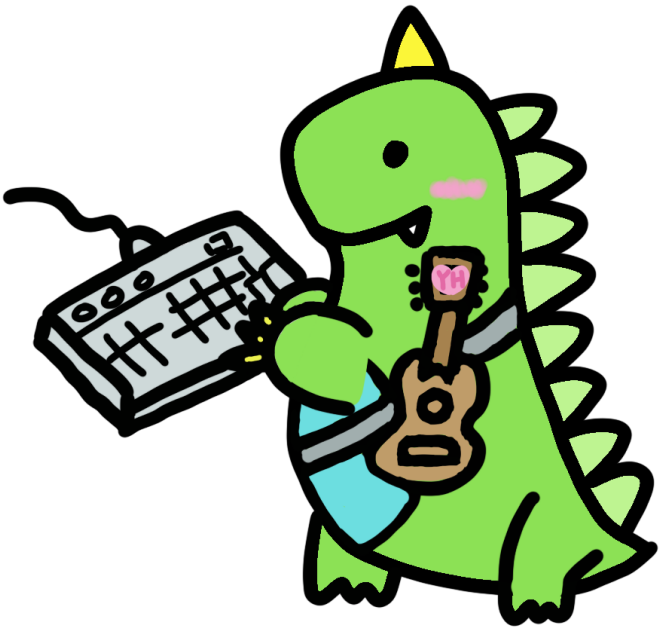
Theorem-You-Must-Know

Some theorem proofs related to Cryptography or computational number theory.

@fffmath



Contents

1 Introduction

The original TeX was created by the famous computer scientist Donald Knuth (Knuth and Bibby, 1984), and added to by Leslie Lamport to make LaTeX (Lamport, 1985).

1.1 How to Use this Book

These are the main ways you can use this material:

- Lattice
- Zero-Knowledge Proof
- You can use it as a template. All of the source files used to make this book are freely available in GitHub at `https://github.com/dwiddows/ebookbook` and Overleaf. The source files are laid out in a way that should make it easy to clone the project and adapt it for your own book.

2 Lattice Based Cryptography

2.1 The two definitions of lattice are equivalent

Definition 1 (Lattice)**.** *Given n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, the lattice generated by them is defined as*

$$\mathbb{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \;\middle|\; x_i \in \mathbb{Z} \right\}$$

Definition 2. *A lattice $\mathbb{L}$ is a discrete additive subgroup of $\mathbb{R}^n$.*

Theorem 1. *The two definitions of lattice are equivalent.*

Proof. We will first show that Definition 1 $\Rightarrow$ Definition 2.

Assume $\mathbb{L}$ is a lattice defined as the set of all integer combinations of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$ which are linearly independent (Definition 1). Then, clearly L is an additive subgroup of $\mathbb{R}^n$. In addition, $\forall \mathbf{x}, \mathbf{y} \in L$, $\mathbf{x} - \mathbf{y} \in L$. Therefore, from the lower bound on a shortest lattice vector,

$$\|\mathbf{x} - \mathbf{y}\| \geq \lambda_1(\mathbb{L}) \geq \min_{i=1,\ldots,n} \|\tilde{\mathbf{b}}_i\|.$$

In other words, the length of any lattice vector must be greater than the length of a shortest lattice vector. Therefore, we can let $\varepsilon = \lambda_1$. So, both properties of Definition 2 are satisfied (L is a discrete additive subgroup of $\mathbb{R}^n$).

We show that Definition 2 $\Rightarrow$ Definition 1. Given a discrete additive subgroup L of $\mathbb{R}^n$, we construct a set of basis using the algorithm below.

