myfigure.pdf}
```

to include the figure in the main text; we include e.g. `\usepackage{graphicx}` to enable this function.

Journals and conferences Most journals and conferences will have a LaTeX template that they expect you to follow, and most include their own document class file, e.g. `\documentclass{svjour3}` is used by Springer. When submitting a paper for review, it's best to use the required document class.

To use e.g. `\documentclass{svjour3}`, the LaTeX compiler will need to find the class file. We usually achieve this by downloading the class file (svjour3.cls in this case) and its dependencies from the journal website, and storing them in a place the compiler can access (usually the same directory as the .tex LaTeX file).

Usepackage

LaTeX functionality is extended through the use of packages. These are called using `\usepackage{color}` for the `color` package, and so on.

Pseudocode `\usepackage{algorithm, algorithmicx, algpseudocode}` are used for displaying pseudocode.

Mathematics We can typeset mathematical expressions and statements in various ways using the packages:

```
\usepackage{amsmath, amsthm, amsfonts, amssymb}
```

and even more with `\usepackage{mathtools}`.

Figures and plots Tikz `\usepackage{tikz}` is used for drawing high-quality figures, and pgfplots `\usepackage{pgfplots}` (which is compatible with Tikz) is used for making high-quality plots. These packages are described in Section 2.4.

Graphics We include external images via `\includegraphics{...}` using packages such as `\usepackage{graphicx}` or `\usepackage{tikz}`.

There's many, many more packages! Other packages which I find useful are

- `\usepackage{url}` for well-typeset URLs,
- `\usepackage{cite}` to ensure citations are in order,
- `\usepackage{enumitem}` for greater functionality in enumerated lists,
- `\usepackage{tcolorbox}` to put boxes around snippets,
- `\usepackage{multirow}` for multi-row entries in tables, and
- `\usepackage{rotating}` for rotating text, which I find particularly useful in conjunction with `multirow` in tables.

Many packages clash with one another, and larger packages make compiling slower, so it's usually best not to include packages unnecessarily.

The LaTeX compiler needs to be able to find the package. The most useful packages are installed by default, but for others we need to download the .sty file somehow. This is usually achieved by (a) downloading it and putting the .sty file in the same directory as the .tex LaTeX file, or (b) installing it using some package management software. (Overleaf has most .sty files already functional.)

Titles

Ordinarily, a document has a title and an author. For this document, I used:

```
\title{Writing Computer Science Papers as a Chinese Postgrad}
```

for the title. After this, I typically use

```
\author{Rebecca J. Stones}
```

for my name. Or perhaps I might extend it to include my affiliation:

```
\author{Rebecca J. Stones\\
  College of Computer Science\\
  Nankai University, Tianjin, China}
```

Here `\\` creates a new line. We include the date with `\date{\today}` (where `\today` gives the current date), but I find it useful to write

```
\date{Last updated: \today}
```

The document contents is surrounded by

```
\begin{document}
```

and

```
\end{document}
```

We also need to add

```
\maketitle
```

between `\begin{document}` and `\end{document}` to tell the compiler to make a title. Ordinarily `\maketitle` occurs directly after `\begin{document}`.

Abstract After `\maketitle` we might want to include an abstract:

```
\begin{abstract}
This is my paper's abstract.
\end{abstract}
```

A simple example of what the preamble as a whole might look like is:

```
\documentclass{article}

\usepackage{amsmath, amsthm, amsfonts, amssymb}

\title{My First Computer Science Paper in English}
\author{Rebecca J. Stones\\
  College of Computer Science\\
  Nankai University, Tianjin, China}
\date{\today}

\begin{document}

\maketitle

\begin{abstract}
This is my paper's abstract.
\end{abstract}

\section{Introduction}

I like the equation $y=mx+c$.

\end{document}
```

This compiles to give:

My First Computer Science Paper in English

Rebecca J. Stones
College of Computer Science
Nankai University, Tianjin, China

October 4, 2018

Abstract

This is my paper's abstract.

1 Introduction

I like the equation $y = mx + c$.

2.2 Main text

2.2.1 Sections

LaTeX has various levels of sections; the most important for writing computer science papers are:

1. *Sections*: `\section{Section name}` which is usually used for "Introduction", "Background", "Experimental Results", "Conclusions", and so on.

2. *Subsections*: `\subsection{Subsection name}` which can be used to break apart sections.

3. *Sub-subsections*: `\subsubsection{Sub-subsection name}` which break up subsections.

4. *Paragraph*: `\paragraph{Paragraph name}` which break up sub-subsections.

There's also `\chapter{Chapter name}` if you're typesetting a book, and also `\subparagraph{Subparagraph name}`. Figure 2.1 gives a toy example of how to use sections.

2.2.2 Mathematics

One of the most important features in LaTeX is typesetting beautiful mathematical expressions in a fairly simple way. There are way too many features to list here: you need to learn to search the Internet for how to typeset formulas.

Many mathematics features are enabled using the `amsmath` package (via the command `\usepackage{amsmath}`). Here, AMS stands for American Mathematical Society.

I usually have `\usepackage{amsmath, amsthm, amsfonts, amssymb}` in the preamble, which also includes support for a range of theorem styles and symbols, and forget it's there. I likewise use the `mathtools` package (via `\usepackage{mathtools}`).

```
\documentclass{article}

\begin{document}

\section{Introduction}\label{se:intro}

This is where we introduce the topic.

\subsection{Ideas}

Here we discuss ideas.

\subsubsection{My first idea}

Good idea!  Also see Section \ref{se:other}.

\paragraph{More detail about some aspect.}

It's really great!

\paragraph{More detail about another aspect.}

Trust me.

\subsubsection{Another idea}

Now I have another idea.

\paragraph{Details.}

I don't want to go into details.

\subsection{Not my idea}\label{se:other}

I should discuss other people's ideas too.

\section{Background}

See Section \ref{se:intro}.

\end{document}
```

1 Introduction

This is where we introduce the topic.

1.1 Ideas

Here we discuss ideas.

1.1.1 My first idea

Good idea! Also see Section 1.2.

More detail about some aspect. It's really great!

More detail about another aspect. Trust me.

1.1.2 Another idea

Now I have another idea.

Details. I don't want to go into details.

1.2 Not my idea

I should discuss other people's ideas too.

2 Background

See Section 1.

Figure 2.1: A side-by-side toy example of using various section levels in LaTeX.

Inline equations Examples:

```
Some examples of inline mathematical equations are
$\sin^2(x)+\cos^2(x)=1$
and $\frac{-b \pm \sqrt{b^2-4ac}}{2a}$
and $2^x = 2^{3^{4^{5^{6^7}}}}$
and $\sum_{k=1}^{10} k^2=385$
and $x \xrightarrow{\text{square}} x^2$.
```

which compiles to:

> Some examples of inline mathematical equations are $\sin^2(x)+\cos^2(x) = 1$ and $\frac{-b\pm\sqrt{b^2-4ac}}{2a}$ and $2^x = 2^{3^{4^{5^{6^7}}}}$ and $\sum_{k=1}^{10} k^2 = 385$ and $x \xrightarrow{\text{square}} x^2$.

Inline equations are considered part of the sentence they belong to.

Displayed equations Examples:

```
One example of a displayed equation is
\[\int_0^\infty \frac{1}{x} dx \neq \frac{2^3}{\ln(n)},\]
another is
\[\sum_{\substack{\text{prime } p \\ p \neq 2}} \frac{1}{p},\]
and one more is
\[\left( \left( \frac{\mu+1}{x+2} \right)^2 \right)^{\log n}.\]
```

which compiles to:

One example of a displayed equation is

$$\int_0^\infty \frac{1}{x} dx \neq \frac{2^3}{\ln(n)},$$

another is

$$\sum_{\substack{\text{prime } p \\ p \neq 2}} \frac{1}{p},$$

and one more is

$$\left(\left(\frac{\mu+1}{x+2} \right)^2 \right)^{\log n}.$$

Some people use `$$ ... $$` instead of `\[ ... \]` for displayed equations. The difference between is not too important (see `tex.stackexchange.com/q/503` for the difference).

Displayed equations are also part of a sentence, so we need to continue using the appropriate punctuation (“.”, “,”, etc.).

Unlike inline equations, we are able to number displayed equations so that they can be later referenced. For example the LaTeX code

```
The equation
\begin{equation}\label{eq:toy_eq}
a^{b^c}=a^{b^c}=b^{a^c}=b^{c^a}=c^{a^b}=c^{b^a}
\end{equation}
is equation \ref{eq:toy_eq},
but we refer to it as \eqref{eq:toy_eq}.
```

which compiles to:

The equation

$$a^{b^c} = a^{b^c} = b^{a^c} = b^{c^a} = c^{a^b} = c^{b^a} \tag{2.1}$$

is equation 2.1, but we refer to it as (2.1).

It’s usual to only number equations when those equation numbers are used later on. We can also use `\begin{equation*}` and `\begin{equation*}` to omit numbers. We can also use `\nonumber` to suppress the equation number.

Multline One of the `mathtools` functions is the ability to write multi-line equations. Here’s one of my examples:

```
\begin{multline*}
\theta=
  \big((0\,1\,2\,3\,4\,5\,6\,7)(8\,9\,10\,11)(12\,13\,14\,15), \\
    (0\,1\,2\,3\,4\,5\,6\,7)(8\,9\,10\,11)(12\,13\,14\,15), \\
    (0\,1\,2\,3\,4\,5\,6\,7)(8\,9\,10\,11)(12\,13\,14\,15)\big)
\end{multline*}
```

which gives

$$\theta = \big((0\,1\,2\,3\,4\,5\,6\,7)(8\,9\,10\,11)(12\,13\,14\,15),\\ (0\,1\,2\,3\,4\,5\,6\,7)(8\,9\,10\,11)(12\,13\,14\,15),\\ (0\,1\,2\,3\,4\,5\,6\,7)(8\,9\,10\,11)(12\,13\,14\,15)\big)$$

which is too long for a single line.

Named mathematical functions and kerning Note that `$sin(x)$`, which compiles to $sin(x)$, is incorrect, since $sin(s) = s \times i \times n(x)$. Instead, we should write `$\sin(x)$`, which compiles $\sin(x)$. Likewise for `$\log$` and `$\ln$`. Consider the following examples:

- $diff$ (typeset `$diff$`), where the spacing in math mode is unsuitable;
- *diff* (typeset `\textit{diff}`), which is what "diff" looks like in italics;
- diff (typeset `$\mathrm{diff}$`), which is what we should use;
- diff (typeset `$\text{diff}$`);
- *diff* (typeset `$\mathrm{\it diff}$`); and
- *diff* (typeset `$\text{\it diff}$`).

They are quite different:

$$diff \quad \mathit{diff} \quad \mathrm{diff}$$

The spacing between letters, called *kerning*[1], is poor in $diff$, and the poor kerning makes it difficult to recognize that $diff$ is meant to be one word "diff", and not the product $d \times i \times f \times f$. It also just looks ugly.

Another example of poor kerning is VAR:

$$VAR \quad \mathit{VAR} \quad \mathrm{VAR}$$

Align We can align a list of equations nicely in LaTeX using the `align` and `align*` environments. For example:

```
We have three equations:
\begin{align}
x+yz & \geq n, \label{eq:ineq1} \\
y+xz & \geq n, \text{ and} \label{eq:ineq2} \\
z+xy & \geq n. \label{eq:ineq3}
\end{align}
See equations \eqref{eq:ineq1}, \eqref{eq:ineq2}, and \eqref{eq:ineq3}.
```

[1] https://en.wikipedia.org/wiki/Kerning

which compiles to:

We have three equations:

$$x + yz \geq n, \tag{2.2}$$
$$y + xz \geq n, \text{ and} \tag{2.3}$$
$$z + xy \geq n. \tag{2.4}$$

See equations (2.2), (2.3), and (2.4).

And if we use `align*`, we leave off the equation numbers:

```
\begin{align*}
x+yz &\geq n, \\
y+xz &\geq n, \\
z+xy &\geq n.
\end{align*}
```

which compiles to:

$$\begin{aligned}
x+yz &\geq n, \\
y+xz &\geq n, \\
z+xy &\geq n.
\end{aligned}$$

This means we cannot refer to these equations using `\ref{...}`.

It's usual to only number equations when those equation numbers are used. We use `\begin{align}` and `\end{align}` together with `\nonumber` if we only want some equations numbered.

2.2.3 Tables

The following example LaTeX code compiles to give Table 2.1. It's a modified table from one of my papers.

```
\begin{table}[htp]
\centering
\begin{tabular}{|l|l|}
\hline
$N_G(v)=\{u\in V: uv \in E\}$ & The open neighborhood of $v$. \\
$N_G[v]=N_G(v) \cup \{v\}$ & The closed neighborhood of $v$. \\
\hline
\end{tabular}
\caption{Table of notation.}\label{ta:notatExample}
\end{table}
```

$N_G(v) = \{u \in V : uv \in E\}$	The open neighborhood of v.
$N_G[v] = N_G(v) \cup \{v\}$	The closed neighborhood of v.

Table 2.1: Table of notation.

Breaking this toy example apart:

- We use a *table environment* between `\begin{table}` and `\end{table}`.
- The table parameters `[htp]` give preference to the placement of the table. In this case, the preference is "here" (`h`), "top of page" (`t`), "page of floats" (`p`). In my experience, this is almost always the best choice.
- The control command `\centering` centers everything within the table environment.
- We have a *tabular environment* between the commands `\begin{tabular}` and `\end{tabular}`.
- The tabular parameters are `{|l|l|l|}` where `|` indicates a horizontal line and `l` indicates left justified. Common alternatives are `c` for centered, `r` for right justified, and `p{2cm}` for paragraphs of with 2cm.
- The symbol `&` indicates the end of the contents in that cell.
- The symbol `\\` indicates the end of the contents of that row.
- The control command `\hline` adds a horizontal line.
- A caption `\caption{ ... }` is given which describes the contents of the table.
- A label `\label{ta:notatExample}` is given which is later referenced in the main text by `\ref{ta:notatExample}`. Here `ta:notatExample` is a name I choose for this table. For example,

