GETTING UP AND RUNNING WITH $\mathcal{A}\kern-.1667em\lower.5ex\hbox{$\mathcal{M}$}\kern-.125em\mathcal{S}$-LaTeX

PHILIP S. HIRSCHHORN

Abstract. Together with the template file `template.tex`, these notes are an attempt to tell you enough about LaTeX and AMS-LaTeX so that you can get started without having to read the book.

Contents

1. Introduction 2
2. Basic LaTeX stuff 3
2.1. The `\documentclass` command 3
2.2. Loading optional packages 3
2.3. `\begin{document}` and `\end{document}` 4
2.4. Sections and subsections 4
2.5. Italics *for emphasis* 5
2.6. Cross references and the table of contents 6
3. Title, Author, and the `\maketitle` command 7
3.1. The title 7
3.2. The author, and the author's address 7
3.3. The date 8
3.4. `\maketitle` 8
4. Theorems, Propositions, Lemmas, etc. 8
4.1. Stating theorems, propositions, etc. 8
4.2. Proofs 9
5. Cross-References 10
5.1. References to sections, theorems and equations 10
5.2. References to page numbers 11
5.3. Bibliographic references 12
6. Mathematics in running text 13
7. Displayed mathematics 13
7.1. Single line displays 13
7.2. Text in displayed mathematics 14
7.3. Displaying several equations without alignment 14
7.4. Displays with linebreaks 15
7.5. Displays with alignment 16
7.6. The `cases` environment 22
8. Commutative diagrams using XY-pic 23
8.1. Changing the spacing 25
8.2. Shifting nodes sideways 27

Date: May 24, 2011, Version 2.2.

8.3. Arrows passing under nodes; crossing arrows 28
8.4. Labeling the arrows 29
8.5. More crossing arrows 33
8.6. Making room for large labels 34
8.7. Different arrow styles 36
8.8. Curved arrows 37
8.9. Sliding arrows sideways; multiple arrows 40
8.10. Specifying both the source and the target of an arrow 40
8.11. Text objects 41
8.12. Arrows off of the grid 42
8.13. Dropping objects into a diagram; annotations 43
9. Including graphics 46
9.1. Graphic formats: Postscript or pdf 46
10. Macro definitions, a.k.a. `\newcommand` 47
11. Lists: `itemize, enumerate, and description` 47
11.1. Referring to enumerated items by number 49
12. The bibliography 49
12.1. Using `amsrefs` 49
12.2. The `\bib` command 50
12.3. Obtaining bibliographic information from MathSciNet 51
12.4. Using a database of possible references 51
12.5. Using `amsrefs` with BibTEX 52
12.6. Options for the `amsrefs` package 53
12.7. Avoiding both `amsrefs` and BibTEX 54
13. The template file 54
References 61

1. Introduction

This is an attempt to get you up and running with AMS-LATEX as quickly as possible. These instructions (along with the template file `template.tex`) won't be a substitute for the full documentation, but they'll give you enough to get started quickly and only occasionally have to refer to the main documentation.

The current version of AMS-LATEX (version 2.2) is a collection of document classes and optional packages for the current version of standard LATEX. AMS-LATEX provides the document classes `amsart`, `amsproc`, and `amsbook` (see section 2.1) to replace the standard document classes `article`, `proc`, and `book`, and several optional packages (mainly `amsmath`) that can be used with the standard LATEX document classes. Thus, using AMS-LATEX is really using a variety of LATEX. If you're new to LATEX, and these last few sentences made no sense to you at all, don't worry about it. You don't have to know what the standard LATEX document classes are in order to use the AMS-LATEX replacements for them.

I'll be assuming that you have at least some experience with either plain TEX, AMS-TEX or LATEX, and I'll try to tell you what you need to know so that you can get started with AMS-LATEX *without* actually reading the LATEX user's guide [7], or even taking much of a look at the AMS-LATEX user's guide [1] or the short math guide for LATEX [5].

If you've never used *any* version of TEX or LATEX, then I recommend "The not so short introduction to LATEX 2$_\varepsilon$" by Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl [8]. This is intended for those with no knowledge of TEX or LATEX, and concisely gives a description of what a LATEX document looks like and how you type text and simple mathematics in a LATEX document.

These instructions come with a template file `template.tex`, which is an attempt to give you enough to fake your way through an AMS-LATEX file *almost* without even reading these instructions. I've included the text of that file in these instructions as section 13, so you might want to take a look at that now, and then just use the table of contents of these instructions to find more information on whatever in that file confuses you.

In case you haven't guessed, these instructions were printed using AMS-LATEX, so you can get some idea what it all looks like.

2. Basic LATEX stuff

In this section, we'll describe the three commands that must appear in every LATEX document: `\documentclass`, `\begin{document}`, and `\end{document}`. The complete explanation of these can be found in the LATEX User's Guide [7] or in *The not so short introduction to LATEX 2$_\varepsilon$* [8]. We'll also explain how to begin a new section or subsection of the paper, and how LATEX manages to get the cross-references right (which is also the explanation of why you need to run a file through LATEX *twice* to be sure that all the cross-references are correct).

2.1. The `\documentclass` command.

Before you type anything that actually appears in the paper, you must include a `\documentclass` command. It's easiest to just put the `\documentclass` command at the very beginning of the file, possibly with a few lines of comments before it.

It's the choice of document class that determines whether you're using AMS-LATEX or just plain old LATEX. AMS-LATEX provides the document classes `amsart`, `amsproc`, and `amsbook` as replacements for the standard LATEX document classes `article`, `proc`, and `book`. If for some reason you prefer to use the standard LATEX classes `article`, `proc`, or `book`, you can still get many of the features of AMS-LATEX by including the command `\usepackage{amsmath}` after your `\documentclass` command. I'll only be discussing the `amsart` document class here. For the others, see the AMS-LATEX User's Guide [1].

The simplest version of the `\documentclass` command is

```
\documentclass{amsart}
```

This will give you the default type size, which is 10 point type. If you'd like to use 12 point type, then you should include the optional argument `[12pt]`; this makes the command

```
\documentclass[12pt]{amsart}
```

2.2. Loading optional packages.

There are at least three optional packages that are of interest. The first is the `amsrefs` package, which makes it much easier to create a bibliography (see section 12.1). To load the `amsrefs` package, you put the line

```
\usepackage[lite]{amsrefs}
```

after the `\documentclass` command. (For an explanation of why we recommend using the optional argument `lite`, see section 12.6.)

Another important package is for when you want to use some of the special symbols contained in the $\mathcal{AMS}$-Fonts package. These are listed, along with all of the standard LaTeX symbols, in `symbols.pdf`, available at

http://www.ctan.org/tex-archive/info/symbols/math/symbols.pdf

If you want the standard names for these symbols to be defined for your use, then you need to use the optional package `amssymb`. Thus, to use the default 10 point type, use `amsrefs` to create a bibliography, and have the special symbols defined, use the commands

```
\documentclass{amsart}
\usepackage[lite]{amsrefs}
\usepackage{amssymb}
```

Another widely used optional package is XY-pic, which enables you to draw commutative diagrams as part of your LaTeX file rather than creating them with a graphics package and importing the graphics. (For commutative diagrams, see section 8). To use XY-pic, you should include the commands

```
\usepackage[all,cmtip]{xy}
\let\objectstyle=\displaystyle
```

That loads the XY-pic package and sets it so that the arrowheads used are the same ones used in the rest of the document and the nodes in the diagram are, by default, in `\displaystyle`. If you'd like the default style for the nodes to be `\textstyle`, you should omit the second of those two lines.

This document uses all of those packages, and so we used the commands

```
\documentclass{amsart}
\usepackage[lite]{amsrefs}
\usepackage{amssymb}
\usepackage[all,cmtip]{xy}
\let\objectstyle=\displaystyle
```

2.3. `\begin{document}` and `\end{document}`. Everything that is to appear in the document must appear in between the `\begin{document}` and `\end{document}` commands. There are no optional arguments for these commands, so they always look the same. Anything following the `\end{document}` command is ignored. In addition to `\usepackage` commands (see section 2.2), you are allowed to have macro definitions (i.e., newcommands; see section 10) before the `\begin{document}`, and that's actually a good place for them, but that's about all.

