
Books in L^AT_EX

Since the introduction of L^AT_EX, the visual quality of articles published in mathematical journals has improved dramatically. Unfortunately, the same cannot be said of books published using L^AT_EX. A record number of very ugly books have appeared.

It is easy to understand why. While `amsart` has been designed to produce high-quality printed output, the standard book document classes do not produce attractive books without additional work.

L^AT_EX provides the `book` and the `amsbook` document classes to serve as foundations for well-designed books. Better quality books have to use document classes designed by professionals. We provide some sample pages from a book using Springer's `svmono.cls` document class. We briefly discuss logical and visual design in Section 4.3.2.

So this chapter is not about how to produce a finished book using L^AT_EX. Our goal is much more modest, how to prepare a book manuscript for your publisher. In Section 18.1 we describe the book document classes `book` and `amsbook`. The table of contents and lists of figures and tables are discussed in Section 18.2.

Typesetting a book involves dozens of files. Section 18.3.3 gives some tips on how to organize them. Section 18.4 covers logical design. Section 18.5 deals with the final preparation of your edited manuscript for your publisher. Finally, Section 18.6

suggests a few more things to do if you typeset your book yourself.

18.1 Book document classes

In this section, we briefly discuss the way in which book and amsbook, the two standard book document classes, differ from the corresponding article document classes. We also show a few pages from a book that was typeset with Springer Verlag's document class for monographs: `svmono.cls`.

18.1.1 Sectioning

Book document classes have chapters, invoked with the `\chapter` command and *parts*, invoked with `\part`. The `\part` command is generally used to group chapters in longer documents, for instance in this book. Parts have no effect on the numbering of chapters, sections, and so on, so Chapter 1 of Part I is not numbered as I.1 but as 1.

Both `\chapter` and `\part` take a title as an argument, but the `\chapter` command also has an optional argument:

```
\chapter[short_title]{title}
```

The optional *short_title* argument is used in the running head. You may need to protect any fragile commands in *title* and *short_title* with the `\protect` command (see Section 5.3.3).

Here is the whole hierarchy:

```
\part

\chapter
  \section
    \subsection
      \subsubsection
        \paragraph
          \subparagraph
```

Book document classes, as a rule, do not number subsubsections or any of the sectioning divisions below that level.

Equations in chapters

By default, equations are numbered from 1 within chapters. So in Chapter 1 as well as in Chapter 3, the equations are numbered (1), (2), and so forth. If you have the

```
\numberwithin{equation}{chapter}
```

command in the preamble, then equations in Chapter 2 are numbered as (2.1), (2.2), and so on.

18.1.2 Division of the body

The book document classes formalize the division of the body into three parts.

Front matter The material that appears in the front of the document, including the title pages (normally four), table of contents, preface, introduction, and so on. \LaTeX numbers these pages using roman numerals. The front matter is introduced with the `\frontmatter` command.

Main matter The main part of the book, including the appendices if any. Page numbering starts from 1 using arabic numerals. The main matter is introduced with the `\mainmatter` command.

Back matter Material that appears in the back of the book, including the bibliography, index, and various other sections, such as the colophon, afterword, and so on. The back matter is introduced with the `\backmatter` command.

For the book document class—and the document classes built on it—in the front and back matter, the `\chapter` command does not produce a chapter number but the title is listed in the table of contents. So you can start your introduction with

```
\chapter{Introduction}
```

Within such a chapter, you should use the *-ed forms of the sectioning commands `\section`, `\subsection`, and so on, otherwise you have sections with numbers such as 0.1.

In the main matter, the `\appendix` command marks the beginning of the appendices. Each subsequent chapter becomes a new appendix. For example,

```
\appendix
\chapter{A proof of the Main Theorem}\label{A:Mainproof}
```

produces an appendix with the given title.

Note that appendices may be labeled and cross-referenced. In Appendix A, sections are numbered A.1, A.2, and so on, subsections in A.1 are numbered A.1.1, A.1.2, and so on. The precise form these numbers take depends, of course, on the document class, packages, and user-specific changes (see Section 15.5.1).

See Section 18.3.1 for a detailed example.

For the `amsbook` document class—and the document classes built on it—the `\chapter` and `\chapter*` commands always produce a title listed in the table of contents.

The following two questions are frequently asked:

My book has only one appendix. How can I get it to be called just “Appendix”, not “Appendix A”?