We will use the following definition of a closed parallelepiped:

Given n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, their closed fundamental parallelepiped is defined as

$$\overline{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \;\middle|\; x_i \in \mathbb{R}, 0 \leq x_i \leq 1 \right\}$$

Pick $\mathbf{y} \in \mathbb{L}$ such that there is no lattice vector between the zero vector and $\mathbf{y}$. Let $\mathbf{b}_1 = \mathbf{y}$. Iterate for all i, $1 \leq i < n$: Assume we have already chosen $\mathbf{b}_1, \ldots, \mathbf{b}_i$. Choose $\mathbf{y}$ not in the span of $\mathbf{b}_1, \ldots, \mathbf{b}_i$. Consider a $\overline{P}(\mathbf{b}_1, \ldots, \mathbf{b}_i, \mathbf{y})$ (See Figure-1 for an example). Now, $\overline{P}$ contains at least one lattice point (namely $\mathbf{y}$) and it contains finitely many lattice points. Now, choose a vector $\mathbf{z} \in \overline{P}(\mathbf{b}_1, \ldots, \mathbf{b}_i, \mathbf{y}) \setminus \text{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_i)$ such that $\text{dist}(\mathbf{z}, \text{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_i))$ is the smallest.

Note that we can do this since we have only finitely many points to choose from. Let $\mathbf{b}_{i+1} = \mathbf{z}$.

We will now show that the above algorithm returns a basis $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$ for the lattice. Clearly, all $\mathbf{b}_i$s are in $\mathbb{R}^m$ and they are linearly independent by the algorithm that we used. We are left to show that $L \subseteq \{\sum x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$.

Let $\mathbf{z} = \sum z_i \mathbf{b}_i$ be an arbitrary lattice vector (where $z_i \in \mathbb{R}$). Let $\mathbf{z}_0 = \sum b_z^i \mathbf{c}_i$ be an element of L. Then, $\mathbf{z} - \mathbf{z}_0 = \sum (z_i - b_z^i c_i)\mathbf{b}_i$ is in L. We will show that all coefficients z_i must be integers. Express

$$\mathbf{z} - \mathbf{z}_0 = (z_n - \lfloor z_n \rfloor)\mathbf{b}_n + \mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}) = (z_n - \lfloor z_n \rfloor)\mathbf{b}_n + \mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}).$$

In other words, vector $\mathbf{z} - \mathbf{z}_0$ is in the span of $\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}$ plus a multiple of $\tilde{\mathbf{b}}_n$ with coefficients $0 \leq b_z^n c_n < 1$.

Now,

$$\mathrm{dist}(\mathbf{z} - \mathbf{z}_0, \mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})) = (z_n - \lfloor b_n \rfloor)\|\tilde{\mathbf{b}}_n\|.$$

This follows because the distance is defined as the orthogonal component of $\mathbf{z} - \mathbf{z}_0$ to the span $\mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})$, which is precisely $(z_n - b_z^n c_n)\|\tilde{\mathbf{b}}_n\|$. Similarly,

$$\mathrm{dist}(\mathbf{b}_n, \mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})) = \|\tilde{\mathbf{b}}_n\|.$$

In addition, since $0 \leq (z_n - b_z^n c_n) < 1$,

$$\mathrm{dist}(\mathbf{z} - \mathbf{z}_0, \mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})) < \mathrm{dist}(\mathbf{b}_n, \mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})).$$

But since $\mathbf{b}_n$ was chosen as the closest vector to $\mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})$, $\mathbf{z} - \mathbf{z}_0$ must be linearly dependent on $\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}$. Therefore, $z_n - \lfloor b_n \rfloor = 0$ and so $z_n \in \mathbb{Z}$.

By recursively repeating the above argument for $\mathbf{z} = \mathbf{z} - \mathbf{z}_i \in \mathrm{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})$ for all $1 < i \leq n$, we obtain that all coefficients z_j for $1 \leq j \leq n$ must be integers.

□

3 End to End: From Source to eReader

We'll start with the end-to-end process for making a book you can read on your eReader device. Once you can do this, change the title, add or remove chapters, and see the changes you've made reflected on your eReader, that will hopefully be the best encouragement to take the parts that are most useful and get started with your own book. The basic outline of the process is shown in Figure 3. The rest of this chapter walks through the steps in more detail.

3.1 Install Dependencies

Typically when you want to use a new piece of software, it has dependencies — other libraries and packages it needs for some of its functionality. For end-user applications, these are usually bundled together so that everything works 'out of the box'. For programming tools, there's normally some system to check which dependencies you already have, and to install only the new ones that you need. How this works depends partly on what platform you're using. Here we'll just list the dependencies you'll need to find and install.