2.4. Sections and subsections. To begin a new section, you give the command

```
\section{Section name}
```

To begin the present section, I gave the command

```
\section{Basic \LaTeX{} stuff}
```

A section number is supplied automatically. If you want to be able to make reference to that section, then you need to *label* it. Since I wanted to be able to demonstrate the cross-reference commands, I actually began this section with the lines

```
\section{Basic \LaTeX{} stuff}
\label{sec:basicstuff}
```

This allows me to type “`section~\ref{sec:basicstuff}`” and have it printed as “section 2”.

To begin a new subsection, you give the command

```
\subsection{Subsection name}
```

To begin the present subsection, I gave the command

```
\subsection{Sections and subsections}
```

A subsection number is supplied automatically. If you want to be able to make reference to that subsection, then you need to *label* it. This subsection was begun with the lines

```
\subsection{Sections and subsections}
\label{sec:sections}
```

so if we type “`section~\ref{sec:sections}`,” it is printed as “section 2.4”.

Labels always take the number of the smallest enclosing structure. Thus, a `\label` command that's inside a section but *not* inside a subsection or Theorem or anything else will take the value of the section counter, while a `\label` command that's inside the statement of a Theorem will take the value of that Theorem number. For more information on this, see section 5.

2.4.1. *Yes, there are subsubsections too.* I began this subsubsection with the command

```
\subsubsection{Yes, there are subsubsections too}
```

Sections without numbers. I began this subsubsection with the command

```
\subsubsection*{Sections without numbers}
```

and got a subsubsection that wasn't numbered. If you give the command

```
\section*{A Section Title}
```

then you'll begin a new section that will not have a number.

2.5. **Italics *for emphasis*.** If you want to use italics to emphasize a word or two, the LaTeX convention is not to explicitly switch to italics, but rather to use the command `\emph` (which means *emphasize*). This command works just like a font change command, except that it switches you *into* italics if the current font is upright and switches you *out of* italics if the current font is italics.

For example, if you type

```
The whole is \emph{more} than the sum of its parts.
```

you'll get

The whole is *more* than the sum of its parts.

but if you type

```
\begin{thm}
  The whole is \emph{more} than the sum of its parts.
\end{thm}
```

you'll get

Theorem 2.1. *The whole is* more *than the sum of its parts.*

Note. The `\emph` command is a recent addition to LaTeX, and it has the feature that it automatically inserts an italic correction where needed. If you don't know what an italic correction is, you can safely ignore this paragraph, but I will at least mention that all those "`\/`" commands frequently seen in TeX (and older LaTeX) documents are all inserting italic corrections; the point of this paragraph is that, with the current version of LaTeX, you don't have to do that anymore.

2.6. **Cross references and the table of contents.** This is an explanation of how LaTeX manages to fill in cross-references (see section 5) to parts of the file it hasn't processed yet, and what those `.aux` and `.toc` files are.

Cross-References. Every time LaTeX processes your file, it writes an *auxiliary* file. Since the file containing these instructions is called `amshelp.tex`, the auxiliary file is called `amshelp.aux`. The auxiliary file contains the definitions of all the keys used for cross-references. When LaTeX begins to process your file, it first looks for an `.aux` file, and reads it in if it exists. Of course, this is the `.aux` file that was produced the *last* time that your file was processed, so the Theorem numbers, Section numbers, etc., are all the ones from the last time the file was processed.

The very first time that LaTeX processes your file, there is no `.aux` file, and so LaTeX gives *lots* of warning messages about undefined labels, or whatever. Ignore all of this. The *next* time that you run LaTeX, there *will* be an `.aux` file, and all the references will be filled in. (Yes, it is possible, at least in theory, for some page number to change every time you run LaTeX on your file, even without any changes in the source file, but this isn't very likely.)

The Table of Contents. If you give the command `\tableofcontents`, then LaTeX will try to write a table of contents at that point, including the page numbers of the sections. Obviously, LaTeX can't know those page numbers or section titles yet, so as LaTeX processes your file, it writes a `.toc` file containing the information it needs. (The `.toc` file for these instructions is `amshelp.toc`.) Once again, LaTeX is always using the information from the *last* time that it processed your file.

If you *do* include a table of contents in your document, and if the table of contents takes up at least a page or so of space, then you might have to run LaTeX *three* times in order to get all of the cross-references right. The reason for this is that the first time you run LaTeX there isn't any `.toc` file listing the section titles, and so the table of contents has nothing in it. The second time you run LaTeX you'll get a table of contents that lists the page numbers for the sections from the last time you ran LaTeX, when the table of contents took up no space at all. Unfortunately, during this second run, the table of contents will be created, and will take up enough space to change the page numbers of the sections from what they were during the first run. Only during the *third* run will the correct page numbers be written into the table of contents. Since this doesn't change the amount of space that the table of contents occupies, this version will be correct.

How do I know when everything is correct? After processing your file, LaTeX checks whether all the cross-reference numbers that it read from the `.aux` file are correct. If any of them are incorrect, it prints a warning on the screen at the very end of the run advising you that labels may have changed and that you should run LaTeX again to get the cross-references right. Unfortunately, LaTeX doesn't seem to check that the table of contents entries are correct, so if you change the name of a section in a way that doesn't make any page references incorrect, you won't be warned to run LATEX again.

3. Title, Author, and the \maketitle command

This stuff should go right after the `\begin{document}` command. I'll give a quick sketch here, which is probably all you'll ever need, but the full explanation is given in *Instructions for preparation of papers and monographs: AMS-LATEX* [2]. If you are already familiar with LATEX, then you should be warned that this part is slightly different from what you do when using the standard LATEX `article` document class.

3.1. The title.

You specify the title with the command

```
\title[Optional running title]{Actual title}
```

These instructions used the title command

```
\title[Running \AmS-\LaTeX]{Getting up and running\\
                                with \AmS-\LaTeX}
```

Notice that you indicate line breaks in the title with a double backslash. If I had decided to omit the line break and also to have the full title printed in the head of the odd numbered pages, I would have used the command

```
\title{Getting up and running with \AmS-\LaTeX}
```

3.2. The author, and the author's address.

The author is specified with an `author` command:

```
\author{Author's name}
```

These directions used the command `\author{Philip S. Hirschhorn}`. The author's address is given in an address command, with double backslashes to indicate line breaks. These instructions used the command

```
\address{Department of Mathematics\\
Wellesley College\\
Wellesley, Massachusetts 02481}
```

If the author's current address is different from the address at which the research was carried out, then you can specify the current address with the command `\curraddr`. For example, you might type

```
\curraddr{Department of Mechanics\\
Brake and Wheel Bearing Division\\
Serene Service Center\\
Salem, Massachusetts 02139}
```

You can also include an email address, with the `\email` command. These instructions used the command

```
\email{psh@math.mit.edu}
```

To acknowledge support, use the command `\thanks`, e.g.,

```
\thanks{Supported in part by NSF grant 3.14159}
```

This will be printed as a footnote on the first page.

Multiple authors. If there are several authors, then each one should have a separate `\author` command, with each individual's address, current address, email address, and thanks following that individual's `\author` command, in its own `\address` command (and `\curraddr` command, and `\thanks` command, and `\email` command). If there *are* several authors, and their combined names are too long for the running head on the even numbered pages, you can give an optional argument to each `\author` command to supply a shortened form to use in the running head, as in

```
\author[P.S. Hirschhorn]{Philip S. Hirschhorn}.
```

(It's apparently a convention that the running head in a multiple author paper should have only initials for the first and middle names, but I don't think that I was invited to that convention.)

3.3. The date.

This is pretty straightforward:

```
\date{Whatever date you please}
```

To have the date of processing used, use the command `\date{\today}`.

3.4. `\maketitle`.

After you've given all of the commands mentioned in this section, you can give the command `\maketitle`. If you *don't* give the command `\maketitle`, a title won't be made. The exact arrangement of all this information is determined by the document class. In particular, the `amsart` document class puts the author's address at the *end* of the paper.

