

## Specialist English, Nankai University, 2018.

**Rebecca J. Stones**

*Email:* rebecca.stones82@njb1.nankai.edu.cn

*Webpage:* <http://cc.nankai.edu.cn/>

I don't intend to set specific contact hours. If you need me:

- ▶ email me (we can discuss via email, or organize a time to meet), or
- ▶ I'm often in room 450 (my office).

### Lecture times:

- ▶ October 8 (2018) to January 14 (2018). Final exam afterwards.  
Midterm (?)
- ▶ *Lecture 1:* 10.00am to 10.45am Mondays.  
*Lecture 2:* 10.55am to 11.40am Mondays.

There's **no prescribed text**—I'm currently turning course notes into a book.

## Preamble

## Assignments

There will be 10 assignments, worth 5% each. [50% of your total mark.]

Tips:

- ▶ **Try to get 100%** on these assignments.
- ▶ **You can email me** if you need help.
- ▶ **Double-check your work.** Look for errors (nobody is immune!).

*Sherlock Holmes (2009), directed by Guy Ritchie:*

**Lord Blackwood:** *Sherlock Holmes... and his loyal dog. Tell me, Doctor, as a medical man, have you enjoyed my work?*

**Dr. John Watson:** *Let me show you how much I've enjoyed it...*

*[He rushes at Blackwood, Holmes holds him back]*

**Sherlock Holmes:** *Watson, don't! Observe...*

*[Watson sees Blackwood's trap]*

**Dr. John Watson:** *How did you see that?*

**Sherlock Holmes:** *Because I was looking for it.*

## Midterm

Last year, the final exam was too long. So I'm intending to do a **midterm exam**, somewhere in the middle of the course.

I.e., I intend to split the exam [worth 50%] into:

- ▶ a midterm, worth 20%, and
- ▶ a final exam, worth 30%.

I haven't decided when yet.

## Course aim

The overall goal of this course is to get students from “never written an academic paper before” to “publishing your first academic paper”.

- ▶ This course is **not aimed at correcting grammar quibbles**.
- ▶ Things I see repeatedly (more important than grammar quibbles):

- |                                    |                                     |
|------------------------------------|-------------------------------------|
| 1. Misleading English.             | 10. Unnecessary notation.           |
| 2. Vagueness.                      | 11. Poorly chosen notation.         |
| 3. Salesmanship.                   | 12. Misplaced content.              |
| 4. Not acknowledging drawbacks.    | 13. Informal English.               |
| 5. Fluff. (Unimportant content.)   | 14. Misused English.                |
| 6. Too much tangential content.    | 15. Unnecessary grammar constructs. |
| 7. Spelling errors.                | 16. Not rounding off numbers.       |
| 8. Low-quality figures and tables. | 17. Written for wrong audience.     |
| 9. Poorly explained mathematics.   |                                     |

**This is what the course is about.**

## Plagiarism

**Do not copy other people’s work** (other students, academics, or from websites).

Plagiarism is a serious academic offense. If you get caught doing this on a scientific paper...

- ▶ You will permanently hurt the reputation of your institute, university, and even your country.  
(If you cheat on this, what else have you cheated on?)
- ▶ You will permanently hurt the reputation of your coauthors and your teacher!!
- ▶ It affects students (your classmates) applying for PhDs, etc.

That being said:

- ▶ I expect you to *utilize the Internet*, and it’s fine to *discuss problems with others*. But **your work should be your work**.
- ▶ If in doubt, give a citation.

## LaTeX and BibTeX

## LaTeX

Computer science papers are ordinarily written using LaTeX. A decent introduction is Wikibooks: [en.wikibooks.org/wiki/LaTeX](http://en.wikibooks.org/wiki/LaTeX).

We write code, such as

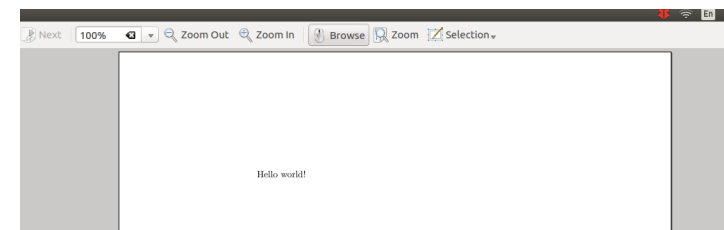
```
\documentclass{article}

\begin{document}

Hello world!

\end{document}
```

which is compiled (using e.g. pdflatex) to give a high-quality output (usually a .pdf).