  ```
  See Table \ref{ta:notatExample} for a table of notation.
  ```

  compiles to

  > See Table 2.1 for a table of notation.

  Importantly, as we edit the LaTeX code, we do not need to renumber this table every time we add in a new table; LaTeX handles the referencing automatically.

The following toy example LaTeX code uses a `\multicolumn` function:

```
\begin{tabular}{ccc}
1 & 2 & 3 \\
\multicolumn{3}{c}{123456} \\
4 & 5 & 6 \\
\end{tabular}
```

which compiles to:

1	2	3
123456		
4	5	6

In the above toy example, the tabular environment is not inside a table environment.

A toy example using `\usepackage{multirow}` and `\usepackage{rotating}` is:

```
\begin{tabular}{|c|r|l|p{1in}|}
\hline
a & b & c & I'm writing in LaTeX \\
\hline
a a a & b b b & c c c & Here's some more LaTeX \\
\hline
\multicolumn{4}{|c|}{I'm in the middle!} \\
\multicolumn{2}{|c}{L} & \multicolumn{2}{c|}{R} \\
\cline{1-2} \cline{4-4}
\multirow{3}{*}{\rotatebox{90}{hello}} & 1 & 2 & 3 \\
 & 4 & 5 & 6 \\
 & 7 & 8 & 9 \\
\hline
\end{tabular}
```

which compiles to

<table>
<tr><td>a</td><td>b</td><td>c</td><td>I'm writing in LaTeX</td></tr>
<tr><td>a a a</td><td>b b b</td><td>c c c</td><td>Here's some more LaTeX</td></tr>
<tr><td colspan="4">I'm in the middle!</td></tr>
<tr><td colspan="2">L</td><td colspan="2">R</td></tr>
<tr><td rowspan="3">hello</td><td>1</td><td>2</td><td>3</td></tr>
<tr><td>4</td><td>5</td><td>6</td></tr>
<tr><td>7</td><td>8</td><td>9</td></tr>
</table>

Here we also use `\cline` for partial horizontal lines.

High-quality tables

Often tables drawn using `\usepackage{booktabs}` look more professional. For example:

```
\begin{tabular}{crrr}
\toprule
& \multicolumn{3}{c}{equivalence} \\
\cmidrule(lr){2-4}
order & isomorphism & isotopism & paratopism \\
\midrule
10 & 0.01 & 0.17 &  0.98 \\
11 & 0.01 & 0.32 &  1.90 \\
12 & 0.02 & 0.58 &  3.60 \\
13 & 0.02 & 1.10 &  6.80 \\
14 & 0.03 & 1.90 & 11.30 \\
15 & 0.04 & 3.00 & 18.80 \\
\bottomrule
\end{tabular}
```

compiles to:

	equivalence		
order	isomorphism	isotopism	paratopism
10	0.01	0.17	0.98
11	0.01	0.32	1.90
12	0.02	0.58	3.60
13	0.02	1.10	6.80
14	0.03	1.90	11.30
15	0.04	3.00	18.80

The only difference is the lines: `\toprule`, `\cmidrule(lr){2-4}`, `\bottomrule`, and the lack of vertical lines.

2.2.4 Figures

The following is a toy example of LaTeX code which shows a figure:

```
\begin{figure}[htp]
\centering
\includegraphics[width=3in]{308px-Harddisk_1.jpg}
\caption{Fujitsu hard disk.  Sourced from Wikimedia Commons
\protect\url{https://commons.wikimedia.org/wiki/File:Harddisk_1.jpg}}
\label{fi:hard_disk}
\end{figure}

Figure \ref{fi:hard_disk} shows an example of a hard disk.
```

This compiles to the following:

Figure 2.2: Fujitsu hard disk. Sourced from Wikimedia Commons `https://commons.wikimedia.org/wiki/File:Harddisk_1.jpg`

Figure 2.2 shows an example of a hard disk.

We break this apart (it's similar to the table environment):

- We use a *figure environment* between `\begin{figure}` and `\end{figure}`.

- The table parameters `[htp]` give preference to the placement of the table.
- The control command `\centering` centers everything.
- We use the command `\includegraphics{...}` to include an image. The option `[width=3in]` says the image is resized to have a width of 3 inches. And the filename is `308px-Harddisk_1.jpg`.

  It's also useful to use `[width=\textwidth]`, or `[width=\columnwidth]` (which are not equal for e.g. two-column papers).

  We can also use options such as

  ```
  \includegraphics[trim={4cm 5cm 6cm 7cm},clip]{...}
  ```

  to trim the margins of the image, which is useful for not including surrounding whitespace.
- A caption `\caption{...}` is given describing the contents of the table.

  In this case, we use `\url{...}` in the caption, which we need to `\protect` to avoid a clash. (Clashes are a fairly common problem in LaTeX.)
- A label `\label{fi:hard_disk}` is given which is later referenced in the main text by `\ref{fi:hard_disk}` to give "Figure 2.2".

Since this is not my image, I acknowledge where it comes from ("Sourced from..."); it is important to do so to avoid plagiarism—it costs nothing to include the citation, and the author shows responsibility.

2.2.5 Other points

Macros It is possible to define macros such as

```
\def\Ncond{N^{\text{cond}}}
```

so that if we call it via

```
$\Ncond$
```

it replaces `\Ncond` with `N^{\text{cond}}` which looks like:

N^{cond}

Macros are useful for when you might later change notation; the macro can simply be edited, and everything changes accordingly. However, this is probably not a good idea if your co-authors need to edit your LaTeX file; they don't want to memorize a bunch of macro names (while they're probably familiar with the LaTeX ones).

Spacing after abbreviations LaTeX makes lots of automatic changes to spacing. Compare the following:

LaTeX	Output	
`Monsters Inc. is great`	Monsters Inc. is great	✘
`Monsters Inc.\ is great`	Monsters Inc. is great	✓
`Monsters Inc.~is great`	Monsters Inc. is great	✓
`Monsters Inc.~ is great`	Monsters Inc. is great	✘ ✘ ✘

In the first example, the period "." is compiled like a full stop at the end of a sentence (so LaTeX adds a longer space afterwards). The second and third examples tell LaTeX to only add a normal-sized space—they are both are fine.

Sometimes when I tell students about this problem, they write it as per the fourth way listed above, which is even worse than if they did nothing. Writing "`Monsters Inc.~is great`" includes a non-breaking space (i.e., the line will not end directly after "Inc.". Writing "`Monsters Inc.~ is great`" includes both a non-breaking space, and an additional space.

Spacing in equations LaTeX makes many space-related changes to equations, such as the following:

Output	LaTeX
$1\ -1$	`$1$ $-1$`
$1-1$	`$1-1$`

The first "−" represents "negative 1" (and thus uses a small space), whereas the second "−" represents "minus 1" (and thus uses a big space) and these receive different spacing. It's important to be aware that LaTeX makes all kinds of changes like this.

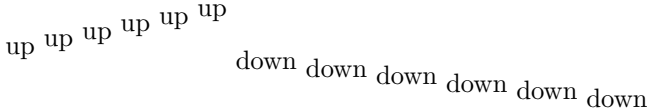
Here's a similar example:

$$
\begin{aligned}
A\cup \\
\cup B \\
A\cup \\
\cup B \\
A\cup B
\end{aligned}
$$