4. Theorems, Propositions, Lemmas, etc.

The instructions in this section assume that you're using the `\newtheorem` commands that I put in the file `template.tex` (see section 13).

4.1. Stating theorems, propositions, etc.

To state a theorem, you do the following:

```
\begin{thm}
  The square of the hypotenuse of a right triangle is equal to the
  sum of the squares of the two adjacent sides.
\end{thm}
```

If you do that, you'll get the following:

Theorem 4.1. *The square of the hypotenuse of a right triangle is equal to the sum of the squares of the two adjacent sides.*

If you thought that it was only a proposition, you'd use

```
\begin{prop}
  The square of the hypotenuse of a right triangle is equal to the
  sum of the squares of the two adjacent sides.
\end{prop}
```

and you'd get

Proposition 4.2. *The square of the hypotenuse of a right triangle is equal to the sum of the squares of the two adjacent sides.*

If you think it's a theorem again, but you'd like to make reference to it in some other part of the paper, you have to choose a *key* with which you'll refer to it, and then *label* the theorem. If you want to use the key *pythagthm*, then it would look like the following:

```
\begin{thm}
  \label{pythagthm}
  The square of the hypotenuse of a right triangle is equal to the
  sum of the squares of the two adjacent sides.
\end{thm}
```

If you later give the command `\ref{pythagthm}`, then that command will expand to the *number* that was assigned to that theorem (in this case, 4.1). For more explanation of cross-references, see section 5.

If you'd like to state a theorem and give a *name* to it, then you can add an optional argument to the `\begin{thm}` command. If you type

```
\begin{thm}[Pythagoras]
  The square of the hypotenuse of a right triangle is equal to the
  sum of the squares of the two adjacent sides.
\end{thm}
```

you'll get

Theorem 4.3 (Pythagoras)**.** *The square of the hypotenuse of a right triangle is equal to the sum of the squares of the two adjacent sides.*

Summary of environments provided in the template. All of the following structures are numbered in the same sequence, in the form SectionNumber.Number. Equations (i.e., displayed formulas, whether they are equations or not) will be numbered in the same sequence.

Theorem Environments

Name	Printed Form	Body font
`thm`	**Theorem**	Italic
`cor`	**Corollary**	Italic
`lem`	**Lemma**	Italic
`prop`	**Proposition**	Italic
`defn`	**Definition**	Normal
`rem`	*Remark*	Normal
`ex`	*Example*	Normal
`notation`	*Notation*	Normal
`terminology`	*Terminology*	Normal

For full details, see the beginning of the template file (reproduced here in section 13), after the comment "The Theorem Environments."

4.2. **Proofs.** To give a proof, you do the following:

```
\begin{proof}
  As any fool can plainly see, it's true!
\end{proof}
```

and you'll get the following:

Proof. As any fool can plainly see, it's true! □

If the theorem said that a condition was both necessary and sufficient for something, and you want to prove each part separately, you can do the following:

```
\begin{proof}[Proof (sufficiency)]
  Well, it's \emph{obviously} sufficient!
\end{proof}
```

and you'll get

Proof (sufficiency). Well, it's *obviously* sufficient! □

That is, the `proof` environment allows you to use an optional second argument that will appear in place of the word `Proof`.

If the proof of Theorem 4.1 does not appear immediately after its statement, you might use the following:

```
\begin{proof}[Proof of Theorem~\ref{pythagthm}]
  As any fool can plainly see, it's true!
\end{proof}
```

and you'd get

Proof of Theorem 4.1. As any fool can plainly see, it's true! □

5. Cross-References

This section explains how to make reference to numbered sections, theorems, equations, and bibliography items, with the correct reference numbers filled in automatically by LaTeX.

5.1. **References to sections, theorems and equations.** For each structure in the manuscript to which you'll be making reference, you must assign a *key* that you'll use to refer to that structure. For sections, theorems, numbered equations, and items in an enumerated list (see section 11.1), you assign the key using the `\label` command and refer to it using either the `\ref` command or the `\eqref` command. Each of these commands takes one argument, which is the *key* you're assigning to the object. The command `\ref{key}` produces the number that was assigned to that structure and the command `\eqref{key}` produces that number enclosed in parentheses. (The `\eqref` command also ensures that the number and parentheses are always in an upright font; see section 5.1.1.) The convention is to use `\eqref` to refer to equation numbers, `\cite` to refer to bibliography entries (see section 5.3), and `\ref` to refer to everything else.

Consider the following example.

Theorem 5.1. *If the maps $f\colon X \to Y$ and $g\colon X \to Y$ are homotopic, then the induced homomorphisms $f_{*}\colon \mathrm{H}_{*}X \to \mathrm{H}_{*}Y$ and $g_{*}\colon \mathrm{H}_{*}X \to \mathrm{H}_{*}Y$ are equal.*

We typed that theorem as follows.

```
\begin{thm}
  \label{homotopy}
  If the maps $f\colon X \to Y$ and $g\colon X \to Y$ are homotopic,
  then the induced homomorphisms $f_{*} \colon \mathrm{H}_{*}X \to
  \mathrm{H}_{*}Y$ and $g_{*} \colon \mathrm{H}_{*}X \to
  \mathrm{H}_{*}Y$ are equal.
```

```
\end{thm}
```

If we now type “`see Theorem~\ref{homotopy}`,” then it will be printed as “see Theorem 5.1.”

So, what exactly is the label labeling? The command `\label{key}` assigns to `key` the value of the *smallest enclosing structure.* For example, we began this section by typing

```
\section{Cross-References}
\label{sec:xreferences}
```

and we began this subsection by typing

```
\subsection{References to sections, theorems and equations}
\label{sec:thmrefs}
```

The phrase “`See section~\ref{sec:xreferences}`” is printed as “See section 5” while the phrase “`See section~\ref{sec:thmrefs}`” is printed as “See section 5.1” because the key `sec:xreferences` was defined inside of section 5 but outside of section 5.1, while the key `sec:thmrefs` was defined inside of section 5.1.

5.1.1. *References to equations.* To make reference to a numbered equation, you assign the *key* as before, but you replace `\ref` with `\eqref`, so that parentheses will be printed around the equation number. For example, if you type

```
\begin{equation}
  \label{additivity}
  \mathrm{H}_{*} \bigvee_{\alpha\in A} X_{\alpha}   \approx
    \bigoplus_{\alpha\in A}\mathrm{H}_{*} X_{\alpha}
\end{equation}
```

then you’ll get

$$\mathrm{H}_{*} \bigvee_{\alpha\in A} X_{\alpha} \approx \bigoplus_{\alpha\in A}\mathrm{H}_{*} X_{\alpha} \tag{5.2}$$

If we now type

```
\begin{thm}
  Equation~\eqref{additivity} is true for all sorts of functors
  $\mathrm{H}$.
\end{thm}
```

then we’ll get

Theorem 5.3. *Equation* (5.2) *is true for all sorts of functors* H.

Notice the parentheses around the equation number, and the fact that even though the theorem is set in slanted type, the equation number is set in an upright font? This is the difference between `\eqref` and `\ref`; the command `\eqref` provides parentheses, arranges it so that the number and surrounding parentheses are in an upright font no matter what the surrounding font, and supplies an italic correction if it’s needed.

5.2. **References to page numbers.** If you want to make reference to the *page* that contains a label, rather than to the structure that is labeled, use the command `\pageref{key}`. For example, if you type

```
See page~\pageref{homotopy} to find Theorem~\ref{homotopy}.
```

you’ll get “See page 10 to find Theorem 5.1.”

5.3. **Bibliographic references.** Each bibliography item receives a *key* as part of its basic structure, and you refer to that item using the command `\cite{key}`.

When using the `amsrefs` package, each item in the bibliography is begun with

```
\bib{key}{TypeOfItem}{
```

For example, the bibliography of these instructions contains the entry

```
\bib{HA}{book}{
  author={Quillen, Daniel G.},
  title={Homotopical Algebra},
  series={Lecture Notes in Mathematics},
  volume={43},
  publisher={Springer-Verlag},
  address={Berlin-New York},
  date={1967}
}
```

If we type “`This is the work of Quillen~\cite{HA}`,” then it will be printed as “This is the work of Quillen [9].” Notice that square brackets have been inserted around the bibliography item number.