The single appendix in my book is being labeled “Appendix A”. How can I change this to just “Appendix”? This appendix has a title, so the answer to the preceding question doesn’t apply.

These questions are answered in the author FAQ of the AMS, go to

<http://www.ams.org/authors/author-faq.html>

18.1.3 Document class options

The options and defaults for the book document classes are the same as those of other document classes (see Sections 11.5 and 12.1.2) with a few exceptions.

Two-sided printing

Options: twoside *default*
 oneside

The twoside option formats the output for printing on both sides of a page.

Titlepage

Options: titlepage *default*
 notitlepage

The titlepage option creates a separate title page. The notitlepage option creates no separate pages.

Chapter start

Options: openright *default*
 openany

A chapter always starts on a new page.

The book document class—and the document classes built on it—use the option openright to start each chapter on an odd page, while the option openany starts each chapter on the first available new page. If you use the default option, end each chapter with the command

```
\cleardoublepage
```

Then if a chapter ends on an odd page, a blank page is added with no header or page number. The \cleardoublepage command is correctly coded if you use amsbook. Otherwise, use the package cleardoublepage.sty (in the samples folder).

The amsbook document class—and the document classes built on it—automatically clears to a right-hand page and leaves a totally blank page if needed.

18.1.4 Title pages

The book document class supports the commands: `\title`, `\author`, `\date`, and `\maketitle` (see Section 12.1.1). The `amsbook` document class supports the same commands as `amsart` (see Section 11.2).

You can design your own title page within the `titlepage` environment, which does not require the use of the `\maketitle` command. Title pages for books, of course, should be created by a book designer for the publisher.

18.1.5 Springer's document class for monographs

We choose `svmono.cls`, Springer's document class for monographs, to demonstrate the sophisticated appearance of a book typeset with a professionally designed document class. On the next three pages, I display the title page, the first Table of Contents page, and the first page of Chapter 1 of Claudio Procesi's book on Lie groups.

Claudio Procesi

Università di Roma La Sapienza

Lie Groups

An Approach through Invariants and Representations



Contents

Introduction	xix
Conventional Notations	xxiii
1 General Methods and Ideas	1
1 Groups and Their Actions	1
1.1 Symmetric Group	1
1.2 Group Actions	2
2 Orbits, Invariants and Equivariant Maps	4
2.1 Orbits	4
2.2 Stabilizer	5
2.3 Invariants	7
2.4 Basic Constructions	7
2.5 Permutation Representations	8
2.6 Invariant Functions	10
2.7 Commuting Actions	11
3 Linear Actions, Groups of Automorphisms, Commuting Groups	11
3.1 Linear Actions	11
3.2 The Group Algebra	13
3.3 Actions on Polynomials	15
3.4 Invariant Polynomials	16
3.5 Commuting Linear Actions	17
2 Symmetric Functions	19
1 Symmetric Functions	19
1.1 Elementary Symmetric Functions	19
1.2 Symmetric Polynomials	21
2 Resultant, Discriminant, Bézoutiant	22
2.1 Polynomials and Roots	22

1

General Methods and Ideas

Summary. In this chapter we will develop the formal language and some general methods and theorems. To some extent the reader is advised not to read it too systematically since most of the interesting examples will appear only in the next chapters. The exposition here is quite far from the classical point of view since we are forced to establish the language in a rather thin general setting. Hopefully this will be repaid in the chapters in which we will treat the interesting results of Invariant Theory.

1 Groups and Their Actions

1.1 Symmetric Group

In our treatment groups will always appear as transformation groups, the main point being that, given a set X , the set of all bijective mappings of X into X is a group under composition. We will denote this group $S(X)$ and call it *the symmetric group* of X .

In practice, the full symmetric group is used only for X a finite set. In this case it is usually more convenient to identify X with the discrete interval $\{1, \dots, n\}$ formed by the first n integers (for a given value of n). The corresponding symmetric group has $n!$ elements and it is denoted by S_n . Its elements are called *permutations*.

In general, the groups which appear are subgroups of the full symmetric group, defined by special properties of the set X arising from some extra structure (such as from a topology or the structure of a linear space, etc.). The groups of interest to us will usually be symmetry groups of the structure under consideration. To illustrate this concept we start with a definition:

Definition. A partition of a set X is a family of nonempty disjoint subsets A_i such that $X = \cup_i A_i$.