These include:

- A LaTeX system, including a program called `make4ht` or the older `htlatex`. This is crucial for making HTML output, as described in Section 3.3.
- Some eBook converter software. The one recommended below is `ebook-convert`, from Calibre (see `https://manual.calibre-eBook.com/ebook-convert.html`, or just search the web).
- Some eReader or previewer software, such as a Kindle device or app, or another eBook app such as Calibre (see above). You'll want this for seeing how your book looks 'for real' (though of course, not all eBook apps look exactly alike, and users have different settings).
- (Recommended) A shell program that runs bash scripts so that you can run the `build.sh` command. If you don't already have a terminal where you can run bash or a close equivalent, it's probably worth installing and using, and if that instruction sounds difficult, the rest of the process may be hard.

Every modern operating system — basically Windows, Linux, MacOS, and similar variants — has a variety of package installation tools. Some of the dependencies above

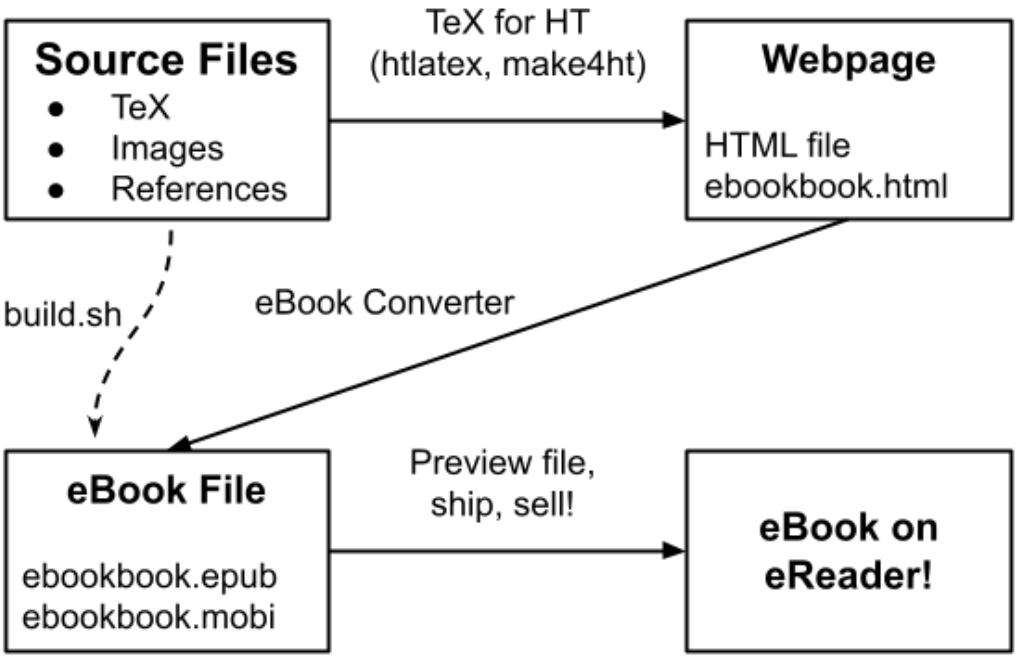


Figure 3.1: Steps in the pipeline for shipping this eBook

can be done at the command-line, sometimes it's easier to download and install something directly from a provider's website.

The best place to keep links to such dependencies and recommended options is the github project wiki at `https://github.com/dwiddows/ebookbook/wiki`.

3.2 Access and Build the LaTeX Source

Next, you'll want to get a copy of the source code for this book. (LaTeX sources count as 'code' for these purposes. It gives instructions to machines, it's easy to make mistakes that show up as error messages, and it's reviewed and stored in source control — so it's like code in these practical senses.)

If you already have `git` installed, this will be something like running `git clone https://github.com/dwiddows/eb` at a command prompt, or downloading the code using some visual client software.

List the contents of this directory and check that you can see the files `ebookbook.tex` and `build.sh`. The first of these is the main TeX document that lays out how to combine the other files into an eBook. The second is a build script — on platforms with bash or a compatible shell installed, you should be able to run just `./build.sh` and get most of the book built using one command. Or at least, to begin with, you should get error messages telling you if anything is missing and needs to be installed. If you're not running a compatible shell, the `build.sh` file at least lists the commands you'll need to run some other way.