```
\begin{align*}
  A  \cup    \\
     \cup B  \\
  A  \cup {} \\
  {} \cup B  \\
  A  \cup B  \\
\end{align*}
```

In this case `\cup` behaves differently depending on the surrounding material: what comes before the `\cup` affects the spacing after the `\cup`. The `{}` is an empty input.

Moving things around We can move things about by e.g. `\raisebox{1in}{up}` which raises the text one inch. We can also lower it by `\raisebox{-1in}{down}`.



We can add positive and/or negative space such as with `\vspace{1in}` for vertical space or `\hspace{1in}` for horizontal space.

One sentenceAnother sentence. (`\hspace{-0.5cm}`)

One sentence. Another sentence. (`\hspace{0.5cm}`)

And there's also:

aaaaaaaa	\!
aaaa aaaa	\,
aaaa aaaa	\;
aaaa aaaa	\ (← there's a space here)
aaaa aaaa	\quad
aaaa aaaa	\qquad

However, it's usually a bad idea to change the spacing, so don't do it unnecessarily.

We also have text alignment:

\flushright

\centering

\flushleft

We also have \hfill, which I find is useful for quotes, such as:

> The secret of getting ahead is getting started.
>
> – Mark Twain

Here I use `\hfill -- Mark Twain`.

2.2.6 Algorithms and pseudocode

The following is a toy example of LaTeX code giving an algorithm in pseudocode:

```
\begin{algorithm}[htp]
    \caption{Matrix multiplication}
     \begin{algorithmic}[1]
        \Require Matrices $M_1,M_2,\ldots,M_n$
        \Ensure The matrix product $M_1 M_2 \cdots M_n$.
        \State $R \gets M_1$
        \For{$i$ from $2$ to $n$}
          \If{$M_i$ is not a matrix}
            \State Return \texttt{fail}
            \Comment Oops!
          \EndIf
          \State $R \gets R M_i$
        \EndFor
        \State \Return $R$
    \end{algorithmic}
    \label{alg:matmult}
\end{algorithm}
Algorithm \ref{alg:matmult} is some algorithm I made up.
```

In order to compile, we need the use the preamble:

```
\usepackage{algorithm, algpseudocode}

\renewcommand{\algorithmicrequire}{\textbf{Input:}}
\renewcommand{\algorithmicensure}{\textbf{Output:}}
```

This loads the packages `algorithm` and `algpseudocode` and changes the words "Require" and "Ensure" to "Input" and "Output", respectively. This affects the lines beginning `\Require` and `\Ensure`.

It compiles to the following:

Algorithm 1 Matrix multiplication

Input: Matrices $M_1, M_2, \ldots, M_n$
Output: The matrix product $M_1 M_2 \cdots M_n$.
1: $R \leftarrow M_1$
2: **for** i from 2 to n **do**
3: **if** M_i is not a matrix **then**
4: Return `fail` ▷ Oops!
5: **end if**
6: $R \leftarrow RM_i$
7: **end for**
8: **return** R

Algorithm 1 is some algorithm I made up.

LaTeX decides where to place Algorithm 1.

We break this apart (it's similar to the table and figure environments):

- We use an *algorithm environment* via the commands `\begin{algorithm}` and `\end{algorithm}`.
- The table parameters `[htp]` give preference to the placement of the table.
- A caption `\caption{...}` is given describing the contents of the table.
- We have `\State` where we write what we do at a given step.
- We have `\Comment` where we add a comment.
- We have `\For{...}` beginning a for loop, which ends with `\EndFor`. We could alternatively use `\ForAll{...}`.
- We have `\If{...}` beginning an if statement, ended with `\EndIf`.
- We have `\Return` indicating a return statement.
- A label `\label{alg:matmult}` is given which is later referenced in the main text by `\ref{alg:matmult}`.

There are other ways of writing algorithms in LaTeX, and this is just one example.

2.3 BibTeX

The usual way of writing references in LaTeX is through BibTeX; it's used to manage references.

2.3.1 What if we don't use BibTeX?

It's possible (but awkward) to write bibliographies directly. For example, we might write:

```
Here are my references:
Anderson and Hilton \cite{AndersonHilton1983}.
Bandi et al.\ \cite{Bandi2004}.
Haynes, Hedetniemi, and Slater \cite{Haynes1998}.

\begin{thebibliography}{}

\bibitem{AndersonHilton1983}
L.~D. Anderson, A.~J.~W. Hilton,
Thanks Evans!,
{\it Proc. London Math. Soc.\/} {\bf47} (1983), 507--522.

\bibitem{Bandi2004}
N.~Bandi, C.~Sun, D.~Agrawal, A.~E. Abbadi,
Hardware acceleration in commercial databases:
 A case study of spatial operations.
In {\em Proc. VLDB Endowment} (2004), 1021--1032.

\bibitem{Haynes1998}
T.~W. Haynes, S.~T. Hedetniemi, P.~J. Slater,
{\em Fundamentals of Domination in Graphs},
Marcel Dekker, New York, 1998.

\end{thebibliography}
```

We use commands like `\cite{Bandi2004,Haynes1998}`, etc., to cite these papers. Writing references in this way gives us a lot of control—we write what we want directly. However, it's complicated for humans to write this LaTeX code, especially when there are many references. Other problems are:

- It doesn't automatically remove unused references. This is needed, since we normally do not reference papers which are not cited in the main paper. Instead, these need to be removed manually.
- If we want to change the references to a different style, we have to rewrite each one.
- References need to be sorted in some way, usually alphabetically, and this is tedious to do manually (especially when adding and deleting references).
- If we're working with multiple LaTeX documents and want to edit an entry, we need to edit multiple files. If they use the same BibTeX file, we need only edit one file.

2.3.2 BibTeX entries

We write a BibTeX text file with entries such as:

```
@article{GibsonPatterson1993,
   author = {Gibson, G. A. and Patterson, D. A.},
   title = {Designing Disk Arrays for High Data Reliability},
   journal = {J. Parallel Distr. Comput.},
   volume = {17},
   number = {1-2},
   year = {1993},
   pages = {4-27}
}

@inproceedings{Gramoli2015,
  title={Disaster-Tolerant Storage with {SDN}},
  author={Gramoli, Vincent and Jourjon, Guillaume and Mehani, Olivier},
  booktitle={Proc. International Conference on Networked Systems (NETYS)},
  pages={278-292},
  year={2015}
}

@book{trivedi2008probability,
  title={Probability \& statistics with reliability,
    queuing and computer science applications},
  author={Trivedi, Kishor S.},
  year={2008},
  publisher={John Wiley \& Sons}
}
```

Most references have one of these formats; we ordinarily don't need additional fields. I named this file `BibTeXexample.bib`. From here, we need to cite these in our paper using `\cite{...}` and tell the compiler where to find these references.

```
\documentclass{article}

\begin{document}

Here are my references:
Gibson and Patterson \cite{GibsonPatterson1993}.
Gramoli, Jourjon, and Mehani \cite{Gramoli2015}.
Trivedi \cite{trivedi2008probability}.