The `\cite` command takes an optional argument, which allows you to annotate the reference. If we type “`see~\cite[Chapter I]{HA}`”, then it will be printed as “see [9, Chapter I]”. If you’re using `amsrefs` (which we strongly recommend), then there’s an alternate form available: If we type “`\cite{HA}*{Chapter I}`”, then we also get “[9, Chapter I]”, and this second form is less likely to cause errors when used, e.g., in the optional argument to a `\begin{theorem}` command (see section 4.1).

5.3.1. *Multiple references.* If you’re using `amsrefs` and when you refer to multiple bibliography items you want to have the item numbers automatically sorted and compressed (e.g., replacing “7, 6, 5, 8” with “5–8”), you can use the `\cites` command. For example, if we type `\cites{HA,yellowmonster}` then we get [4, 9].

If you want some of the references in the list to have annotations, you put your `\cite` commands into the argument of a `\citelist` command. For example, if we type `\citelist{\cite{HA}*{Chapter I} \cite{yellowmonster}}` then we get [4; 9, Chapter I].

5.3.2. *Author-year citations.* If you’re using `amsrefs` and you use the `author-year` option to convert your references to the author-year format (see section 12.6), then there are two variations on the `\cite` command (`\ycite` and `\ocite`) that are useful. If we were using the `author-year` option in this document, then

“`\cite{yellowmonster}`” would be typeset as “(Bousfield and Kan, 1972)”
“`\ycite{yellowmonster}`” would be typeset as “(1972)”
“`\ocite{yellowmonster}`” would be typeset as “Bousfield and Kan (1972)”

Thus, you could type “For further details, see `\cite{yellowmonster}`”, or “Bousfield and Kan `\ycite{yellowmonster}` showed…”, or “This can be found in `\ocite{yellowmonster}`”.

6. Mathematics in running text

This is pretty much exactly as it is in plain TeX, except that you have an extra option (which you can ignore). The simplest thing is to just enclose between dollar signs any material that should be in math mode. Thus, if you type

```
Let $f\colon X \to Y$ be a continuous function.
```

you'll get

Let $f\colon X \to Y$ be a continuous function.

(Note that if we had typed that as "`f: X \to Y`" then the spacing around the colon would be wrong, since the colon symbol is typeset as a binary relation, with spaces on the two sides equal.) The only novelty that LaTeX introduces is that, instead of using a dollar sign to toggle math mode on and off, you can use '`\(`' to *begin* math mode, and '`\)`' to *end* math mode. Thus, the example above could also be typed as

```
Let \(f\colon X \to Y\) be a continuous function.
```

This provides a tiny bit more error checking, but can otherwise be safely ignored.

The reference [5] is an excellent concise summary of the features of AMS-LaTeX for typesetting mathematics, both displayed and in running text. It also includes lists of symbols available for mathematics.

7. Displayed mathematics

For a complete discussion of the environments for displayed mathematics, including options for customizing equation numbers, see [1, section 3].

7.1. **Single line displays.**

To display mathematics and number the display (so that you can refer to it from elsewhere in the paper) you use the `equation` environment. (LaTeX calls all such numbers *equation numbers*, whether or not the display has anything to do with equations.) If you type

```
\begin{equation}
  \pi_{1}(X \vee Y) \approx \pi_{1}X * \pi_{1}Y
\end{equation}
```

you'll get

$$\pi_1(X \vee Y) \approx \pi_1 X * \pi_1 Y \tag{7.1}$$

If you'd like to be able to make reference to the equation number, you need to *label* the equation, using a *key* that you can use for referencing it:

```
\begin{equation}
  \label{pi1eqn}
  \pi_{1}(X \vee Y) \approx \pi_{1}X * \pi_{1}Y
\end{equation}
```

If you later type "`see formula~\eqref{pi1eqn}`" you'll get "see formula (7.1)." (For more on cross-references to formulas, see section 5.1.)

To display a single line of mathematics without an equation number, you use the `equation*` environment. (This is a common LaTeXism: Adding an asterisk to the name of a numbered LaTeX environment often gives the unnumbered equivalent.) If you type

```
\begin{equation*}
  \pi_{1}(X \vee Y) \approx \pi_{1}X * \pi_{1}Y
\end{equation*}
```

then you'll get

$$\pi_1(X \vee Y) \approx \pi_1 X * \pi_1 Y$$

The `displaymath` environment is equivalent to this; you can produce the same unnumbered display by typing

```
\begin{displaymath}
  \pi_{1}(X \vee Y) \approx \pi_{1}X * \pi_{1}Y
\end{displaymath}
```

You can also produce that same unnumbered display by typing

```
\[
  \pi_{1}(X \vee Y) \approx \pi_{1}X * \pi_{1}Y
\]
```

but some people think that that makes your LaTeX source file less readable. Although you can also produce the same thing by typing

```
$$
  \pi_{1}(X \vee Y) \approx \pi_{1}X * \pi_{1}Y
$$
```

using double dollar signs is definitely deprecated.

7.2. **Text in displayed mathematics.** To include text in a displayed mathematics environment, you use the `\text` command, as in `\text{here's some text}`. The `\text` command is preferable to the standard LaTeX `\mbox` command because `\text` correctly adjusts its size for use in subscripts and superscripts. For example, if you type

```
\begin{equation*}
  \pi_{1}(X) \approx G * H
  \quad\text{where $G$ is torsion and $H$ is torsion free}
\end{equation*}
```

then you'll get

$$\pi_1(X) \approx G * H \quad\text{where G is torsion and H is torsion free}$$

and if you type

```
\begin{equation*}
  X_{n} = \coprod_{\text{monomorphisms $[k] \to [n]$}} S_{k}
\end{equation*}
```

then you'll get

$$X_n = \coprod_{\text{monomorphisms $[k] \to [n]$}} S_k$$

7.3. **Displaying several equations without alignment.** You can put several displayed lines together, each one centered, with no alignment between the different lines, using the `gather` environment. When typing this, the lines are separated by a double backslash `\\`. For example, if you type

```
\begin{gather}
  (X\otimes L) \amalg_{(X\otimes K)} (Y\otimes K)
    \longrightarrow Y\otimes L\\
  X^{L} \longrightarrow X^{K} \times_{Y^{K}} Y^{L}
\end{gather}
```

then you'll get

$$(X \otimes L) \amalg_{(X\otimes K)} (Y \otimes K) \longrightarrow Y \otimes L \tag{7.2}$$

$$X^{L} \longrightarrow X^{K} \times_{Y^{K}} Y^{L} \tag{7.3}$$

The `gather*` environment would produce the same thing without the equation numbers. You can also label each line so that you can refer to them: If you had typed that as

```
\begin{gather}
  \label{eq:push}
  (X\otimes L) \amalg_{(X\otimes K)} (Y\otimes K)
    \longrightarrow Y\otimes L\\
  \label{eq:pull}
  X^{L} \longrightarrow X^{K} \times_{Y^{K}} Y^{L}
\end{gather}
```

and then typed "`see \eqref{eq:push} or \eqref{eq:pull}`", you'd get "see (7.2) or (7.3)".

There is also a `gathered` environment, which does not produce a display of its own but rather produces a block of centered lines of mathematics that can be used inside of another displayed mathematics environment (see section 7.5.3).