A partition of a number n is a (non-increasing) sequence of positive numbers:

$$m_1 \geq m_2 \geq \dots \geq m_k > 0 \text{ with } \sum_{j=1}^k m_j = n.$$

18.2 Tables of contents, lists of tables and figures

A long document, as a rule, has a table of contents. It may also include a list of figures and a list of tables.

18.2.1 Tables of contents

What goes into the table of contents?

For the `amsbook` document class—and the document classes built on it—all titles, not the short titles, of the sectioning commands, whether *-ed or not, subject only to the value of the `tocdepth` counter, as described in the last subsection of Section 15.5.1. For instance, if `tocdepth` is set to 2, the default, then the titles of chapters, sections, and subsections are included in the table of contents, and subsubsections are excluded.

This leaves us with the problem, what do we do if the title is too long? You cannot break the line with `\\`, because this would appear in table of contents. The AMS coded the following solution: enter the line break in the form

```
\except{toc}{\linebreak}
```

For the `book` document class—and the document classes built on it—the title or optional argument of the sectioning commands, subject to the value of the `tocdepth` counter, with the following exceptions:

- In Section 10.4.1 we discuss the *-ed versions of sectioning commands. They are excluded from the table of contents.
- If the sectioning command has a short title, then it is the short title that is utilized.

The example in Section 11.2 shows why this is important. If you have `\\` in the title, you must have a short title without it, otherwise the `linebreak` would show up in the running head and the table of contents.

When you typeset your document with a table of contents, \LaTeX creates a file with the `toc` extension. The next time the document is typeset, the `toc` file is typeset too and included in your typeset document at the point where the command

```
\tableofcontents
```

appears in the source file, normally in the front matter. If your source file is named `myart.tex`, the `toc` file is named `myart.toc`. This file lists all the sectioning units as well as their titles and page numbers.

If you already have a `toc` file, the `\tableofcontents` command typesets a table of contents using the previously created `toc` file and creates a new `toc` file.

\LaTeX adds a line to the table of contents, formatted like a section title, if you include the command

```
\addcontentsline{toc}{section}{text_to_be_added}
```

in your source file. There are three arguments:

1. The first argument informs L^AT_EX that a line, the third argument, should be added to the toc file.
2. The second argument specifies how the line should be formatted in the table of contents. In our example, the second argument is `section`, so the line is formatted as a section title in the table of contents. The second argument must be the name of a sectioning command.
3. The third argument is the text to be added.

You can add an unformatted line to the table of contents with the command

```
\addtocontents{toc}{text_to_be_added}
```

Such a command can also be used to add vertical spaces into the table of contents. For instance, if you want to add some vertical space before a part, you should insert the following line before the sectioning command for the part:

```
\addtocontents{toc}{\protect\vspace{10pt}}
```

The toc file is easy to read. The following are typical lines from the table of contents file for a document using the book document class:

```
\contentsline{section}{\numberline {5-4.}Top matter}{119}
\contentsline{subsection}{\numberline {5-4.1.}
Article info}{119}
\contentsline {subsection}{\numberline {5-4.2.}
Author info}{121}
```

Section 15.5.1 explains how you can specify which levels of sectioning appear in the table of contents. Section 2.3 of *The L^AT_EX Companion*, 2nd edition [46] lists the style parameters for the table of contents. It also shows you how to define new toc-like files and use multiple tables of contents in a single document, for instance, adding a mini table of contents for each chapter.

Tip You may have to typeset the document three times to create the table of contents and set the numbering of the rest of the document right.

1. The first typesetting creates the toc file.
2. The second inserts the table of contents with the old page numbers into the typeset document, re-records in the aux file the page numbers, which may have changed as a result of the insertion, and cross-references in the aux file, and generates a new toc file with the correct page numbers.

3. The third typesetting uses these new aux and toc files to typeset the document correctly and creates a new toc file.

Fragile commands in a movable argument, such as a section (short) title, must be `\protect`-ed (see Section 5.3.3). Here is a simple example using the table of contents. If the document contains the `\section` command

```
\section{The function \(\ f(x^{2}) \)}
```

the section title is stored in the toc file as

```
\contentsline {section}{\numberline
{1}The function\relax $ f(x^{2}) \relax \GenericError { }
{LaTeX Error: Bad math environment delimiter}{Your
command was ignored.\MessageBreak Type I <command>
<return> to replace it with another command,\MessageBreak
or <return> to continue without it.}}{1}
```

and the log file contains the message

```
! LaTeX Error: Bad math environment delimiter.
```

```
...
1.1 ...continue without it.}}{1}
```

Error messages usually refer to a line in the source file, but in this case the error message refers to a line in the toc file.