**LaTeX has a lot of features!** TeX is Turing complete! You can program in it. (But don't...)

I'm not going to be able to go through all the features. I'll give a quick introduction to the most important ones.

To learn LaTeX a lot faster:

- ▶ Search the Internet for [latex example](#), or [latex template](#), and so on, and play with LaTeX code yourself.
- ▶ We *search for how to do things on the Internet* if we don't know how to do something. I find [tex.stackexchange.com](http://tex.stackexchange.com) helpful.

The 10 assignments should all be written using LaTeX.

## LaTeX software

We edit our LaTeX file in a text editor, after which its compiled (e.g., with `pdflatex`).

Useful software:

- ▶ **MikTeX** ([miktex.org](http://miktex.org)) is a LaTeX system for Windows; can be used with the editor **TeXnicCenter** ([texniccenter.org](http://texniccenter.org)). [I used to use this.]
- ▶ **TeX Live** ([tug.org/texlive](http://tug.org/texlive)) is a LaTeX system for Linux (and Windows too (?)). [This is what I use.]
- ▶ **Overleaf** ([overleaf.com](http://overleaf.com)) is a LaTeX editor and compiler that can be used from a web browser. [I use this too.]

There's *many* LaTeX software packages; see Wikipedia:

[en.wikipedia.org/wiki/Comparison\\_of\\_TeX\\_editors](http://en.wikipedia.org/wiki/Comparison_of_TeX_editors).

Which one is "best" is whichever you're most familiar with.

## LaTeX Preamble: documentclass

The first line in LaTeX code is usually `\documentclass{article}` or similar. This tells the compiler what to expect.

- ▶ `\documentclass{article}` is the default.

Options:

```
\documentclass[12pt]{article}
\documentclass[twocolumn]{article}
```

- ▶ `\documentclass{beamer}` for talk slides (like these).
- ▶ `\documentclass{standalone}` useful for high-quality .pdf figures.

Options:

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

Many publishers use their own class file...

E.g. `\documentclass{svjour3}` is used by Springer.

## LaTeX Preamble: usepackage

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

- ▶ `\usepackage{algorithm, algorithmicx, algpseudocode}` are used for displaying pseudocode.
- ▶ `\usepackage{amsmath, amsthm, amsfonts, amssymb}` are used for mathematics. Also `\usepackage{mathtools}` for even more.
- ▶ `\usepackage{tikz}` is used for drawing high-quality figures, and `\usepackage{pgfplots}` (which is compatible with tikz) is used for making high-quality plots.
- ▶ `\usepackage{CJK}` is used for writing Chinese, Japanese, and Korean.
- ▶ `\usepackage{graphicx}` is used for including external images through `\includegraphics{...}`.

There are many, many packages!

## Algorithm example

### Algorithm 1

**Input:**  $L \in \text{PLR}(r, s, n)$ , and permutations  $\alpha \in S_r$  and  $\beta \in S_s$   
**Output:** the autotopisms  $(\alpha, \beta, \gamma)$  of  $L$  with first and second components  $\alpha$  and  $\beta$ , respectively, or **fail** if no such autotopism exists