\bibliographystyle{siam}
\bibliography{BibTeXexample}

\end{document}
```

This compiles to give:

Here are my references: Gibson and Patterson [1]. Gramoli, Jourjon, and Mehani [2]. Trivedi [3].

References

[1] G. A. Gibson and D. A. Patterson, *Designing disk arrays for high data reliability*, J. Parallel Distr. Comput., 17 (1993), pp. 4–27.

[2] V. Gramoli, G. Jourjon, and O. Mehani, *Disaster-tolerant storage with SDN*, in Proc. International Conference on Networked Systems (NETYS), 2015, pp. 278–292.

[3] K. S. Trivedi, *Probability & statistics with reliability, queuing and computer science applications*, John Wiley & Sons, 2008.

In the above case, `\bibliographystyle{siam}` indicates the `siam` style is used,

which is my personal preference, but there are other `bibliographystyle` styles we can use.

2.3.3 BibTeX fields

BibTeX fields are fairly self explanatory. The author name(s) are added to

```
author = {Gramoli, Vincent and Jourjon, Guillaume and
    Mehani, Olivier}
```

Here **author names are separate by** `and`; if we don't write `and` between consecutive author names, BibTeX will interpret the names as a single name—one of BibTeX's quirks.

If the paper has a huge number of authors, it can be truncated using "and others":

```
author = {Kan Huang and Junlin Lu and others}
```

Note "et al." (e.g. `author = {Kan Huang and Junlin Lu et al.}`) doesn't work similarly, another of BibTeX's quirks.

The title of the paper is entered into the `title` field, e.g.

```
title = {Disaster-Tolerant Storage with {SDN}}
```

In this example, it is important to put brackets around `{SDN}`, otherwise it compiles in lowercase (another BibTeX quirk). This is important for acronyms like CPU (and SDN, in this case) and proper nouns (like software and author names).

We typically have a `journal` entry in an `@article` BibTeX entry. For example:

```
journal = {J. Parallel Distr. Comput.}
```

Ordinarily, standardized journal abbreviations are used (such as in the above example). It's often necessary to search for journal abbreviations online.

We typically have a `booktitle` entry in an `@inproceedings` BibTeX entry. For example

```
booktitle = {Proc. International Conference on
    Networked Systems (NETYS)}
```

or even

```
booktitle = {Proc. NETYS}
```

Both of these formats are acceptable in practice.

The `year` field is mostly straightforward, e.g. `year = {1993}`. Journal papers usually have a `volume` field, such as `volume = {17}`. There may also be a `number` field, but often it's best to just delete the `number` field (it's not that important); many BibTeX styles ignore the `number` field anyway.

The `pages` field is mostly straightforward, e.g., we write `pages = {4-27}`. However, we should be aware that some slight differences in this field change the results:

```
@article{Smith1,
  author = {John Smith},
  title = {An article title},
  pages = {1-2}
}

@article{Smith2,
  author = {John Smith},
  title = {An article title},
  pages = {1 - 2}
}

@article{Smith3,
  author = {John Smith},
  title = {An article title},
  pages = {1}
}
```

This compiles to the following:

[1] John Smith. An article title. pages 1–2.

[2] John Smith. An article title. pages 1 – 2.

[3] John Smith. An article title. page 1.

So in changing from `pages = {1-2}` to `pages = {1 - 2}`, we change the output (a BibTeX quirk). The result of `pages = {1}` is BibTeX interpreting it as a one-page paper, with the page numbered 1.

2.3.4 Other BibTeX quirks

Plain bibliography style If we use `\bibliographystyle{plain}`, the references in the previous example would look like:

References

[1] G. A. Gibson and D. A. Patterson. Designing disk arrays for high data reliability. *J. Parallel Distr. Comput.*, 17(1-2):4–27, 1993.

[2] Vincent Gramoli, Guillaume Jourjon, and Olivier Mehani. Disaster-tolerant storage with SDN. In *Proc. International Conference on Networked Systems (NETYS)*, pages 278–292, 2015.

[3] Kishor S. Trivedi. *Probability & statistics with reliability, queuing and computer science applications.* John Wiley & Sons, 2008.

We see that there are inconsistencies now: e.g., the author names are inconsistent (some are abbreviated, e.g., G. A. Gibson, while others are not, e.g., Vincent Gramoli).

Online BibTeX entries We cannot trust BibTeX entries available online! Here are some BibTeX citations from CiteSeerX

```
@INPROCEEDINGS{Kim06understandingthe,
    author = {Youngjae Kim and Sudhanva Gurumurthi An},
    title = {Understanding the Performance-Temperature Interactions in Disk I/O of Server Workloads},
    booktitle = {Interactions in Disk I/O of Server Workloads. In Proceedings of HPCA},
    year = {2006},
    pages = {179--189}
}

@INPROCEEDINGS{Purakayastha95characterizingparallel,
    author = {Apratim Purakayastha and Carla Schlatter Ellis and David Kotz and Ntis Nieuwejaar and Michael Best},
    title = {Characterizing parallel file-access patterns on a large-scale multiprocessor},
    booktitle = {IN PROCEEDINGS OF THE NINTH INTERNATIONAL PARALLEL PROCESSING SYMPOSIUM},
    year = {1995},
    pages = {165--172},
    publisher = {IEEE Computer Society Press}
}
```

and this is what they look like when compiled:

References

[1] Youngjae Kim and Sudhanva Gurumurthi An. Understanding the performance-temperature interactions in disk i/o of server workloads. In *Interactions in Disk I/O of Server Workloads. In Proceedings of HPCA*, pages 179–189, 2006.

[2] Apratim Purakayastha, Carla Schlatter Ellis, David Kotz, Ntis Nieuwejaar, and Michael Best. Characterizing parallel file-access patterns on a large-scale multiprocessor. In *IN PROCEEDINGS OF THE NINTH INTERNATIONAL PARALLEL PROCESSING SYMPOSIUM*, pages 165–172. IEEE Computer Society Press, 1995.

CiteSeerX is full of bugs and errors!! For reference [1], "i/o" is not capitalized and there are two book titles (probably "Proc. HPCA" is fine). For reference [2], the book title should not be capitalized, it says "In *IN*", and we don't need to give the publisher's name "IEEE Computer Society Press".

Moreover, how we write a BibTeX entry will depend on how we write the other BibTeX entries—they need to be consistent with one another; online sources cannot achieve this.

2.4 Tikz and pgfplots

It is possible to draw beautiful technical drawings using the LaTeX package tikz (`\usepackage{tikz}`) and it is possible to make beautiful plots using pgfplots (`\usepackage{pgfplots}`).

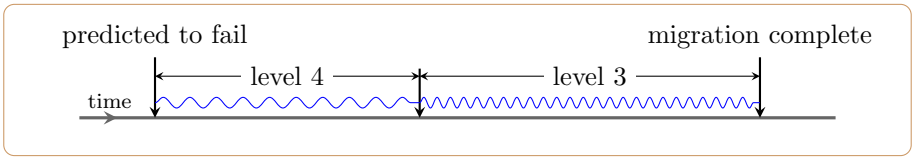
Generally, if someone is writing an important computer science paper, they put effort into making it look great. Thus if a paper does not look great, it logically follows that it's not an important paper. Moreover, figures are the most prominent feature to the reader (beyond the title and abstract). Many readers will only look at the figures in a paper, and if the figures are not well presented or are unclear, they might decide that the paper is not worth their time to read (too much effort!).

There is a webpage with tikz a pgfplots examples `http://www.texample.net/tikz/examples/`, although many are somewhat complicated.

2.4.1 Tikz

In some ways tikz is simple, in other ways it's horribly complicated. I find it best to draw what I want on a piece of paper, and break it down into circles, rectangles, lines, and so on. Then I proceed item by item, adding it to the figure, until all items are included.

To illustrate, if we want to draw



we just start with nothing, and add one item at a time. The main item in the above example is the timeline. So we draw a line from $(0, 0)$ to $(10, 0)$, and make it look significant:

```
\draw[very thick,
  color=black!60,
  postaction={decorate},
  decoration={markings,
              mark=at position 0.05 with {\arrow{stealth}}
             }
    ] (0,0) -- (10,0);
\draw[anchor=south] node at (0.4,0) {\footnotesize time};
```

To use the decoration, we need `\usetikzlibrary{decorations.markings}` in the preamble of the LaTeX file. This gives:

Then we add a brown rectangle around the outside:

```
\draw[rounded corners,color=brown!80]
                  (-1,-0.5) rectangle (11,1.5);
```

So now we have

We want to mark three points on the line, so we begin by defining three coordinates:

```
\coordinate (predict) at (1,0);
\coordinate (lev4end) at (4.5,0);
\coordinate (comp) at (9,0);
```

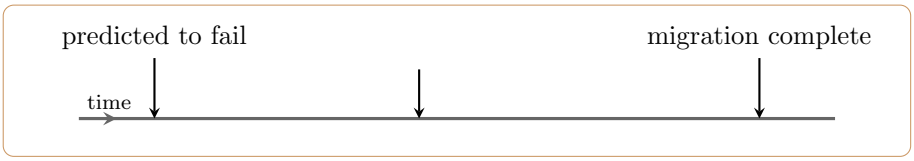
This doesn't change the drawing; it just defines the coordinates. We then add arrows to mark these points:

```
\draw[thick,-stealth] ($(predict)+(0,0.8)$) -- (predict);
\draw[anchor=south] node at ($(predict)+(0,0.8)$)
                                {predicted to fail};

\draw[thick,-stealth] ($(lev4end)+(0,0.65)$) -- (lev4end);

\draw[thick,-stealth] ($(comp)+(0,0.8)$) -- (comp);
\draw[anchor=south] node at ($(comp)+(0,0.8)$)
                                {migration complete};
```

Here the format `($(comp)+(0,0.8)$)` represents adding $(0, 0.8)$ to the coordinate named `(comp)`. To do this, we need to include `\usetikzlibrary{calc}` in the preamble. The result looks like:

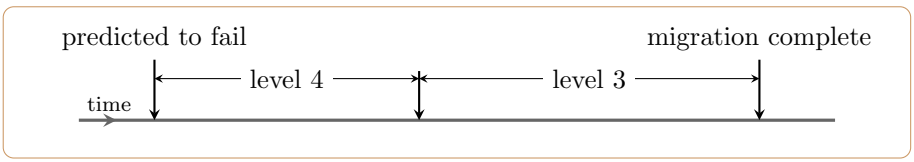


Now we want to add some arrows indicating where the "levels" apply.

```
\draw[thin,stealth-stealth] ($(predict)+(0,0.55)$)
          -- node[midway,fill=white]{level 4}
                                ($(lev4end)+(0,0.55)$);

\draw[thin,stealth-stealth] ($(lev4end)+(0,0.55)$)
          -- node[midway,fill=white]{level 3}
                                ($(comp)+(0,0.55)$);
```

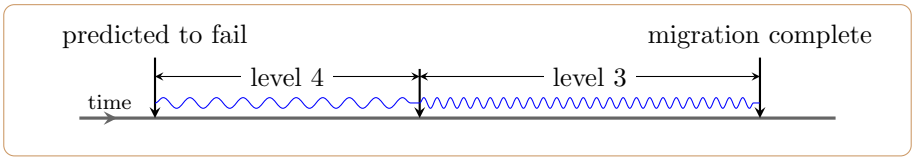
This gives:



And finally we add some wiggly lines to indicate the level:

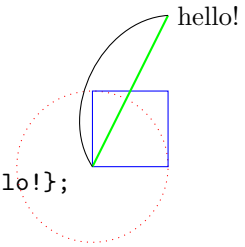
```
\draw[draw=blue,
      decorate,
      decoration={snake,amplitude=0.2em},
      segment length=10]
         ($(predict)+(0,0.2)$) to ($(lev4end)+(-0,0.2)$);
\draw[draw=blue,
      decorate,
      decoration={snake,amplitude=0.2em},
      segment length=5]
         ($(lev4end)+(0,0.2)$) to ($(comp)+(-0,0.2)$);
```

To use this, we need `\usetikzlibrary{decorations.pathmorphing}` in the preamble. This gives the final result:



Mostly, using tikz is about telling it to "put a rectangle here", "put a circle here", "draw a line between these two points", and "write [foo] here". Here's how to do this:

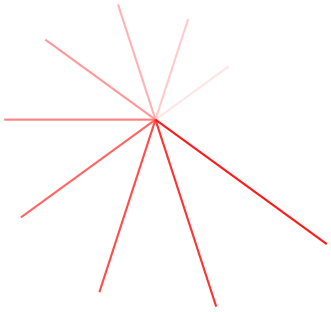
```
\begin{tikzpicture}
  \draw[red,dotted] (0,0) circle (1);
  \draw[blue] (0,0) rectangle (1,1);
  \draw[thick,green] (0,0) to (1,2);
  \draw (0,0) to[bend left=60] (1,2);
  \draw[anchor=west] node at (1,2) {hello!};
\end{tikzpicture}
```



It's also helpful to use polar coordinates:

```
\begin{tikzpicture}
  \draw[thick,red!0]  (0,0) to (0*360/10:1.0);
  \draw[thick,red!10] (0,0) to (1*360/10:1.1);
  \draw[thick,red!20] (0,0) to (2*360/10:1.2);
  \draw[thick,red!30] (0,0) to (3*360/10:1.3);
  \draw[thick,red!40] (0,0) to (4*360/10:1.4);
  \draw[thick,red!50] (0,0) to (5*360/10:1.5);
  \draw[thick,red!60] (0,0) to (6*360/10:1.6);
  \draw[thick,red!70] (0,0) to (7*360/10:1.7);
  \draw[thick,red!80] (0,0) to (8*360/10:1.8);
  \draw[thick,red!90] (0,0) to (9*360/10:1.9);
\end{tikzpicture}
```

which gives



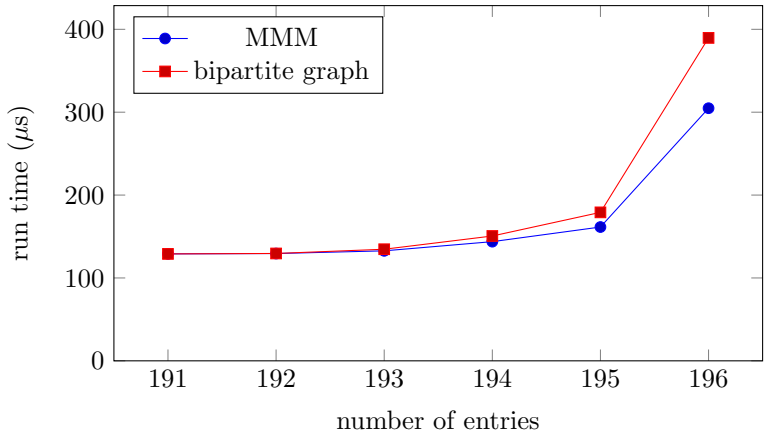
Tikz can also draw nice matrices (via `\usetikzlibrary{matrix}`):