So the next step is to run `./build.sh` and ideally it should typeset a copy of this book. If the `ebook-convert` command is available it should even make the eBook files described in the next section.

If this doesn't work, you may need to find and install some `pdflatex` and most importantly `make4ht` or `htlatex` programs that work on your machine. (See the Dependencies section above.)

So long as you can run "`make4ht ebookbook`" and create a file `ebookbook.html`, you can go on to the next step.

3.3 From HTML to ePub ... or Mobi

The most important output from the previous step is a file called `ebookbook.html`. This is formatted for display as a webpage in a browser. This is different from the more common use of TeX to make documents such as academic papers, which are nowadays normally created as PDF files. An HTML file is a collection of content (for example, words and images to display), and typesetting suggestions (for example, make this text a heading, and make this image 40% of the screen width). By contrast, a PDF file has precise instructions about how big to make each character and which page to put it on. So it makes sense that HTML is more like an eBook: instead of saying what text will appear on which page, it gives directives about what text should be bigger and smaller, and this combines with the user's device settings to decide which page it should appear on.

So the `ebookbook.html` file (rather than the corresponding PDF file or any other page-layout format) will be used to create your eBook format.

You can turn your HTML file from a webpage into an eBook by installing and using a converter such as Calibre `ebook-convert` or Amazon's *Kindle Previewer* tool. Like saving an image as a `.jpg`, `.gif` or `.png`, you'll need to select a format to convert to. Options include:

- `.epub` is a cross-platform format that it supposed to be used for any electronic book.
- `.mobi` is an old Amazon Kindle format — and it happens to be the one you can use for e-mailing files to your Kindle.

The `build.sh` script that comes with this book uses `ebook-convert` and creates both `.epub` and `.mobi` files as output. This also takes command-line arguments so that you can specify metadata like the author name and cover image:

```
ebook-convert ebookbook.html ebookbook.epub -cover images/cover.jpg -authors "YOUR NAME" -language English
```

3.4 Previewing Your Book on an eReader

Hopefully by now you have an eBook. Or at least, a file called something like `ebookbook.epub`. So how do you *read* your book?

For this you'll need an eReader — perhaps an app on your computer or phone, or an eReader device such as a Kindle.

Assuming that most of your writing will be on a computer, that's probably the first place you'll want to see your work. For example, I usually open the Calibre app and load the `.epub` file, or do both at once with the command `calibre ebookbook.epub`. Once the book is imported and loaded, this gives the result shown in Figure 3.2.

To view on an eReader device or phone, you often have to send it to the device or load it on in some way — again, there are a range of methods. For Amazon Kindles, you need to find / set up an email address for the device itself, and send the book to that email address. As an extra complexity, this method only works using the older `.mobi` files. Also Kindle preloads like this do not (at the moment) display the cover page of the book in your library or menu page, because Amazon retrieves this information for *published* books from elsewhere. So the preview in the library view is somewhat disappointing, but it works (Figure 3.3).

Another way is to sign up for a publishing account and start to submit your book for publication. On the Kindle publishing site, for example, this process includes opportunities to preview your book as it would appear on a phone and a Kindle device, before you set up pricing options and get closer to hitting 'publish'. For this book at least, these online previewers gave a more accurate rendering of how the book will look to readers than emailing a `.mobi` file to my Kindle device.