- 1: Set  $\gamma[k] \leftarrow \text{undefined}$  for all  $k \in [n]$
- 2: Set  $\gamma^{-1}[k] \leftarrow \text{undefined}$  for all  $k \in [n]$
- 3: **for all** entries  $(i, j, k) \in \text{Ent}(L)$  **do**
- 4:   **if**  $L[\alpha(i), \beta(j)] = \text{undefined}$  **then**
- 5:     Return fail                    $\triangleright$  *clash*: entry  $(i, j, k)$  maps to an empty cell
- 6:   **end if**
- 7:   Set  $k' \leftarrow L[\alpha(i), \beta(j)]$                     $\triangleright$   $\gamma$  maps  $k$  to  $k'$
- 8:   **if**  $\gamma[k] = \text{undefined}$  **then**
- 9:     Set  $\gamma[k] \leftarrow k'$  and  $\gamma^{-1}[k'] \leftarrow k$
- 10:   **else**
- 11:     **if**  $\gamma[k] \neq k'$  **then**
- 12:       Return fail                    $\triangleright$  *clash*:  $\gamma$  also maps  $k$  to something else
- 13:     **end if**
- 14:     **if**  $\gamma^{-1}[k'] \neq k$  **then**
- 15:       Return fail                    $\triangleright$  *clash*:  $\gamma$  also maps something else to  $k'$
- 16:     **end if**
- 17:   **end if**
- 18: **end for**
- 19: **return**  $(\alpha, \beta, \gamma)$

```
\documentclass{article}
\usepackage{amsmath}
\usepackage{algorithm, algpseudocode}
\renewcommand{\algorithmicrequire}{\textbf{Input:}}
\renewcommand{\algorithmicensure}{\textbf{Output:}}
\begin{document}
\begin{algorithm}[htp]
\caption{}
\begin{algorithmic}[1]
\Require $L \in \mathsf{PLR}(r, s, n)$, and permutations
  $\alpha \in S_r$ and $\beta \in S_s$
\Ensure the autotopisms $(\alpha, \beta, \gamma)$ of $L$
with first and second components $\alpha$ and $\beta$,
respectively, or \texttt{fail} if no such autotopism exists
\State Set $\gamma[k] \gets \texttt{undefined}$
for all $k \in [n]$
\State Set $\gamma^{-1}[k'] \gets \texttt{undefined}$
for all $k' \in [n]$
\ForAll{entries $(i, j, k) \in \mathsf{Ent}(L)$}
\If{($L[\alpha(i), \beta(j)] = \texttt{undefined}$)}
\Comment{\textit{clash}: entry $(i, j, k)$
maps to an empty cell}
\EndIf
\State Set $k' \gets L[\alpha(i), \beta(j)]$
\Comment{($\gamma$ maps $k$ to $k'$)}
\If{($\gamma[k] = \texttt{undefined}$)}
\State Set $\gamma[k] \gets k'$
and $\gamma^{-1}[k'] \gets k$
\Else
\If{($\gamma[k] \neq k'$)}
\State Return \texttt{fail}
\Comment{\textit{clash}: $\gamma$ also maps $k$ to something else}
\EndIf
\If{($\gamma^{-1}[k'] \neq k$)}
\State Return \texttt{fail}
\Comment{\textit{clash}: $\gamma$ also maps something else to $k'$}
\EndIf
\EndIf
\EndFor
\State \Return $(\alpha, \beta, \gamma)$
\end{algorithmic}
\end{algorithm}
\end{document}
```

## Algorithms (preamble)

More details. The preamble:

```
\usepackage{algorithm, algpseudocode}

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

\begin{document}

\begin{algorithm}[htp]
\caption{}
\begin{algorithmic}[1]
```

Input/output:

```
\Require $L \in \mathsf{PLR}(r, s, n)$, and permutations
  $\alpha \in S_r$ and $\beta \in S_s$
\Ensure the autotopisms $(\alpha, \beta, \gamma)$ of $L$
with first and second components $\alpha$ and $\beta$,
respectively, or \texttt{fail} if no such autotopism exists
```

### Algorithm 1

**Input:**  $L \in \text{PLR}(r, s, n)$ , and permutations  $\alpha \in S_r$  and  $\beta \in S_s$

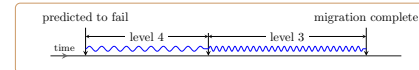
**Output:** the autotopisms  $(\alpha, \beta, \gamma)$  of  $L$  with first and second components  $\alpha$  and  $\beta$ , respectively, or **fail** if no such autotopism exists