The correct form for this section title is

```
\section{The function \protect\(\ f(x^{2}) \protect\)}
```

or, even simpler,

```
\section{The function $f(x^{2})$}
```

Note that this example is merely an illustration of unprotected fragile commands in movable arguments. As a rule, avoid using formulas in (sectioning) titles.

18.2.2 Lists of tables and figures

If you place a `\listoftables` command in the document, \LaTeX stores information for the list of tables in a `lot` file. The list of tables is inserted into the body of your document at the point where the command appears, normally in the front matter, following the table of contents.

A list of figures, similar to a list of tables, can be compiled with the command `\listoffigures`. This command creates an auxiliary file with the extension `lof`.

An optional argument of the `\caption` commands in your tables and figures can replace the argument in the list of tables and figures. Typically, the optional argument is used to specify a shorter caption for the list of tables or list of figures. There are other uses. For instance, you may notice that, as a rule, captions should be terminated by periods. If in the list of tables or list of figures, your book style fills the space between the text and the page number with dots, the extra period looks bad. This problem goes away if you use the following form of the `\caption` command:

```
\caption[title]{title.}
```

There are analogs of the table of contents commands for use with tables and figures. The command

```
\addtocontents{lot}{line_to_add}
```

adds a line to the list of tables or to the list of figures with the first argument `lof`.

18.2.3 Exercises

For exercises, `amsbook` provides the `xcb` environment. It is used for a series of exercises at the end of a section or chapter. The argument of the environment specifies the phrase (such as `Exercises`) to begin the list:

```
\begin{xcb}{Exercises}
\begin{enumerate}
\item A finite lattice  $L$  is modular iff it does not
  contain a pentagon.\label{E:pent}
\item Can the numbers of covering pairs in\label{E:incr}
  Exercise~\ref{E:pent} be increased?\label{E:incr}
\end{enumerate}
\end{xcb}
```

which typesets as

Exercises

- (1) A finite lattice L is modular iff it does not contain a pentagon.
- (2) Can the numbers of covering pairs in Exercise 1 be increased?

18.3 Organizing the files for a book

An article is typically one `tex` file and maybe some EPS and/or PDF files for the illustrations. On the other hand, a book, like this one, is composed of hundreds of files. In this section, I describe how the files for this book are organized.

There are three commands that help with the organization:

```
\include \includeonly \graphicspath
```

We discuss these commands in this section.

18.3.1 *The folders and the master document*

All the files for this book are in a folder `MiL4` and in this folder the most important document is `MiL4.tex`, the *master document*.

The master document, `MiL4.tex` reads, in a somewhat simplified form, as follows:

```
%MiL4 master document
\documentclass[leqno]{book}
\usepackage{MiL4}
\usepackage{makeidx}
\makeindex
\usepackage{cleardoublepage}
\includeonly{
%frontmatter,
%intro,
%Chapter1,% terminology
...
Chapter18,% books
%appA,% install
...
}
\begin{document}
\frontmatter
\include{frontmatter}
\tableofcontents
\listoftables\listoffigures
\include{intro}%Intro
\mainmatter
\include{Chapter1}%Setting up
...
\include{Chapter18}% writing books

\appendix
\include{AppendixA}%install
...
\backmatter
\printindex
\end{document}
```

Some parts of the master file deserve comment, for example, the third line,

```
\usepackage{MiL4}
```

loads the command file

```
MiL4.sty
```

which contains all the commands defined for the book and the code for the book style. Since the book style is based on `book.cls`, in line 5, we load the `makeidx` package and print the index with `\printindex` (see Section 17.1).

Line 7 states

```
\usepackage{cleardoublepage}
```

This creates blank pages after chapters that end on an odd page number (see Section 18.1.3). For the three `...matter` commands, see Section 18.1.2.

18.3.2 *Inclusion and selective inclusion*

This book is pieced together by the `\include` commands in the master document. For example,

```
\include{Chapter18}
```

inserts the contents of the file `Chapter18.tex`, starting on a new page, as though its contents had been typed at that place in the document. The master document for this book has 29 `\include` commands.

Rule ■ File termination

Terminate every file you `\include` with an `\endinput` command.