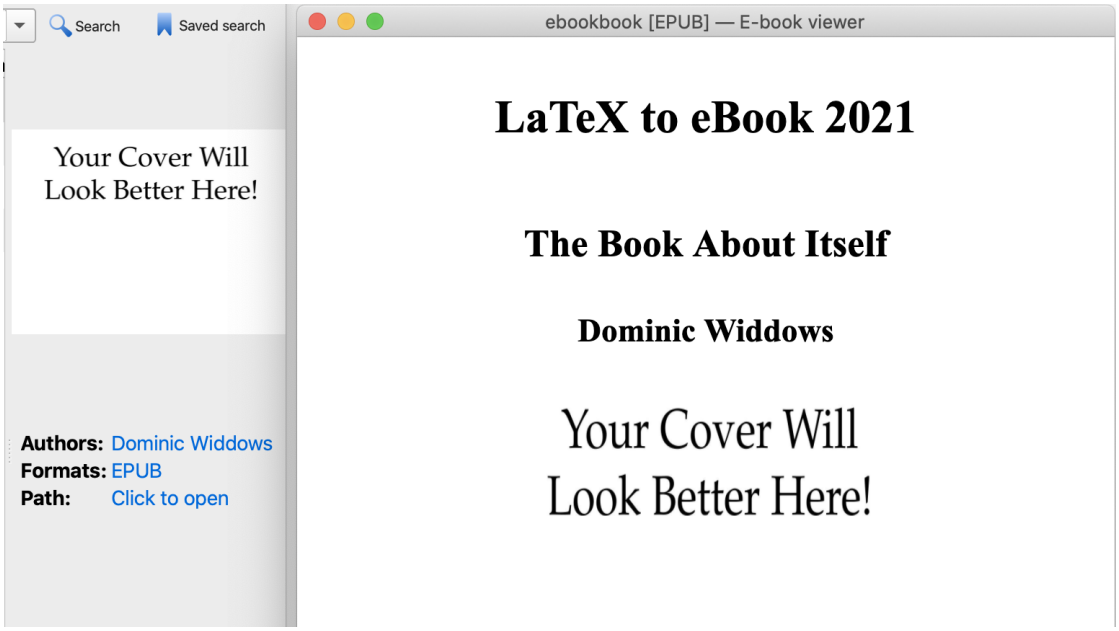


Figure 3.2: Thumbnail image and cover page in Calibre

One way or another, try to make sure you can make an eBook file and view it somewhere.

3.5 Make it Your Own!

Hopefully by now you're at the point where there is an eBook file that you can easily read on some eReader device or app.

The next thing you should do is change something. Change the title in the `cover_page.tex` file from 'LaTeX to eBook 2021' to whatever title you want. Rerun the steps above and hopefully you'll see exactly what you intended: a copy of the eBook, but with your title instead.

At this point you're off to the races. That doesn't mean that it's all plain sailing from here: there will likely be glitches and hurdles along the way. But the main thing is that you have a template, examples of several LaTeX constructs that work with eBooks, and you're able to start turning this into your own book. You may want to save your work separately at this point, call the project something else, and if you use source control, start checking in your work in some way that makes it clear that it's a new project, rather than a work-in-progress on the old project that's intended to be merged back in at some point.

Experiment with removing directives to `\input` different chapters, and watch the book get shorter. Try changing image files to some other graphic and make sure they change appropriately. And try editing the AUTHOR in the `build.sh` file and make sure the

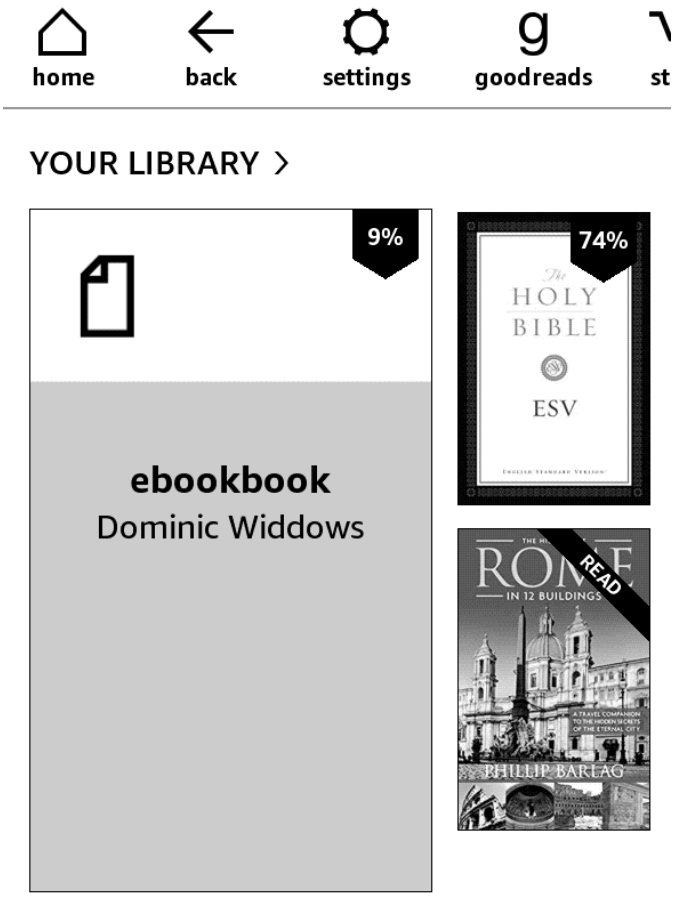


Figure 3.3: Book thumbnail image loaded onto a Kindle

right name shows up when the book is viewed in an eReader.