## Algorithms (for loops; if statements; comments)

- 3: **for all** entries  $(i, j, k) \in \text{Ent}(L)$  **do**
- 4:   **if**  $L[\alpha(i), \beta(j)] = \text{undefined}$  **then**
- 5:     Return fail                    $\triangleright$  *clash*: entry  $(i, j, k)$  maps to an empty cell
- 6:   **end if**
- 7:   Set  $k' \leftarrow L[\alpha(i), \beta(j)]$                     $\triangleright$   $\gamma$  maps  $k$  to  $k'$
- 8:   **if**  $\gamma[k] = \text{undefined}$  **then**
- 9:     Set  $\gamma[k] \leftarrow k'$  and  $\gamma^{-1}[k'] \leftarrow k$
- 10:   **else**
- 11:     **if**  $\gamma[k] \neq k'$  **then**
- 12:       Return fail                    $\triangleright$  *clash*:  $\gamma$  also maps  $k$  to something else
- 13:     **end if**
- 14:     **if**  $\gamma^{-1}[k'] \neq k$  **then**
- 15:       Return fail                    $\triangleright$  *clash*:  $\gamma$  also maps something else to  $k'$
- 16:     **end if**
- 17:   **end if**
- 18: **end for**

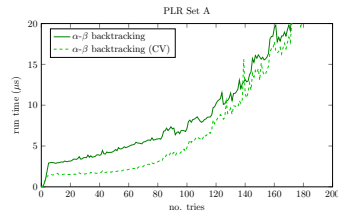
```
\ForAll{entries $(i, j, k) \in \mathsf{Ent}(L)$}
\If{($L[\alpha(i), \beta(j)] = \texttt{undefined}$)}
\State Return \texttt{fail}
\Comment{\textit{clash}: entry $(i, j, k)$
maps to an empty cell}
\EndIf
\State Set $k' \gets L[\alpha(i), \beta(j)]$
\Comment{($\gamma$ maps $k$ to $k'$)}
\If{($\gamma[k] = \texttt{undefined}$)}
\State Set $\gamma[k] \gets k'$
and $\gamma^{-1}[k'] \gets k$
\Else
\If{($\gamma[k] \neq k'$)}
\State Return \texttt{fail}
\Comment{\textit{clash}: $\gamma$ also maps $k$ to something else}
\EndIf
\If{($\gamma^{-1}[k'] \neq k$)}
\State Return \texttt{fail}
\Comment{\textit{clash}: $\gamma$ also maps something else to $k'$}
\EndIf
\EndFor
```

## Tikz example



```
\documentclass{article}
\usepackage{tikz}
\usetikzlibrary{calc}
\usetikzlibrary{decorations.pathmorphing}
\usetikzlibrary{decorations.markings}
\begin{document}
\begin{tikzpicture}
\draw[rounded corners,color=brown!80] (-1,-0.5) rectangle (11.1,5);
\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};
\coordinate (predict) at (1,0);
\coordinate (lev4end) at (4.5,0);
\coordinate (comp) at (9,0);
\draw[thick,-stealth] ($(predict)+(0,0.8)$) -- (predict);
\draw[anchor=south] node at ($(predict)+(0,0.8)$) {\textit{predicted to fail}};
\draw[thick,-stealth] ($(lev4end)+(0,0.65)$) -- (lev4end);
\draw[thick,-stealth] ($(comp)+(0,0.8)$) -- (comp);
\draw[anchor=south] node at ($(comp)+(0,0.8)$) {\textit{migration complete}};
\draw[thin,stealth-stealth] ($(predict)+(0,0.55)$)
-- node[midway,fill=white] (level 4) {$\textit{level 4}$}
($(lev4end)+(0,0.55)$);
\draw[thin,stealth-stealth] ($(lev4end)+(0,0.55)$)
-- node[midway,fill=white] (level 3) {$\textit{level 3}$}
($(comp)+(0,0.55)$);
\draw[draw=blue,decorate,decoration={snake,amplitude=0.2em,segment length=10}
] ($(predict)+(0,0.2)$) to ($(lev4end)+(-0.2,0.2)$);
\draw[draw=blue,decorate,decoration={snake,amplitude=0.2em,segment length=5}
] ($(lev4end)+(0,0.2)$) to ($(comp)+(-0.2,0.2)$);
\end{tikzpicture}
\end{document}
```