If you terminate an `\include`-ed file with `\end{document}`, L^AT_EX gives a warning such as:

```
(\end occurred when \iftrue on line 6 was incomplete)
(\end occurred when \ifnum on line 6 was incomplete)
```

If you use `\include` commands in the master file, as in the example in Section 18.3.1, then you can use the `\includeonly` command for selective inclusion. The lines of the `\includeonly` command parallel the `\include` commands. Block comment all the lines of the argument of the `\includeonly` command, and uncomment the chapter you are working on. In the example above, I am working on this chapter.

The argument of the `\includeonly` command is a list of files separated by commas. If you want to typeset the whole book, uncomment all the lines.

18.3.3 Organizing your files

The MiL4 folder, containing the files of this book, contains the master document, `MiL4.tex`, the command file `MiL4.sty`, and all the `tex` files listed in the master document, that is, the chapters, the frontmatter, the introduction, the appendices, and of course, all the auxiliary files that `LATEX` creates.

This book contains about 300 illustrations in a subfolder `Graphics` of the folder `MiL4.tex`. We have to tell `LATEX` to look for the illustrations in this folder. We do this with the command

```
\graphicspath{{Graphics\}}
```

in the preamble. If you have two folders, `Illustr1` and `Illustr2` for illustrations, the `\includegraphics` command takes the form

```
\graphicspath{{Illustr1\}{Illustr2\}}
```

Even if you have more than one folder for the illustrations, you must make sure that each graphics file has a unique name.

We place the `\graphicspath` command in `MiL4.sty`.

In the above commands, `\` is appropriate for Mac and UNIX computers. For a PC, use `/` instead.

If you submit a `dvi` file, you cannot use the `\graphicspath` command.

18.4 Logical design

The discussion of logical and visual design in Section 4.3.2 applies to books even more than to articles. Since books are long and complex documents, errors in the logical design are much harder to correct.

Let us review some common-sense rules.

Rule 1 ■ Stick with the sectioning commands provided by the document class. Define the non-standard structures you wish to use as environments.

Here is an example which is obviously bad:

```
\vspace{18pt}
```

```
\noindent \textbf{Theorem 1.1.}
```

```
\textit{This is bad.}
```

```
\vspace{18pt}
```

And a good way to achieve the same result:

```
\begin{theorem}\label{T:Goodtheorem}
This is a good theorem.
\end{theorem}
```

The bad example creates a number of difficulties.

- You have to number the theorems yourself. Adding, deleting, and rearranging theorems becomes difficult and updating cross-references is even harder.
- It is difficult to keep such constructs consistent.
- If the publisher decides to increase the white space before and after the theorems to 20 points, finding and changing all the appropriate commands becomes a tedious and error prone task.

Rule 2 ■ Define frequently used constructs as commands.

Rather than

```
\textbf{Warning! Do not exceed this amount!}
```

define

```
\newcommand{\important}[1]{\textbf{#1}}
```

and type your warnings as

```
\important{Warning! Do not exceed this amount!}
```

You or your editor can then change all the warnings to a different style with ease.

Rule 3 ■ Avoid text style commands.

If you use small caps for acronyms, do not type

```
\textsc{ibm}
```

but rather define

```
\newcommand{\ibm}{\textsc{ibm}}
```

and then


```
\ibm
```

or more generally

```
\newcommand{\acronym}[1]{\textsc{#1}}
```

and then

```
\acronym{ibm}
```

Rule 4 ■ Avoid white space commands.

Occasionally, you may feel that there should be some white space separating two paragraphs, so you do the following:

```
paragraph 1
```

```
\medskip
```

```
paragraph 2
```

It would be better to define a new command, say `\separate`, as

```
\newcommand{\separate}{\medskip}
```

and type the previous example as

```
paragraph 1
```

```
\separate
```

```
paragraph 2
```

Now such white space can be adjusted throughout the entire document by simply redefining one command. Note that redefining `\medskip` itself may have unintended side effects:

- Many environments depend on L^AT_EX's definition of `\medskip`.
- You may have used `\medskip` in other situations as well.

Here is a short list of commands you should avoid:

<code>\bigskip</code>	<code>\hfil</code>	<code>\hspace</code>	<code>\parskip</code>	<code>\vfill</code>	<code>\vspace</code>
<code>\break</code>	<code>\hfill</code>	<code>\kern</code>	<code>\smallskip</code>	<code>\vglue</code>	
<code>\eject</code>	<code>\hglue</code>	<code>\medskip</code>	<code>\vfil</code>	<code>\vskip</code>	

18.5 *Final preparations for the publisher*

Throughout this book, there are a number of “don’ts”. Most are practices you should avoid while writing articles. When writing a book, it is even more important not to violate these rules.