```
\begin{tikzpicture}
  \matrix[matrix of nodes,nodes={draw}]{
    |[fill=red!30]| 1 & |[fill=blue!30]| 2 \\
    |[fill=blue!30]| 2 & |[fill=red!30]| 1 \\
  };
\end{tikzpicture}
```

which gives

2.4.2 Pgfplots

Pgfplots makes beautiful plots.



This is compiled using:

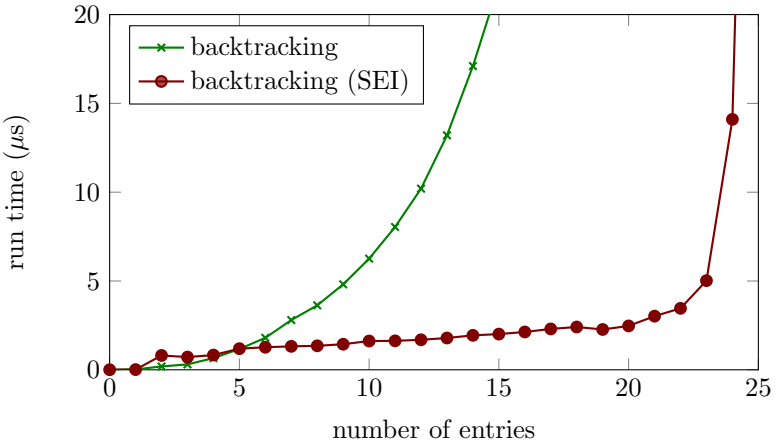
```
\begin{tikzpicture}
\begin{axis}[
  xlabel={number of entries},
  ylabel={run time ($\mu$s)},
  ymin=0,
  width=4in,
  height=0.618*4in,
  legend pos=north west
]

\addplot table [x=A,y expr={1000000*\thisrow{B}},col sep=space] {
A B
191 0.00012904
192 0.000129462
193 0.000132714
194 0.000143799
195 0.000161442
196 0.000304797
};
\addlegendentry{MMM};

\addplot table [x=A,y expr={1000000*\thisrow{C}},col sep=space] {
A C
191 0.00012901
```

```
192 0.000129545
193 0.000134647
194 0.000150615
195 0.000179139
196 0.00038963
};
\addlegendentry{bipartite graph};
\end{axis}
\end{tikzpicture}
```

It's usually easier to compile the pgfplots code from a text file. For example:



This is

```
\begin{tikzpicture}
\begin{axis}[
  xlabel={number of entries},
  ylabel={run time ($\mu$s)},
  ymin=0,
  ymax=20,
  xmin=0,
  xmax=25,
  width=4in,
  height=0.618*4in,
  legend pos=north west,
  legend style={
    cells={anchor=west},
    fill opacity=0.7,
    draw opacity=1,
    text opacity=1
  },
]

\addplot[thick,mark=x,green!50!black]
  table [x=ent,y expr={1000000*\thisrow{back}},col sep=space]
    {pgfplots_example_data.txt};
```

```
\addlegendentry{backtracking};

\addplot[thick,mark=*,red!50!black]
  table [x=ent,y expr={1000000*\thisrow{back-SEI}},col sep=space]
    {pgfplots_example_data.txt};
\addlegendentry{backtracking (SEI)};

\end{axis}
\end{tikzpicture}
```

where the data file (`pgfplots_example_data.txt`) has the contents

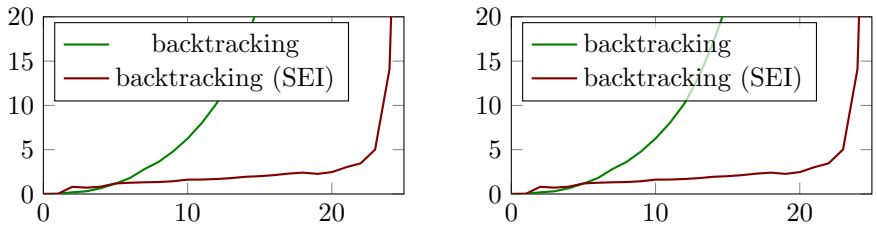
```
ent back back-SEI
0 4.40E-09 4.30E-09
1 2.31E-08 1.14E-08
2 1.87E-07 8.08E-07
3 3.06E-07 7.16E-07
4 6.59E-07 8.30E-07
5 1.17E-06 1.19E-06
6 1.80E-06 1.27E-06
7 2.80E-06 1.32E-06
8 3.63E-06 1.35E-06
9 4.81E-06 1.44E-06
10 6.26E-06 1.62E-06
11 8.04E-06 1.63E-06
12 1.02E-05 1.69E-06
13 1.32E-05 1.79E-06
14 1.71E-05 1.94E-06
15 2.17E-05 2.01E-06
16 2.82E-05 2.13E-06
17 3.63E-05 2.30E-06
18 4.55E-05 2.41E-06
19 5.92E-05 2.27E-06
20 7.59E-05 2.47E-06
21 9.97E-05 3.02E-06
22 0.000128956 3.46E-06
23 0.000164134 5.02E-06
24 0.000199825 1.41E-05
25 0.000231922 6.80E-05
```

which are some experimental results. Most of this is self explanatory. However, I feel that these settings make the legend look better:

```
  legend style={
    cells={anchor=west},
    fill opacity=0.7,
    draw opacity=1,
    text opacity=1
  },
```

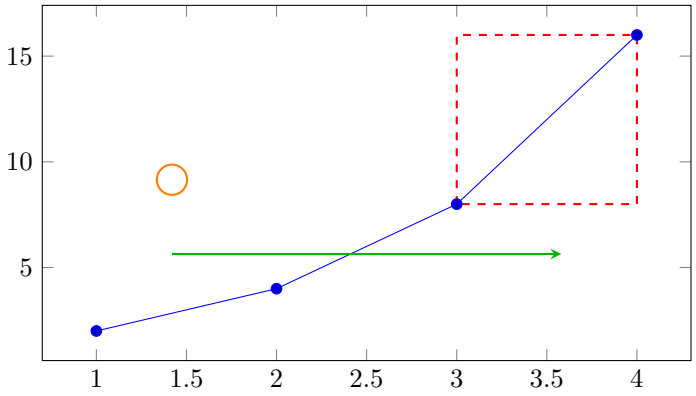
The command `cells={anchor=west}` aligns the caption text to the left. I use

the command `fill opacity=0.7` for when the legend overlaps the plots. To illustrate the difference in a smaller example:



The one on the right looks better to me. Pgfplots is customizable like this in many ways.

It is also possible to use tikz to draw on top of pgfplots plots:



This is achieved by the code:

```
\begin{tikzpicture}
\begin{axis}[
  width=4in,
  height=0.618*4in,
  legend pos=north west,
]

\addplot table [x=A,y=B,col sep=space] {
A B
1 2
2 4
3 8
4 16
};
\draw[thick,orange] (1cm,2cm) circle (0.2cm);
\draw[thick,red,dashed] (axis cs:3,8) rectangle (axis cs:4,16);
\draw[thick,green!70!black,-stealth] (rel axis cs:0.2,0.3)
                                     -- (rel axis cs:0.8,0.3);
\end{axis}
\end{tikzpicture}
```

2.5 Exercises

Exercise 2.1. Typeset the following table using a

```
\begin{tabular} ... \end{tabular}
```

environment with `\multirow` and `\rotatebox`:

		cache size (MB)	response time (sec.)
Disk 1		100	20.22
		200	12.21
		300	3.89
Disk 2		100	50.40
		200	40.22
		300	10.15

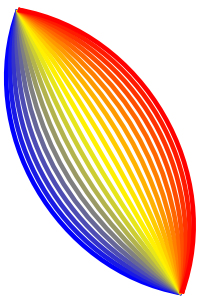
Exercise 2.2. Typeset Schrödinger's Equation:

$$i\hbar\frac{\partial\Psi}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\Psi}{\partial x^2} + V(x)\Psi(x,t) \equiv \tilde{H}\Psi(x,t). \tag{2.5}$$

and refer to it using `\eqref{...}` like so (2.5). You'll probably need to search online for how to typset $\hbar$, and ∂, and $\tilde{H}$.

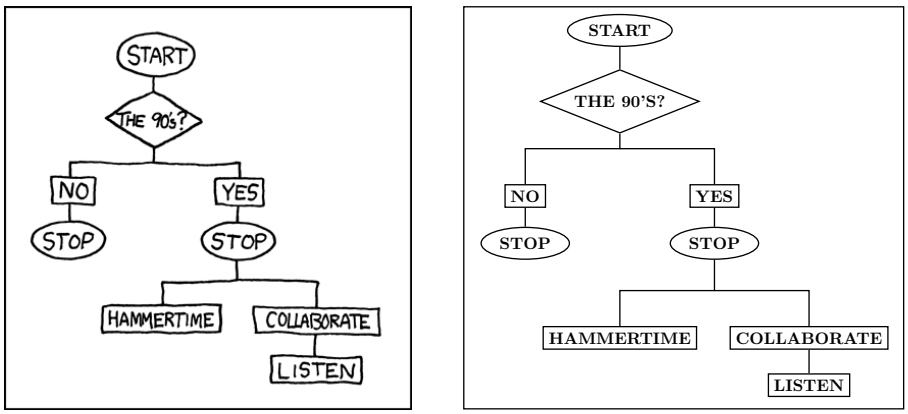
Exercise 2.3. Typeset these: ä á ă a â ã $\bar{a}$ $\dot{a}$ $\ddot{a}$ $\mathfrak{a}$ $\mathbf{a}$. The first one is typeset `\"{a}` (in text mode) which is different from `\ddot{a}` (in math mode).

Exercise 2.4. Typeset the following image using `tikz`:

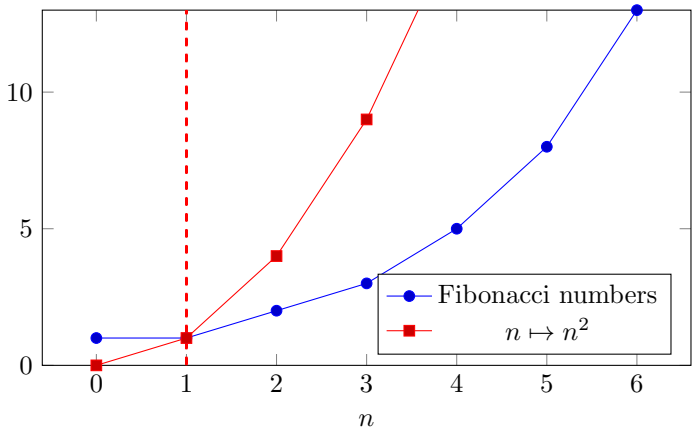


It requires three things: `\begin{tikzpicture}[rotate=30]` to rotate the picture, e.g. `color=red!40!yellow` to change the colors, and `[bend right=-50]` to bend the lines.

Exercise 2.5. Typeset the following xkcd comic flow chart (`https://www.xkcd.com/210/`) using `tikz`:



Exercise 2.6. Typeset a simple plot using pgfplots, such as:



Exercise 2.7. This is an example of a CiteSeerX BibTeX entry:

```
@INPROCEEDINGS{Mclaughlin14scalableand,
    author = {Adam Mclaughlin and David A. Bader},
    title = {Scalable and High Performance
      Betweenness Centrality on the GPU},
    booktitle = {in Proceedings of the 26th
      ACM/IEEE International Conference on High
      Performance Computing, Networking, Storage,
      and Analysis (SC},
    year = {2014}
}
```

Identify three bugs in this BibTeX entry should it be used unmodified.

Exercise 2.8. This is an example of an ACM BibTeX entry:

```
@inproceedings{Polychroniou:2015:RSV:2723372.2747645,
 author = {Polychroniou, Orestis and Raghavan, Arun
    and Ross, Kenneth A.},
 title = {Rethinking SIMD Vectorization for In-Memory Databases},
 booktitle = {Proceedings of the 2015 ACM SIGMOD International
```

```
      Conference on Management of Data},
 series = {SIGMOD '15},
 year = {2015},
 isbn = {978-1-4503-2758-9},
 location = {Melbourne, Victoria, Australia},
 pages = {1493--1508},
 numpages = {16},
 url = {http://doi.acm.org/10.1145/2723372.2747645},
 doi = {10.1145/2723372.2747645},
 acmid = {2747645},
 publisher = {ACM},
 address = {New York, NY, USA},
 keywords = {simd, vectorization, xeon phi},
}
```

Identify which parts of this BibTeX entry are actually necessary. How many times does the year "2015" appear?

Exercise 2.9. What is wrong with this BibTeX entry?

```
@INPROCEEDINGS{tem_1,
  Author         = {Comelli, P, Ferragina, P, Granieri, M.N, and Stabile, F},
  Title          = {Optical recognition of motor vehicle license plates},
  Year           = {1995},
  Volume         = {44},
  Series         = {4},
  Month          = {November},
  Pages          = {790-799},
  Publisher      = {Vehicular Technology, IEEE Transactions}
}
```

This is from `https://tex.stackexchange.com/q/130756` (and there's an answer at that site). There's multiple issues here:

- Is `@inproceedings` appropraite?
- What is wrong with the `Author` field?
- Which fields are not needed, and can be deleted?
- What is the almost invisible error in the `Pages` field (related to unicode character encoding)?