## Pgfplots example



```
\documentclass[crop]{standalone}

\usepackage{tikz}
\usepackage{pgfplots}
\pgfplotsset{compat=1.13}

\begin{document}

\begin{tikzpicture}
\begin{axis}[
title={PLR Set A},
xlabel={no. tries},
ylabel={run time (\mu s)},
xmin=0,
xmax=200,
ymin=0,
ymax=20,
width=\textwidth,
height=0.618\textwidth,
thick,
legend style={
cells={anchor=west},
legend pos=north west
}
]

\addplot[green!50!black] table [x=nr_entry_tries,
y expr={1000000*\thisrowno{1}},
col sep=space,mark=none] {AB456a.dat};
\addlegendentry{$\alpha$-$\beta$ backtracking};
\addplot[green!80!black,dashed] table [x=nr_entry_tries,
y expr={1000000*\thisrowno{2}},
col sep=space,mark=none] {AB456a.dat};
\addlegendentry{$\alpha$-$\beta$ backtracking (CV)};
\end{axis}
\end{tikzpicture}

\end{document}
```

Here AB456a.dat is an input data file with a column labeled nr\_entry\_tries.

## Image example; URL example



Figure 1: Fujitsu hard disk. Sourced from Wikimedia Commons [https://commons.wikimedia.org/wiki/File:Harddisk\\_1.jpg](https://commons.wikimedia.org/wiki/File:Harddisk_1.jpg)

Figure 1 shows an example of a hard disk.

```
\documentclass{article}

\usepackage{graphics}
\usepackage{url}

\begin{document}

\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.

\end{document}
```

Using the `\usepackage{url}` package, we can write `\url{www.google.com}` which will display as [www.google.com](http://www.google.com).

It allows line breaks in long URLs:

<https://tex.stackexchange.com/questions/51040/avoiding-url-error-url-used-in-a-moving-argument-in-a-custom-environment>.

It's not always great. Better to find a short URL:

<https://tex.stackexchange.com/q/51040>

This doesn't work within a `\caption{...}` (as above) and we need to add `\protect`.

## Titles

My title!

Rebecca J. Stones

Last updated: September 24, 2017

**Abstract**

This is my abstract. It's not very long.

Hello world!

```
\documentclass{article}

\title{My title!}
\author{Rebecca J. Stones}
\date{Last updated: \today}

\begin{document}

\maketitle

\begin{abstract}
This is my abstract. It's not very long.
\end{abstract}

Hello world!

\end{document}
```

Ordinarily, we want a document to have

1. a title, where we use `\title{...}`,
2. an author, where we use `\author{...}`,
3. a date, where we use `\date{...}` and possibly `\today`, and
4. an abstract, which is included in an environment `\begin{abstract}` and `\end{abstract}`.

## Titles (cont.)

There's other things: affiliations, email addresses, keywords, classification numbers, acknowledgements. These should follow the LaTeX template of the journal or conference.

We can add them in if needed:

```
\documentclass{article}

\usepackage{url}

\title{My title!}
\author{Rebecca J. Stones\textit{[0.3em]}
\small College of Computer Science,\textit{[0.3em]}
\small Nankai University, China\textit{[0.3em]}
\small \url{rebecca.stones82@njb1.nankai.edu.cn}}
\date{Last updated: \today}

\begin{document}

\maketitle

\begin{abstract}
This is my abstract. It's not very long.
\fill\textit{Key words}: Nankai; university
\end{abstract}

Hello world!

\end{document}
```

## Sections

We use `\section{...}` for sections, `\subsection{...}` for subsections, `\subsubsection{...}` for sub-subsections, and `\paragraph{...}` which break up sub-subsections.

### 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.

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

There's also `\chapter{...}`.

Also, we can `\label{...}` a section and `\ref{...}` to it.

## A randomly generated example...

The Effect of "Fuzzy" Algorithms on  
Steganography

SClgen

September 21, 2017

### Abstract

Compact symmetries and hash tables have garnered limited interest from both cyberneticists and theorists in the last several years. In this paper, we argue the study of Scheme, which embodies the intuitive principles of hardware and architecture. It at first glance seems counterintuitive but has ample historical precedence. We construct a self-learning tool for improving DHTs (Grucche), disproving that this clients can be made self-learning, introspective, and encrypted.

## 1 Introduction

The investigation of scatter/gather I/O is a natural quagmire. On the other hand, a confusing question in networking is the understanding of psychoacoustic technology.