When the editors, including the copy editor, are finished with your manuscript and you have the document class designed for the book, then you can start on the final preparations.

Step 1 ■ Eliminate all T_EX commands.

T_EX commands, that is, Plain T_EX commands that are not part of L^AT_EX (not listed as L^AT_EX commands in the index of this book) may interfere with L^AT_EX in unexpected ways. They may also cause problems with the style file that is created for your book.

Step 2 ■ Collect all your user-defined commands and environments together in one place, preferably in a separate command file (see Section 15.3).

Step 3 ■ Make sure that user-defined commands for notations and user-defined environments for structures are used consistently throughout your document.

This book uses the command `\doc` for document names, so `intrart` is typed as `\doc{intrart}`. Of course, `\texttt{intrart}` gives the same result, but if you intermix `\doc{intrart}` and `\texttt{intrart}` commands, you lose the ability to easily change the way document names are displayed.

Step 4 ■ Watch out for vertical white space adding up.

- Do not directly follow one displayed math environment with another. Multiple adjacent lines of displayed mathematics should all be in the same environment.
 - If your style file uses interparagraph spacing, avoid beginning paragraphs with displayed math.
-

For instance,

```
\[
    x=y
\]
```

$$\begin{array}{c} \backslash[\\ \quad x=z \\ \backslash] \end{array}$$

is wrong. Use, instead, an `align` or `gather` environment.

Step 5 ■ If possible, do not place “tall” mathematical formulas inline. All formulas that might change the interline spacing, as a rule, should be displayed.

You can find examples on pages 23 and 370. Here is one more example, double hat accents used inline: $\hat{\hat{A}}$.

Step 6 ■ Read the `log` file.

- Watch for line-too-wide warnings (see Section 2.3).
 - Check for font substitutions (see Section 5.6.7).
-

If you find lines that are too wide:

- Fix wide lines by rewording the sentence or adding optional hyphens (see Section 5.4.9).
- Break displayed formulas so that they fit comfortably within the line.

Adobe Acrobat Professional has a preflight utility in the Advanced menu. It will check whether the PDF version of your typeset document has all the fonts it requires.

Step 7 ■ Do not assume that gray boxes or color illustrations appear when published exactly the way that they look on your monitor or printer.

Color work requires calibration of monitors and printers. It is often best left to the experts at the publisher.

Step 8 ■ Do not assume that the application that created your EPS files (see Section 10.4.3) can create high-quality EPS files.

Many applications can create EPS files or convert files to EPS format. Very few do it right. Ask your publisher what applications they recommend.

Font substitutions can also cause problems:

- A font that was used in typesetting the document may not be the font you intended. Missing fonts are substituted and the substitute fonts are rarely satisfactory.
- A special trap: Your publisher may have more, or maybe fewer, fonts than you do! As a result, the font substitutions on your publisher's system may be different from those on yours. Make sure that the fonts you use are not substituted.

18.6 *If you create the PDF file for your book*

Many publishers take your manuscript, prepared as described in Section 18.5, and guide it through the final steps for printing. Some books, however, are prepared by the authors for printing using a custom document class for books and submitted to the publisher as PDF files. If your book falls into this category, there are a few more things you should do before you create the final PDF file for your book.

Adjust the pages

Make sure that you are satisfied with the way the document is broken into pages by L^AT_EX and with the placement of the `figure` and `table` environments (see Section 10.4.3). If necessary, you should make last-minute changes to adjust page breaks. You may find the `\enlargethispage` command (see Section 5.7.3) very helpful at this stage. Just be sure to apply it on both facing pages.

To ensure that

- Page numbers in the index are correct
- `\pageref` references (see Section 10.4.2) are correct
- Marginal comments (see Section 5.9.4) are properly placed
- Tables and figures are properly placed

insert page breaks where necessary. Where pages break, add the three commands `\linebreak`, `\pagebreak`, and `\noindent`. Here is an example. The bottom of page 3 and the top of page 4 of my book *General Lattice Theory* [28] are shown in Figure 18.1.