7.4. **Displays with linebreaks.** For a long display that must be broken across several lines, you can use the `multline` environment. (There is also a `multline*` environment, which is similar except that it omits the equation number.) When typing this, the lines are separated by a double backslash `\\`. The first line will be shifted left of center, the last will be shifted right of center, and the lines in between those will be centered. For example, if you type

```
\begin{multline}
  \label{eq:BigComp}
  \mathrm{F} X\otimes\Delta[n]   \xrightarrow{1 \otimes D}
    \mathrm{F} X\otimes(\Delta[n]\times\Delta[n])\\
  \xrightarrow{\sigma}
    \mathrm{F}\bigl(X\otimes(\Delta[n]\times\Delta[n])\bigr)
    \approx
    \mathrm{F}\bigl((X\otimes\Delta[n])\otimes\Delta[n]\bigr)\\
  \xrightarrow{\mathrm{F}(\alpha\otimes 1)}
    \mathrm{F}(Y\otimes\Delta[n]) \xrightarrow{\mathrm{F}(\beta)}
    \mathrm{F}(Z)
\end{multline}
```

then you'll get

$$
\begin{aligned}
&\mathrm{F}X \otimes \Delta[n] \xrightarrow{1\otimes D} \mathrm{F}X \otimes (\Delta[n]\times\Delta[n])\\
&\qquad\xrightarrow{\sigma} \mathrm{F}\bigl(X\otimes(\Delta[n]\times\Delta[n])\bigr) \approx \mathrm{F}\bigl((X\otimes\Delta[n])\otimes\Delta[n]\bigr)\\
&\qquad\qquad\xrightarrow{\mathrm{F}(\alpha\otimes 1)} \mathrm{F}(Y\otimes\Delta[n]) \xrightarrow{\mathrm{F}(\beta)} \mathrm{F}(Z)
\end{aligned}
\tag{7.4}
$$

You can then type "`the composition \eqref{eq:BigComp}`" and it will appear as "the composition (7.4)".

7.5. **Displays with alignment.** There are a number of ways to produce displays with multiple lines and horizontal alignment between the lines.

- The `align` and `align*` environments allow horizontal alignment of characters chosen from each line (see section 7.5.1).
- The `aligned` and `split` environments produce alignments that can be be part of a larger display (see section 7.5.2).
- The `gathered` environment produces a block of several lines, each centered as in the `gather` environment (see section 7.3), that can be part of a larger display (see section 7.5.3). (Of course, this doesn't involve horizontal alignment, but the `gathered` environment is often used in conjunction with other display environments.)
- The `align` and `align*` environments also allow for multiple alignment points (see section 7.5.4).
- The `flalign` and `flalign*` environments allow multiple alignment points and place the outermost aligned blocks flush against the margins (see section 7.5.5).
- The `alignat` and `alignat*` environments allow multiple alignment points and allow you to choose the spacing between the columns (see section 7.5.6).
- The `alignedat` environment is similar to the `alignat*` environment, except that it produces an alignment intended to be part of a larger display (see section 7.5.7).

7.5.1. *Displays with a single alignment point.* To display several lines of mathematics with alignment between the lines, you use the `align` environment. (There is also an `align*` environment, which is similar except that it omits the equation numbers.) When typing this, the lines are separated by a double backslash `\\` and each line has an ampersand `&` immediately preceding the symbol to be aligned with the corresponding symbols on the other lines. For example, if you type

```
\begin{align}
  \label{eq:pi1}
  \pi_{1}(X\vee Y) &\approx \pi_{1}X * \pi_{1}Y\\
  \label{eq:additivity}
  \widetilde{\mathrm{H}}_{*}(X\vee Y) &\approx
    \widetilde{\mathrm{H}}_{*}X \oplus \widetilde{\mathrm{H}}_{*}Y
\end{align}
```

then you'll get

$$
\pi_1(X\vee Y) \approx \pi_1 X * \pi_1 Y \tag{7.5}
$$

$$
\widetilde{\mathrm{H}}_*(X\vee Y) \approx \widetilde{\mathrm{H}}_* X \oplus \widetilde{\mathrm{H}}_* Y \tag{7.6}
$$

and if you type "see `\eqref{eq:pi1} or \eqref{eq:additivity}`" then you'll get "see (7.5) or (7.6)". For another example, if you type

```
\begin{align*}
  \mathcal{M}\bigl((\operatorname{colim} \boldsymbol{X})
    \otimes K,Y\bigr) &\approx
    \mathcal{M}(\operatorname{colim} \boldsymbol{X},Y^K)\\
  &\approx \lim \mathcal{M}(\boldsymbol{X},Y^K)\\
  &\approx \lim \mathcal{M}(\boldsymbol{X} \otimes K,Y)\\
  &\approx \mathcal{M}\bigl(\operatorname{colim}
    (\boldsymbol{X} \otimes K),Y\bigr)
\end{align*}
```

then you'll get

$$\begin{aligned}\mathcal{M}\bigl((\operatorname{colim}\boldsymbol{X})\otimes K,Y\bigr)&\approx\mathcal{M}(\operatorname{colim}\boldsymbol{X},Y^K)\\&\approx\lim\mathcal{M}(\boldsymbol{X},Y^K)\\&\approx\lim\mathcal{M}(\boldsymbol{X}\otimes K,Y)\\&\approx\mathcal{M}\bigl(\operatorname{colim}(\boldsymbol{X}\otimes K),Y\bigr)\end{aligned}$$

Both the `align` and `align*` environments allow you to have multiple columns in the display; for this, see section 7.5.4.

7.5.2. *Alignments that are part of a larger display.* The `aligned` environment does not create a display of its own. Instead, it allows you to have several lines of mathematics, with alignment as in the `align*` environment (see section 7.5.1), which can then be used inside of a displayed mathematics environment.

There is also a `split` environment, which is similar to the `aligned` environment except that the `aligned` environment has options (changing the vertical alignment (see the examples in this section) or having multiple alignment points (see section 7.5.4)) that are not possible with the `split` environment. Thus, a `split` environment can always be replaced by an `aligned` environment.

For example, if you type

```
\begin{equation}
  \label{eq:dblaligned}
  \begin{aligned}
    a &= b + c\\
    d &= e + f
  \end{aligned}
  \qquad\text{which, of course, leads to}\qquad
  \begin{aligned}
    g &= h + i\\
    j &= k + l
  \end{aligned}
\end{equation}
```

then you'll get

$$\begin{aligned}a&=b+c\\d&=e+f\end{aligned}\qquad\text{which, of course, leads to}\qquad\begin{aligned}g&=h+i\\j&=k+l\end{aligned}\tag{7.7}$$

The above example illustrates that, by default, when an `aligned` (or `aligned*`) environment is used as part of a display involving other elements, it is vertically aligned at its center. The `aligned` environment can take an optional argument of `[t]` or `[b]` to cause the alignment to be at either the top or the bottom row. For example, if you type

```
\begin{equation}
  \label{eq:dbltop}
  \begin{aligned}[t]
    a &= b + c\\
    d &= e + f
  \end{aligned}
  \qquad\text{which, of course, leads to}\qquad
  \begin{aligned}[t]
    g &= h + i\\
    j &= k + l
  \end{aligned}
\end{equation}
```

then you'll get

$$\begin{aligned}[t] a &= b + c\\ d &= e + f \end{aligned} \qquad\text{which, of course, leads to}\qquad \begin{aligned}[t] g &= h + i\\ j &= k + l \end{aligned} \tag{7.8}$$

and if you type

```
\begin{equation}
  \label{eq:dblbot}
  \begin{aligned}[b]
    a &= b + c\\
    d &= e + f
  \end{aligned}
  \qquad\text{which, of course, leads to}\qquad
  \begin{aligned}[b]
    g &= h + i\\
    j &= k + l
  \end{aligned}
\end{equation}
```

then you'll get

$$\begin{aligned}[b] a &= b + c\\ d &= e + f \end{aligned} \qquad\text{which, of course, leads to}\qquad \begin{aligned}[b] g &= h + i\\ j &= k + l \end{aligned} \tag{7.9}$$