## 2 Related Work

Several introspective and ubiquitous heuristics have been proposed in the literature. Our system also investigates the deployment of the lookaside buffer, but without all the unnecessary complexity.

## 3 Performance Results

We now discuss our performance analysis. Our overall evaluation strategy seeks to prove three hypotheses: (1) that signal-to-noise ratio stayed constant across successive generations of Macintosh SEs; (2) that RAID no longer adjusts a heuristic's unstable code complexity; and finally (3) that instruction rate is an outmoded way to measure clock speed.

### 3.1 Hardware and Software Configuration

One must understand our network configuration to grasp the genesis of our results. Soviet security experts ran a software emulation on the KGB's autonomous cluster to measure the computationally signed nature of interactive modalities.

## 4 Conclusion

Here we demonstrated that wide-area networks and evolutionary programming can agree to surmount this quandary. Grucche has set a precedent for the synthesis of forward-error correction, and we expect that system administrators will emulate our application for years to come.

```
\documentclass[preview]{standalone}
\title{The Effect of "Fuzzy" Algorithms on Steganography}
\author{SClgen}
\date{\today}
\begin{document}
\maketitle
\begin{abstract}
Compact symmetries and hash tables have garnered limited interest from both cyberneticists and theorists in the last several years. In this paper, we argue the study of Scheme, which embodies the intuitive principles of hardware and architecture. It at first glance seems counterintuitive but has ample historical precedence. We construct a self-learning tool for improving DHTs (Grucche), disproving that this clients can be made self-learning, introspective, and encrypted.
\end{abstract}
\section{Introduction}
The investigation of scatter/gather I/O is a natural quagmire. On the other hand, a confusing question in networking is the understanding of psychoacoustic technology.
\section{Related Work}
Several introspective and ubiquitous heuristics have been proposed in the literature. Our system also investigates the deployment of the lookaside buffer, but without all the unnecessary complexity.
\section{Performance Results}
We now discuss our performance analysis. Our overall evaluation strategy seeks to prove three hypotheses: (1) that signal-to-noise ratio stayed constant across successive generations of Macintosh SEs; (2) that RAID no longer adjusts a heuristic's unstable code complexity; and finally (3) that instruction rate is an outmoded way to measure clock speed.
\subsection{Hardware and Software Configuration}
One must understand our network configuration to grasp the genesis of our results. Soviet security experts ran a software emulation on the KGB's autonomous cluster to measure the computationally signed nature of interactive modalities.
\section{Conclusion}
Here we demonstrated that wide-area networks and evolutionary programming can agree to surmount this quandary. Grucche has set a precedent for the synthesis of forward-error correction, and we expect that system administrators will emulate our application for years to come.
\end{document}
```

## Mathematics: inline equations

In LaTeX, we can use inline mathematics by surrounding it by dollar signs `$...$`. For example

- ▶ `$y=mx+c$` displays as  $y = mx + c$ ,
- ▶ `$x^2$` displays as  $x^2$ ,
- ▶ `$x^{x^2}$` displays as  $x^{x^2}$ , and `$x^{x\{x^2\}}$` displays as  $x^{x^{x^2}}$ ,
- ▶ `$\sin(\log(\sqrt{x+1}))$` displays as  $\sin(\log(\sqrt{x+1}))$ ,
- ▶ `$\frac{-b \pm \sqrt{b^2-4ac}}{2a}$` displays as  $\frac{-b \pm \sqrt{b^2-4ac}}{2a}$ ,
- ▶ We can write Greek letters `$\sigma$` and `$\Sigma$` which gives  $\sigma$  and  $\Sigma$ ,
- ▶ We can do summations: `$\sum_{k=1}^n k = \frac{1}{2}n(n+1)$` displays as  $\sum_{k=1}^n k = \frac{1}{2}n(n+1)$ ,
- ▶ and integrals `$\int_0^\infty x^2 dx$` displays as  $\int_0^\infty x^2 dx$ .