Now let us assume that we have to manually do this page break because some index items attached to this paragraph generate incorrect page numbers. The paragraph split by the page break is

In other words, lattice theory singles out a special type of poset for detailed investigation. To make such a definition worthwhile, it must be shown that this class of posets is a very useful class, that there are many

such posets in various branches of mathematics (analysis, topology, logic, algebra, geometry, and so on), and that a general study of these posets will lead to a better understanding of the behavior of the examples. This was done in the first edition of G.~Birkhoff's `\emph{Lattice Theory} \cite{gB40}`. As we go along, we shall see many examples, most of them in the exercises. For a general survey of lattices in mathematics, see G.~Birkhoff `\cite{gB67}` and H.~H.~Crapo and G.-C.~Rota `\cite{CR70}`.

When typesetting this paragraph, L^AT_EX inserts a page break following

This was done in the first edition of G.~Birkhoff's.

So we edit four lines as follows:

```
understanding of the behavior of the examples.
This was done in the first edition of G.~Birkhoff's
\linebreak
```

```
\pagebreak
```

```
\noindent \emph{Lattice Theory} \cite{gB40}. As we go
along, we shall see many examples, most of them in the
```

This change does not affect the appearance of the typeset page, but now pages 3 and 4

In other words, lattice theory singles out a special type of poset for detailed investigation. To make such a definition worthwhile, it must be shown that this class of posets is a very useful class, that there are many such posets in various branches of mathematics (analysis, topology, logic, algebra, geometry, and so on), and that a general study of these posets will lead to a better understanding of the behavior of the examples. This was done in the first edition of G. Birkhoff's

4 I. First Concepts

Lattice Theory [1940]. As we go along, we shall see many examples, most of them in the exercises. For a general survey of lattices in mathematics, see G. Birkhoff [1967] and H. H. Crapo and G.-C. Rota [1970].

Figure 18.1: A page break.

are separated by a `\pagebreak`. Make sure that any `\index` or `\label` commands are moved to the appropriate half of the paragraph. Now all index commands generate the correct page numbers.

Of course, if the page break is between paragraphs, only the `\pagebreak` command is needed. If the break occurs in the middle of a word, use `\-\\linebreak` to add a hyphen.

This method works about 95 percent of the time. Occasionally, you have to drop either the `\linebreak` or the `\pagebreak` command.

Check for missing fonts and other defects

Open the PDF file of your book in Adobe Reader (or even better, in Adobe Acrobat Pro). Under File, go to Properties... and click on the Fonts tab. You will find a long list of fonts. Each one should be marked Embedded Subset.

If all your fonts are embedded, you are in good shape.

Adobe Acrobat Pro has an excellent set of utilities to check whether your PDF file is ready for printing. You find them under Advanced>Preflight.... In the Preflight window, choose Digital printing (B/W)—unless your book will print in color, in which case choose Digital printing (color). Click on Execute. Adobe Acrobat Pro will correct all the mistakes it finds in the file and presents a detailed report.

Adobe Acrobat Pro also comes to the rescue if some fonts are not embedded. In the Preflight window, expand the PDF analysis group and select List text using non-embedded fonts and click on Execute. The report will list all pages with fonts missing and if you select a page, Snap view will show you the trouble spot.

In the help system of Adobe Acrobat Pro, search for Customize Adobe PDF settings and Embed fonts using the TouchUp Text tool for detailed instructions on how to embed the missing fonts.

Other adjustments

- Move the figure and table environments (see Section 10.4.3) physically close to where they appear in the typeset version, and change the optional argument of the figure and table environments to !h.
- Balance the white space on each page as necessary.
- Generate the index only after the page breaks are fixed.

Polish the auxiliary files

- Typeset the document one last time and then place the `\nofiles` command in the preamble (see Section D.3.4) to make sure that the auxiliary files are not overwritten.

- Normally, you should not have to edit the table of contents (toc) file or the lot and lof files (see Section 18.2) and your style file should take care of the formatting. Sometimes, however, an unfortunate page break makes editing necessary. In an appropriate place, you may want to add to the text the command

```
\addtocontents{toc}{\pagebreak}
```

to avoid such edits.

- Create the index (ind) file from the new aux file, as described in Section 17.3. *The Chicago Manual of Style*, 15th edition [11] has a section on bad breaks, remedies, and *Continued* lines in the index. Break the ind file into pages. To minimize bad breaks, use the `\enlargethispage` command where necessary (see Section 5.7.3). Add any *Continued* entries.