The **aligned** environment doesn't produce any equation numbers of its own. If the enclosing display environment produces a number, and if the `aligned` environment doesn't use the optional argument of `[t]` or `[b]`, then that number is centered vertically in the contents of the `aligned`. This can also be useful if you need to display one very long equation that you need to split over several lines and you want the equation number to be centered vertically. For example, if you type

```
\begin{equation}
  \label{eq:longeq}
  \begin{aligned}
    F(a,b,c,d) &= \int_{a}^{b} \sin^{3}x \cos^{2}x \, dx
      + \int_{c}^{d} \sin^{3}x \cos x \, dx\\
    &\quad + \int_{a}^{b} \sin^{2} x \cos^{2} x \, dx + \int_{c}^{d}
```

```
      \sin x \cos x \, dx
  \end{aligned}
\end{equation}
```

then you'll get

$$\begin{aligned} F(a,b,c,d) = \int_{a}^{b} \sin^{3} x \cos^{2} x \, dx + \int_{c}^{d} \sin^{3} x \cos x \, dx \\ + \int_{a}^{b} \sin^{2} x \cos^{2} x \, dx + \int_{c}^{d} \sin x \cos x \, dx \end{aligned} \tag{7.10}$$

The display created by an `aligned` environment can be used as an element of a display. For example, if you type

```
\begin{equation*}
  \left .
    \begin{aligned}
      f(x) &= \sin^{2} x\\
      g(x) &= \cos^{2} x\\
      h(x) &= \sin x + \cos x
    \end{aligned}
  \right \}
  \qquad \text{A nice collection of functions}
\end{equation*}
```

then you'll get

$$\left. \begin{aligned} f(x) &= \sin^{2} x \\ g(x) &= \cos^{2} x \\ h(x) &= \sin x + \cos x \end{aligned} \right\} \qquad \text{A nice collection of functions}$$

7.5.3. *Several centered lines as part of a larger display.* The `gathered` environment does not create a display of its own. Instead, it allows you to have several lines of mathematics, each one centered as in the `gather` environment (see section 7.3), which can then be used inside of a displayed mathematics environment. For example, if you type

```
\begin{equation}
  \begin{gathered}
    x^{2} + y^{2} = z^{2}\\
    a = b + c
  \end{gathered}
  \qquad \text{and, in addition,}\qquad
  \begin{aligned}
    A &= B+C\\
    &= D+E
  \end{aligned}
\end{equation}
```

then you'll get

$$\begin{gathered} x^{2} + y^{2} = z^{2} \\ a = b + c \end{gathered} \qquad \text{and, in addition,} \qquad \begin{aligned} A &= B+C \\ &= D+E \end{aligned} \tag{7.11}$$

The gathered environment can take the same optional argument of [t] or [b] that the aligned argument can take (see section 7.5.2) to cause the vertical alignment to be on either the top or the bottom line. For example, if you type

```
\begin{equation}
  \begin{gathered}[t]
    x^{2} + y^{2} = z^{2}\\
    a = b + c
  \end{gathered}
  \qquad \text{and, in addition,}\qquad
  \begin{aligned}[t]
    A &= B+C\\
    &= D+E
  \end{aligned}
\end{equation}
```

then you'll get

$$\begin{gathered}[t] x^{2} + y^{2} = z^{2}\\ a = b + c \end{gathered} \qquad \text{and, in addition,}\qquad \begin{aligned}[t] A &= B+C\\ &= D+E \end{aligned} \tag{7.12}$$

7.5.4. *Multiple alignment points.* The align and align* environments (see section 7.5.1) as well as the aligned environment (see section 7.5.2) can be used to create displays in which each line is broken into several pieces, with each piece horizontally aligned at a chosen character. That is: they can be used to create a display with several columns, with alignment within each column.

When typing this, the lines are separated by a double backslash \\, the different displays on each line are separated by an ampersand &, and the symbols to be aligned are preceded by an ampersand &. For example, if you type

```
\begin{align*}
  K &\approx G * H&   i&= j+k&  B &\subset C\\
  H &\approx A_{0}*B_{0}&  i'&= j'+k'&  C &= D\cap E\\
  G &\approx \coprod_{\alpha\in A} L_{\alpha}&  i''&=j''+k''&
                    A &= D \cup E
\end{align*}
```

then you'll get

$$\begin{aligned} K &\approx G * H & i&= j+k & B &\subset C\\ H &\approx A_{0}*B_{0} & i'&= j'+k' & C &= D\cap E\\ G &\approx \coprod_{\alpha\in A} L_{\alpha} & i''&=j''+k'' & A &= D \cup E \end{aligned}$$

When typing such an alignment with n columns, each line will have at most $2n+1$ columns.

For another example, if you type

```
\begin{align*}
  \pi_{1}(X\vee Y) &\approx \pi_{1}(X) * \pi_{1}(Y)
    &&\text{(by the van Kampen theorem)}\\
  &\approx G*H &&\text{(by the computation in the previous section)}
\end{align*}
```

then you'll get

$$\begin{aligned} \pi_1(X \vee Y) &\approx \pi_1(X) * \pi_1(Y) && \text{(by the van Kampen theorem)}\\ &\approx G * H && \text{(by the computation in the previous section)} \end{aligned}$$

If you want to specify the separation between the columns in the alignment, you should use the `alignat` environment (see section 7.5.6).

7.5.5. *Alignments flush left and flush right.* To produce alignments similar to those in section 7.5.4 except with the leftmost column flush left and the rightmost column flush right, you use the `flalign` environment (or, to omit the equation numbers, the `flalign*` environment). For example, if you type

```
\begin{flalign*}
  K &\approx G * H&   i&= j+k&  B &\subset C\\
  H &\approx A_{0}*B_{0}&  i'&= j'+k'&  C &= D\cap E\\
  G &\approx \coprod_{\alpha\in A} L_{\alpha}&  i''&=j''+k''&
                                         A &= D \cup E
\end{flalign*}
```

then you'll get

$$\begin{aligned} K &\approx G * H & i &= j+k & B &\subset C\\ H &\approx A_0 * B_0 & i' &= j'+k' & C &= D\cap E\\ G &\approx \coprod_{\alpha\in A} L_\alpha & i'' &= j''+k'' & A &= D\cup E \end{aligned}$$

7.5.6. *Multiple alignment points with chosen spacing.* To produce alignments as in section 7.5.4 except with the ability to choose the horizontal space between the columns, you use the `alignat` environment (or, to omit the equation numbers, the `alignat*` environment). These environments don't insert any horizontal space between the columns, and so you can insert the exact amount of space you want by including it at the beginning of one of the columns.

The format of `alignat` is slightly different from that of the `align` environment in that you must include an argument specifying the number of columns. For example, if you type

```
\begin{alignat}{2}
  K &\approx G*H&  \qquad&\text{(by an earlier theorem)}\\
  A &\approx \lim_{i\in I} A_{i}&  &\text{(by the definition of $A$)}
\end{alignat}
```

then you'll get

$$K \approx G * H \qquad \text{(by an earlier theorem)} \tag{7.13}$$

$$A \approx \lim_{i\in I} A_i \qquad \text{(by the definition of A)} \tag{7.14}$$

Note that it's only necessary to insert the `\qquad` space in one row; the alignment forces the space to appear in all rows.

7.5.7. *Multiple alignment points with chosen spacing as part of a larger display.* The `alignedat` environment is similar to the `aligned` environment (see section 7.5.2), except that the `alignedat` environment allows you to choose the spacing between the columns. That is, the `alignedat` environment doesn't insert any space between the columns, so you can insert the exact amount of space you want by including it at the beginning of one of the columns. The `alignedat` environment also has a required argument stating the number of columns. For example, if you type

```
\begin{equation*}
  \left \{
    \begin{alignedat}{2}
      a &= b + c& d &= e + f\\
      A &= B + C \qquad&  D &= E + F
    \end{alignedat}
  \right \}
  \qquad \text{Twin pairs of equations}
\end{equation*}
```

then you'll get

$$\left\{\begin{alignedat}{2} a &= b + c & d &= e + f\\ A &= B + C \qquad& D &= E + F \end{alignedat}\right\}\qquad\text{Twin pairs of equations}$$

Note that it's only necessary to insert the `\qquad` space in one row; the alignment forces the space to appear in all rows.