## Mathematics: displayed equations

In LaTeX, we can use displayed mathematics by surrounding it by `\[...\]`. For example `\[\sum_{k=1}^n k = \frac{n(n+1)}{2}\]` displays as

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

and we can number the equations using an equation environment:

```
\begin{equation}\label{eq:A}
A = \sqrt{1-\frac{a^2}{b^2}}
\end{equation}
is equation \eqref{eq:A}.
```

produces:

$$A = \sqrt{1 - \frac{a^2}{b^2}} \quad (1)$$

is equation (1).

## Floats: tables

a	b	c	I'm writing in LaTeX
a a a	b b b	c c c	Here's some more LaTeX
I'm in the middle!			
L		R	
hello	1	2	3
	4	5	6
	7	8	9

Table 1: Example table in LaTeX.

Look at Table 1—it's great!

```
\documentclass{article}

\usepackage{multirow}
\usepackage{rotating}

\begin{document}

\begin{table}[htp]
\centering
\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}{I'm in the middle!} \\
\hline
L & & R & \\
\hline
hello & 1 & 2 & 3 \\
& 4 & 5 & 6 \\
& 7 & 8 & 9 \\
\hline
\end{tabular}
\caption{Example table in LaTeX.}\label{ta:example}
\end{table}

Look at Table \ref{ta:example}---it's great!

\end{document}
```

Some people prefer tables drawn using the `\usepackage{booktabs}` package.

## Floats: figures

We had this example from before:



Figure 1: Fujitsu hard disk. Sourced from Wikimedia Commons [https://commons.wikimedia.org/wiki/File:Harddisk\\_1.jpg](https://commons.wikimedia.org/wiki/File:Harddisk_1.jpg)

Figure 1 shows an example of a hard disk.

```
\documentclass{article}

\usepackage{graphicx}
\usepackage{url}

\begin{document}

\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.

\end{document}
```

## References: *not* BibTeX

We can write our own references such as in the following:

Here are my references: Anderson and Hilton [1]. Bandi et al. [2]. Haynes, Hedetniemi, and Slater [3].

### References

- [1] L. D. Anderson, A. J. W. Hilton, Thanks Evans!, *Proc. London Math. Soc.* **47** (1983), 507--522.
- [2] N. Bandi, C. Sun, D. Agrawal, A. E. Abbadi, Hardware acceleration in commercial databases: A case study of spatial operations. In *Proc. VLDB Endowment* (2004), 1021--1032.
- [3] T. W. Haynes, S. T. Hedetniemi, P. J. Slater, *Fundamentals of Domination in Graphs*, Marcel Dekker, New York, 1998.

```
\documentclass{article}

\begin{document}

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}

\end{document}
```

*Not a good idea:* If we need to change bibliography styles, we need to rewrite every reference.

## References: BibTeX

Instead we use BibTeX to generate this for us. We write a `.bib` file (say `myreferences.bib`) and use this LaTeX code:

```
\bibliographystyle{siam}
\bibliography{myreferences}
```

BibTeX generates a `.bbl` file (`myreferences.bbl`) containing the formatted references.

Here `siam` is a format style. (My favorite.) There's other styles like `plain`.

**Tip:** BibTeX entries available online contain many bugs!! Check them carefully.

## References: BibTeX (journal)

BibTeX entry:

```
@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}}
}
```

Gibson and Patterson [1].

### 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.

Tips:

- ▶ Write: Gibson and Patterson `\cite{GibsonPatterson1993}`.
- ▶ Be careful writing author names.
- ▶ Use proper journal abbreviations.
- ▶ Proofread the .pdf output.

## References: BibTeX (conference)

BibTeX entry:

```
@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}}
}
```

Gramoli, Jourjon, and Mehani [1].

### References

- [1] V. GRAMOLI, G. JOURJON, AND O. MEHANI, *Disaster-tolerant storage with SDN*, in Proc. International Conference on Networked Systems (NETYS), 2015, pp. 278–292.

Tips:

- ▶ Write: Gramoli, Jourjon, and Mehani `\cite{Gramoli2015}`.
- ▶ Note {SDN} in brackets in title = {...}, otherwise it will be lowercase in the .pdf.
- ▶ Proofread the .pdf output!

## References: BibTeX (book)

BibTeX entry:

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

Trivedi [1].

### References

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

Tips:

- ▶ Write: Trivedi `\cite{trivedi2008probability}`.
- ▶ Proofread the .pdf output!!