If this works, you can be pretty confident that your book is on the right track. You'll be able to organize the content into TeX files, image files, etc., and create an eBook!

3.6 Publication

Nearly all of this book is about how to create and preview eBook files, and this is like a software development process — you should be able to run the pipeline over and over again and keep testing that the change you made had the desired effect.

Publishing and marketing your book is a different process: you'll be using an online browser app, signing up for accounts, uploading files, filling in forms, you'll eventually click "submit", and hopefully see your book available for download / purchase after a short while (in the case of this book, a few hours). Look for opportunities to preview your book as part of the submission process.

This process with Amazon Kindle has was straightforward enough form-filling. You can upload your own cover image, or use the online tools to create one. Submission for publication brings up questions like pricing and (related) copyright agreements. To claim a 70% royalty for original work, rather than a 35% royalty, Amazon says you have to price your book between $2.99 and $9.99. Hence the $2.99 price tag for this book.

It takes at least a few hours before your book is live and available — and it may take longer to start appearing in search results.

Note that you can upload new versions of your book any time. Updates (at least minor ones) get automatically processed and released in just the same way as the initial submission. With this book, for example, I submitted two revisions within a few hours, because of course once I was viewing the book 'for real' on my phone, there were a couple of typos and small mistakes I wanted to fix. So in the case of this book, I published it and updated the live version a few times before even telling people it was available.

If you found this book in the Kindle store and can read this as a result, it worked. And if I could do this, you can too!

4 Typesetting Topics

This chapter will go through some of the TeX features you'll probably want to use at some point. So far, *most* of the things I usually do with TeX can be made to work for eBook outputs, but there are lots of commands and options that don't work and you need to know which variants to use.

4.1 References and Citations

One of the easy things that 'just works' most of the time in LaTeX is references. For example, the TeX source for this chapter starts with:

```
\chapter{Typesetting Topics}
\label{chapter:typesetting}
```

Then if I want to refer to the chapter, I write `Chapter \ref{chapter:typesetting}` which gets rendered as 'Chapter 4'. Adding `label` commands for tables and figures works in the same way.

This isn't rocket science, but if you've ever tried renumbering by-hand, you know how valuable this is! Similarly, bibliographic references such as `\cite{knuth1984texbook}` work correctly, giving a citation looking like Knuth and Bibby (1984). So far BibTeX has worked fine for this book.

4.2 Graphics

Nearly all eBooks have graphics in them somewhere, even if just for a cover page. The `graphicx` package works well for this. For example, the cover image for the title page of this book is included using the command:

```
\includegraphics[width=0.8\linewidth]{images/cover.png}
```

Typically for webpages and eBooks, the size of images is expected to vary with the page size and settings, so using a context-sensitive width like a proportion of the linewidth is more appropriate than a fixed-width declaration like '10cm'. Many eReaders enable users to click on images to see them in more detail, which helps.

One frequent problem is that HTML typesetting using `make4ht` or a similar process doesn't always preserve the aspect ratio of your images. For example, the same setting may be used for width and height, making all images square. The problem is solved by adding a bounding box file using the `extractbb` command, which is typically included

Figure 4.1: Map of countries in Southeast Asia

with TeX distributions. For this book, this step is included in the project's `build.sh` script, at least for `.jpg` and `.png` files, and it can easily be extended to more filetypes. Or you can run `extractbb -x` on these files yourself.