7.6. **The `cases` environment.** There is a `cases` environment, which constructs the usual display of several cases, and which is used as a part of one of the displayed mathematics environments. For example, if you type

```
\begin{equation}
  \label{eq:abs}
  |x| =
  \begin{cases}
    x& \text{if $x \ge 0$}\\
    -x& \text{if $x < 0$}
  \end{cases}
\end{equation}
```

then you'll get

$$|x| = \begin{cases} x & \text{if } x \ge 0\\ -x & \text{if } x < 0 \end{cases} \tag{7.15}$$

For another example, if you type

```
\begin{align*}
  d_{i} \sigma &=
  \begin{cases}
    \alpha_{1} \xrightarrow{\sigma_{1}} \alpha_{2}
          \xrightarrow{\sigma_{2}} \cdots
          \xrightarrow{\sigma_{n-1}} \alpha_{n}
      &\text{if $i=0$}\\
    \alpha_0 \xrightarrow{\sigma_{0}} \cdots
          \xrightarrow{\sigma_{i-2}} \alpha_{i-1}
          \xrightarrow{\sigma_{i}\sigma_{i-1}} \alpha_{i+1}
          \xrightarrow{\sigma_{i+1}} \cdots
          \xrightarrow{\sigma_{n-1}} \alpha_{n}
      &\text{if $0<i<n$}\\
    \alpha_{0} \xrightarrow{\sigma_{0}} \alpha_{1}
```

```
          \xrightarrow{\sigma_1} \cdots
          \xrightarrow{\sigma_{n-2}} \alpha_{n-1}
      &\text{if $i=n$}
  \end{cases}\\
  s_{i} \sigma &= \alpha_{0} \xrightarrow{\sigma_{0}} \cdots
    \xrightarrow{\sigma_{i-1}} \alpha_{i}
    \xrightarrow{1_{\alpha_{i}}} \alpha_{i}
    \xrightarrow{\sigma_{i}} \alpha_{i+1}
    \xrightarrow{\sigma_{i+1}} \cdots
    \xrightarrow{\sigma_{n-1}} \alpha_{n}
\end{align*}
```

then you'll get

$$\begin{aligned} d_i\sigma &= \begin{cases} \alpha_1 \xrightarrow{\sigma_1} \alpha_2 \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_{n-1}} \alpha_n & \text{if } i=0 \\ \alpha_0 \xrightarrow{\sigma_0} \cdots \xrightarrow{\sigma_{i-2}} \alpha_{i-1} \xrightarrow{\sigma_i\sigma_{i-1}} \alpha_{i+1} \xrightarrow{\sigma_{i+1}} \cdots \xrightarrow{\sigma_{n-1}} \alpha_n & \text{if } 0<i<n \\ \alpha_0 \xrightarrow{\sigma_0} \alpha_1 \xrightarrow{\sigma_1} \cdots \xrightarrow{\sigma_{n-2}} \alpha_{n-1} & \text{if } i=n \end{cases} \\ s_i\sigma &= \alpha_0 \xrightarrow{\sigma_0} \cdots \xrightarrow{\sigma_{i-1}} \alpha_i \xrightarrow{1_{\alpha_i}} \alpha_i \xrightarrow{\sigma_i} \alpha_{i+1} \xrightarrow{\sigma_{i+1}} \cdots \xrightarrow{\sigma_{n-1}} \alpha_n \end{aligned}$$

8. Commutative diagrams using XY-pic

XY-pic is a powerful package that enables you to draw very complex commutative diagrams within your LATEX file, avoiding the need to create the graphics separately and then import them. XY-pic isn't a part of $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX, but XY-pic and $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX work quite well together. We'll describe the most important features of XY-pic's commutative diagram macros here. The full documentation for commutative diagrams can be found in the *XY-pic User's Guide* [11], and the full documentation for all of XY-pic, which can do quite a bit more than draw commutative diagrams, can be found in the *XY-pic Reference Manual* [12].

To use XY-pic to draw commutative diagrams, you include the lines

```
\usepackage[all,cmtip]{xy}
\let\objectstyle=\displaystyle
```

in the preamble of your document, i.e., after the `\documentclass` command and before the `\begin{document}` command. You can omit the second of those two lines if you want the nodes of your diagram to be in `\textstyle` by default.

You create a diagram using the `\xymatrix` command, immediately followed by a pair of braces that enclose the diagram specification. The diagram is described in a manner similar to the way matrices are described:

(1) There is a rectangular array of nodes of the diagram. A node should not begin with a macro or with an asterisk, and so it's safest (although not always required) to enclose each node in braces `{}`.
(2) The nodes in a row are separated by an ampersand `&`, and the rows are separated by a double backslash `\\`.
(3) An arrow is specified immediately following the node that is its source (although this is not actually required; see section 8.10). A straight line arrow is specified by `\ar[target]`, where `target` is
   - `r` for the node one place to the right,
   - `l` for the node one place to the left,
   - `d` for the node one place down,

- `u` for the node one place up,
- `rr` for the node two places to the right,
- `dr` for the node one place down and one place to the right,
- etc.

It's also possible to have labels on the arrows (see section 8.4), arrows that are built of things other than a single solid line with a single arrowhead at the end (see section 8.7), curved arrows (see section 8.8), and arrows that pass over or under other arrows (see sections 8.3 and 8.5).

For example, if you type

```
\begin{equation}
  \xymatrix{
    {A}\ar[r] \ar[d]
    &{B} \ar[d]\\
    {C} \ar[r]
    &{D}
  }
\end{equation}
```

then you'll get

$$\begin{tikzcd} A \arrow[r] \arrow[d] & B \arrow[d] \\ C \arrow[r] & D \end{tikzcd} \tag{8.1}$$

If you'd like the equation number to be centered vertically with respect to the diagram, you should enclose the `\xymatrix` command in a `\vcenter`, as in

```
\begin{equation}
  \vcenter{
    \xymatrix{
      {A}\ar[r] \ar[d]
      &{B} \ar[d]\\
      {C} \ar[r]
      &{D}
    }
  }
\end{equation}
```

in which case you'll get

$$\begin{tikzcd} A \arrow[r] \arrow[d] & B \arrow[d] \\ C \arrow[r] & D \end{tikzcd} \tag{8.2}$$

Not all nodes need appear. To omit a node in the middle of a line, just type the `&` that moves you on to the next node. To omit a node at the end of the line, just type the line-ending `\\` and go on to the next line. For example, if you type

```
\begin{equation*}
  \xymatrix{
    {A} \ar[rr] \ar[dr] \ar[d]
    && {B} \ar[dd]\\
    {C} \ar[d] \ar[drr]
    & {D} \ar[l] \ar[ur]\\
    {E} \ar[rr]
    &&{F}
  }
\end{equation*}
```

you'll get

$$\begin{tikzcd}
A \arrow[rr] \arrow[d] \arrow[dr] & & B \arrow[dd] \\
C \arrow[d] \arrow[drr] & D \arrow[l] \arrow[ur] & \\
E \arrow[rr] & & F
\end{tikzcd}$$

Note that that diagram has three rows and three columns, but two of the nodes in the second column and one of the nodes in the third column are missing.

8.1. **Changing the spacing.** You can change the space between the rows of a diagram and the space between the columns of a diagram. The following commands can be inserted following the command `\xymatrix` and before the left brace that begins the diagram specification:

`@=dimen`	set row and column spacing to `dimen`
`@R=dimen`	set row spacing to `dimen`
`@C=dimen`	set column spacing to `dimen`

The `dimen` in all of these commands can be positive, zero, or negative. (Negative `dimen`'s can be useful in diagrams with very wide entries; see (8.3).) Although you can use whatever units you like for `dimen` (e.g., inches (`in`), centimeters (`cm`), points (`pt`), etc.), it's a good idea to measure vertical distances in `ex` (roughly the height of an upper case "X") and horizontal distances in `em` (roughly the width of an upper case "M"), so that the distance will be roughly correct even if you change font sizes.