# More about LaTeX...



## Arrays

```
\[ \begin{array}{c|cccc}
+ & 0 & 1 & 2 & 3 \\ \hline
0 & 0 & 1 & 2 & 3 \\
1 & 1 & 2 & 3 & 4 \\
2 & 2 & 3 & 4 & 5 \\
3 & 3 & 4 & 5 & 6 \\ \hline
\end{array} \]
```

compiles to

+	0	1	2	3
0	0	1	2	3
1	1	2	3	4
2	2	3	4	5
3	3	4	5	6

a displayed equation containing an array.

## Arrays (nested)

```
\[ \begin{array}{cc}
\begin{array}{|cc|} \hline
1 & 2 \\ \hline
\end{array} & \begin{array}{|cc|} \hline
5 & 6 \\ \hline
\end{array} \\
\begin{array}{|cc|} \hline
9 & 10 \\ \hline
\end{array} & \begin{array}{|cc|} \hline
13 & 14 \\ \hline
\end{array} \\
\begin{array}{|cc|} \hline
11 & 12 \\ \hline
\end{array} & \begin{array}{|cc|} \hline
15 & 16 \\ \hline
\end{array} \\ \hline
\end{array} \]
```

compiles to

1 2	5 6
3 4	7 8
9 10	13 14
11 12	15 16

an array, with arrays as entries.

## Itemize

```
\begin{itemize}
\item This is an item.
\item[$\star$] This item has a star!
\item This item has items of its own.
\begin{itemize}
\item Oh...
\item Another item.
\end{itemize}
\end{itemize}
```

compiles to

- This is an item.
- ★ This item has a star!
- This item has items of its own.
  - Oh...
  - Another item.

an itemized list, containing another itemized list.

## Enumerate

```
\begin{enumerate}
\item This is an item.
\item This is another item.
\item Look:
\begin{enumerate}
\item This item has items of its own.
\item Another item.
\end{enumerate}
\end{enumerate}
One more:
\begin{enumerate}[resume]
\item Forgot this one.
\end{enumerate}
```

compiles to

1. This is an item.
2. This is another item.
3. Look:
  - (a) This item has items of its own.
  - (b) Another item.
- One more:
4. Forgot this one.

an enumerated list, containing another enumerated list.

Using `\usepackage{enumitem}`, we can resume enumeration.

## Theorems

Making use of `\usepackage{amsmath, amssymb, amsthm}` we can define our own theorem environments:

```
\newtheorem{theo}{Theorem}
```

Once we have done that:

```
\begin{theo}
For all  $x \in \mathbb{R}$ , we have  $x=x$ .
\end{theo}
```

```
\begin{proof}
This is axiomatic.
\end{proof}
```

compiles to give

**Theorem 1.** *For all  $x \in \mathbb{R}$ , we have  $x = x$ .*

*Proof.* This is axiomatic. □

## Other tidbits

- ▶ **Macros** We can define macros such as `\def\Ncond{N^{\text{cond}}}`. Afterwards, if we call it via `$_\Ncond$`, it replaces `\Ncond` with `N^{\text{cond}}` which looks like:  $N^{\text{cond}}$ .
- ▶ **Spacing** If we write “Monsters Inc. is great” the period “.” is compiled like a full stop at the end of a sentence (so LaTeX adds a longer space afterwards).
  - ▶ Instead use “Monsters Inc.\ is great” for correct spacing.
  - ▶ Or use “Monsters Inc.\~is great” which is a non-breaking space.
- ▶ **Raisebox** We can move things about by e.g. `\raisebox{1in}{hi!!}` which raises the text `hi!!` up one inch. We could instead lower it one inch by `\raisebox{-1in}{hi!!}`
- ▶ **Vertical space and horizontal space** We can add positive and/or negative space such as with `\vspace{1in}` for vertical space or `\hspace{1in}` for horizontal space.
- ▶ **Math symbols** using `\usepackage{amssymb}`

`$_\mathbb{A}$,  $\mathcal{A}$ ,  $\mathfrak{A}$`

$\mathbb{A}$ ,  $\mathcal{A}$ ,  $\mathfrak{A}$