Images are often put in the LaTeX `figure` environment, which can include captions and a label for references. For example, the map in Figure 4.1 is created using the commands:

```
\begin{figure}
\begin{center}
  \includegraphics[width=\linewidth]{images/sea_countries.png}
  \htbr
  \caption{Map of countries in Southeast Asia}
  \label{fig:sea_map}
\end{center}
\end{figure}
```

Note the `\htbr` command, which is defined in the macros as `\ifdefined\HCode{\HCode{<br/><br/>}}\`

Artist	Great Works
Leonardo da Vinci	The Mona Lisa
Charlie Watts	Satisfaction

Table 4.1: Inspiring works

This is an extra TeX command for hypertext: it forces some vertical space to be included between the image and the caption, because the usual TeX command `vspace` seems to have no effect on the HTML (and hence ePub) output here.

From here, the figure can be referenced using the command `\ref{fig:sea_map}`. (Made using `https://github.com/dwiddows/pilmaps`, a free mapping tool in python that supports low-level, hands-on rendering control — feel free to try it out.) So far I haven't found an effective way of controlling the placement of figures — for example, directives like `\begin[ht]{figure}` don't affect the HTML or the ePub output, and I haven't got wraparound text to work.

4.3 Tables

Basic tables typeset just fine in `.epub` format. They don't format properly in the Kindle preview of `.mobi` files sent via email, but look better in the Kindle `.epub` preview.

For example, the following LaTeX gives the output in Table 4.1.

```
\begin{table}
  \centering
  \begin{tabular}{|c|l|}
    \hline
    \textbf{Artist} & \textbf{Great Works} \\
    \hline
    Leonardo da Vinci & The Mona Lisa \\
    Charlie Watts & Satisfaction \\
    \hline
  \end{tabular}
  \caption{Inspiring works}
  \label{tab:artist_works}
\end{table}
```

With standard LaTeX it is easy to make tables that are too big for pages or that typeset poorly for other of reasons. I expect this can be even more of a problem with small-screen eBooks, so you'll want to design any tables accordingly and check output carefully.

4.4 Equations

One of the benefits of LaTeX is that it's easy to typeset mathematical notation like formulae and equations. So far these look to work well in HTML and eBook formats.

For example, here is the TeX and its rendering for Euler's formula:

```
\[ e^{ix} = \cos x + i \sin x \]
```

$$e^{i\pi} = -1$$

And here is the corresponding TeX and output for a Fourier series:

```
\[ f(x) = \sum_{k=0}^{\infty} [  a_k \sin(kx) + b_k \cos(kx) ] \]
```

$$f(x) = \sum_{k=0}^{\infty} [a_k \sin(kx) + b_k \cos(kx)]$$

Array environments for typesetting rows and columns in equations also work, for example:

$$u = \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix} \qquad v = \begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix} \qquad u^T v = 2+0-6 = -4 \qquad uv^T = \begin{pmatrix} 2 & -1 & 3 \\ 0 & 0 & 0 \\ -4 & 2 & -6 \end{pmatrix}$$

This set of equations may become typeset in a smaller font than those above, to fit them horizontally on a small screen. If these start to get too small, consider breaking lines up when typesetting mathematics for eBooks.

4.5 Different Languages

It's possible (and sometimes easy) to typeset documents using languages other than English and character sets other than the Latin alphabet.

For example, this is in Russian: Это на русском.

This is in Greek:

Αυτό είναι στα ελληνικά.

These examples are made using the `babel` package, and by setting encoding instructions in the initial document setup (see the `lib/packages.tex` file), and then in the LaTeX source writing something like:

```
This is in Greek:
\selectlanguage{greek}
( add your text here )

\selectlanguage{english}
This is back in English.
```

I have not got Chinese or Thai working properly yet. There is an example file in `tests/languages.tex` showing this.

4.6 Contents and Index

The table of contents should get typeset automatically if you use this template.

The story of how traditional indexes and concordances influenced the design of the inverted indexes used by search engines is fascinating (Witten et al., 1999, Ch 1). For electronic books, this process is quite complete: the index is created automatically, and accessed via the search interface, not by the user scrolling through topics in alphabetical order. Therefore I'm not including an explicit index section.

If you want to make a paper version of your book as well as an eBook, you may want to make an alphabetical index. In this case, it may be worth trying a more sophisticated template such as the one that comes with Clemens Lode's book on using LaTeX for books and eBooks (Lode, 2019). This includes conditional compilation so that different commands and even sections are used for the PDF version that leads to a paper book and the HTML version that leads to an eBook.

4.7 Difficulties

Several things have proved tricky and may just not work with typesetting to HTML.

Exact placement is sometimes impossible. Figures, tables, and captions may appear in various places, and often don't obey TeX typesetting directives. For example, putting an `fbox` (framebox) around a figure makes it float to the left, irrespective of centering commands. Captions may easily fall across a page boundary, so one tip is to keep captions small (one line), just enough for the reader of the main text to see that they are looking at the right table or image.

Controlling font size in different environments has been hard, particularly with typewriter fonts used for verbatim text such as rendering LaTeX commands that you don't want to be interpreted as LaTeX commands. In retrospect, trying to write an eBook about LaTeX itself was perhaps not the most sensible decision! But this has been fixed using the right combination of commands from the `fancyvrb` package, included here as macros like `\sverb` for 'small verbatim' and `\surl` for 'small URL'.

Some of these things lead to useful design considerations. Remember that we're designing for a small screen, whose font size and spacing is up to the user. Don't struggle to make a particular page look perfect on your device only to see later that for someone with a larger font setting, this content is broken across several pages. Try to make your book design clear, legible, and somewhat flexible, and a huge variety of readers will be able to enjoy it!

5 Maintenance and Troubleshooting

We're nearly done with this short template book — at least with the first version. But as with most topics to do with software and electronic information, things will keep changing. Bugs will hopefully get fixed, new bugs will arise, different devices will support different formats, standards may change — for example, it may become possible to load `.epub` files directly onto an Amazon Kindle, rather than having to use the `.mobi` format solely for this purpose. This last chapter will include a few suggestions on how to ask questions and report changes.

5.1 Report Bugs on Github

The open source github repository for this project is `https://github.com/dwiddows/ebookbook/`. If something recommended in this book doesn't work for you, and if you have to change or add extra commands to make it work, please report this as an issue there. The github project wiki can be used for keeping instructions up-to-date and adding links to more resources. This is likely to be much more effective the reporting problems on sites where the book is sold. If you want to review the book itself, post a review. But if you want to report a problem and get help, please use the project github site.

When there are major developments or enough new things to include, I will add these to future editions of this book (if there are any!). But by design, books are meant to be relatively stable, whereas project websites and wikis are designed to have immediate updates. If a new 'edition' comes out too often, we can lose track of what to refer to.

So tl;dr: please leave actual reviews on marketplace sites, but report bugs on github. Not just because I want to avoid a bunch of frustrated bug reports as book reviews, but also because if bugs are reported on the github project, they're more likely to be addressed and fixed!

5.2 That's All For Now

This was always intended to be a short book, just enough to provide a working template, to demonstrate a few typesetting features, and to make the book whose end you're just reaching.

I hope it's given you some confidence to get started, and in particular, I hope the structure works for you. Clone the project, install a few dependencies, try it out, and ideally within a short time you won't be thinking much about this book — it will already be becoming *your* book.

Good luck and enjoy your writing!

Dominic Widdows
September 2021

About the Author

Dominic Widdows is a mathematician, computational linguist, and software engineer. He has worked on differential geometry and Oxford, natural language processing at Stanford, and many projects at MAYA Design, Google, Microsoft, Grab, and LivePerson, where he works particularly on conversational AI and internationalization.

As an author, his work includes the book Geometry and Meaning, and over 50 scientific papers, in areas including pure mathematics, computer science, language processing, bioinformatics, information extraction, logistics, and quantum computing. As a developer, his open source contributions include work on the SkyMap Planetarium App, the SemanticVectors project, and PILMaps for drawing maps.

Enthusiasm for scientific publishing and open source development combined to make this book.

Bibliography

Knuth, D. E. and Bibby, D. (1984). *The TeXbook*, volume 15. Addison-Wesley Reading.

Lamport, L. (1985). *LaTeX: A Document Preparation System*, volume 410. Addison-Wesley Professional; 2nd edition (June 30, 1994).

Lode, C. (2019). *Better Books with LaTeX: Self-Publish Your Book on Amazon and Google*. Clemens Lode Verlag eK.

Witten, I. H., Moffat, A., and Bell, T. C. (1999). *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, 2 edition.