For example, if you type

```
\begin{displaymath}
  \vcenter{
    \xymatrix@=2ex{
      {A} \ar[r] \ar[d]
      & {B} \ar[d]\\
      {C} \ar[r]
      & {D}
    }% \xymatrix
  }% \vcenter
  \qquad\text{or}\qquad
  \vcenter{
    \xymatrix{
      {A} \ar[r] \ar[d]
      & {B} \ar[d]\\
      {C} \ar[r]
      & {D}
    }% \xymatrix
  }% \vcenter
  \qquad\text{or}\qquad
  \vcenter{
    \xymatrix@C=10ex{
      {A} \ar[r] \ar[d]
      & {B} \ar[d]\\
      {C} \ar[r]
      & {D}
    }% \xymatrix
  }% \vcenter
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}[column sep=tiny, row sep=tiny] A \arrow[r] \arrow[d] & B \arrow[d] \\ C \arrow[r] & D \end{tikzcd} \qquad\text{or}\qquad \begin{tikzcd} A \arrow[r] \arrow[d] & B \arrow[d] \\ C \arrow[r] & D \end{tikzcd} \qquad\text{or}\qquad \begin{tikzcd}[column sep=10ex] A \arrow[r] \arrow[d] & B \arrow[d] \\ C \arrow[r] & D \end{tikzcd}$$

For an example of the usefulness of negative column spacing: If you type

```
\newcommand{\pushout}[3]{#1\mathbin{\mathord{\amalg}_{#2}}#3}
\begin{equation}
  \label{diag:wide}
  \vcenter{
    \xymatrix@C=-12em{
      {\pushout{(A\otimes L)}
               {A\otimes K}
               {(B\otimes K)}
             } \ar[r] \ar[dr]
      & {B\otimes L}
      & {\pushout{(P\otimes L)}
                 {P\otimes M}
                 {(Q\otimes M)}
             } \ar[l] \ar[dl]\\
      & {\pushout{\bigl(\pushout{(A\otimes L)}
                                {A\otimes K}
                                {(B\otimes K)}\bigr)}
                 {\pushout{(X\otimes L)}
                          {X\otimes K}
                          {(Y\otimes K)}}
                 {\bigl(\pushout{(P\otimes L)}
                                {P\otimes M}
                                {(Q\otimes M)}\bigr)}
             } \ar[u]
    }
  }
\end{equation}
```

then you'll get

$$\begin{tikzcd}
(A\otimes L)\amalg_{A\otimes K}(B\otimes K) \arrow[r] \arrow[dr] & B\otimes L & (P\otimes L)\amalg_{P\otimes M}(Q\otimes M) \arrow[l] \arrow[dl] \\
& \big((A\otimes L)\amalg_{A\otimes K}(B\otimes K)\big)\amalg_{(X\otimes L)\amalg_{X\otimes K}(Y\otimes K)}\big((P\otimes L)\amalg_{P\otimes M}(Q\otimes M)\big) \arrow[u] &
\end{tikzcd} \tag{8.3}$$

which would never fit on the page without negative column spacing.

8.2. **Shifting nodes sideways.** You can shift a node to the right by enclosing it in braces and preceding the opening brace with `*+[r]` and shift it to the left by enclosing it in braces and preceding the opening brace with `*+[l]`. This can be useful, for example, if you want a node to consist of both the name of something and its definition, but you want the arrows to the node to point to the name, rather than to the center of the node. For example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar[r] \ar[d]
    & *+[r]{B = \{x \in W \mid f(x) > 0\}} \ar[d]\\
    {C} \ar[r]
    & {D}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \arrow[r] \arrow[d] & B = \{x\in W \mid f(x) > 0\} \arrow[d] \\
C \arrow[r] & D
\end{tikzcd}$$

Note that the large node at the upper right caused the columns to be far apart, even though that node was shifted to the right before being placed onto the page. To counteract that, you can decrease the column spacing: If you type

```
\begin{displaymath}
  \xymatrix@C=-2em{
    {A} \ar[r] \ar[d]
    & *+[r]{B = \{x \in W \mid f(x) > 0\}} \ar[d]\\
    {C} \ar[r]
    & {D}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \arrow[r] \arrow[d] & B = \{x\in W \mid f(x) > 0\} \arrow[d] \\
C \arrow[r] & D
\end{tikzcd}$$

8.3. **Arrows passing under nodes; crossing arrows.** Arrows can be created that go to a sequence of nodes, passing under (i.e., leaving a small gap at) all the intermediate nodes. If an arrow is created that passes under an empty node while a second arrow crosses that node in the normal way, the effect is that the first arrow passes under the second arrow. (For another way to have one arrow pass under another, see section 8.5.)

To draw an arrow that passes under a sequence of nodes and then goes on to a final node, the `\ar` is followed by `’[node]` for each node that you pass under, followed by `[finalnode]`. For example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar ’[r] ’[rr] [rrr]
    & {B}
    & {}
    & {D}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \arrow[r, dash] & B \arrow[r, dash] & {} \arrow[r] & D
\end{tikzcd}$$

For an example of crossing arrows, if you type

```
\begin{displaymath}
  \xymatrix@=3ex{
    {A} \ar[rr] \ar[dd] \ar’[dr][ddrr]
    && {B} \ar[dd] \ar[ddll]\\
    & {}\\
    {C} \ar[rr]
    && {D}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \arrow[rr] \arrow[dd] \arrow[ddrr] & & B \arrow[dd] \arrow[ddll] \\
& {} & \\
C \arrow[rr] & & D
\end{tikzcd}$$

Note that that diagram has three rows and three columns, but nothing appears in either the second row or the second column. (For an explanation of the command `@=3ex`, which changes the size of the diagram, see section 8.1.) For a more elaborate example, if you type

```
\begin{displaymath}
  \xymatrix@=2ex{
    {A} \ar[rr] \ar[dr] \ar[dd]
    && {B} \ar[dr] \ar’[d][dd]\\
    & {A’} \ar[rr] \ar[dd]
    && {B’} \ar[dd]\\
    {C} \ar’[r][rr] \ar[dr]
    && {D} \ar[dr]\\
    & {C’} \ar[rr]
    && {D’}
    }
\end{displaymath}
```

then you’ll get

$$\begin{tikzcd}
A \arrow[rr] \arrow[dr] \arrow[dd] & & B \arrow[dr] \arrow[dd] & \\
& A' \arrow[rr] \arrow[dd] & & B' \arrow[dd] \\
C \arrow[rr] \arrow[dr] & & D \arrow[dr] & \\
& C' \arrow[rr] & & D'
\end{tikzcd}$$

8.4. **Labeling the arrows.** It’s possible to label an arrow, on one or both sides of the arrow. (It’s also possible to have the label “break” the arrow; for this, see section 8.4.4.) By default, the label is located halfway from the center of the source to the center of the target. This will often be halfway along the arrow, but not if the source and the target are of different sizes. There are also options to place the label along the midpoint of the arrow (see section 8.4.1) or at an arbitrary point between the center of the source and the center of the target (see section 8.4.2).

To put a label above an arrow (where “above” means when the paper is oriented so that the arrow goes left to right), you type `^{thelabel}` either before or after the target. To put a label below an arrow (where “below” means when the paper is oriented so that the arrow goes left to right), you type `_{thelabel}` either before or after the target. Thus, if you type

```
\begin{displaymath}
  \xymatrix{
    {A}\ar[r]^{f} \ar[d]_{g}
    &{B} \ar[d]^{h}\\
    {C} \ar[r]_{k}^{\text{above}}
    &{D}
  }
\end{displaymath}
```

then you’ll get

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "g"'] & B \arrow[d, "h"] \\
C \arrow[r, "\text{above}", "k"'] & D
\end{tikzcd}$$

8.4.1. *Centering the labels on the arrows.* If you type

```
\begin{displaymath}
  \xymatrix{
    {X\cup_{Y}Z} \ar[r]^{f} \ar[d]_{g}
    &{B} \ar[d]^{h}\\
    {C} \ar[r]_{k}
    &{D}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
X \cup_Y Z \arrow[r, "f"] \arrow[d, "g"'] & B \arrow[d, "h"] \\
C \arrow[r, "k"'] & D
\end{tikzcd}$$

Note that the label f is halfway from the center of the arrow's source to the center of the arrow's target, but it is not centered along the arrow. To have that label centered along the arrow, we insert a `-` immediately following the `^`, so that we type it as

```
\ar[r]^-{f}
```

and we then get

$$\begin{tikzcd}
X \cup_Y Z \arrow[r, "f"] \arrow[d, "g"'] & B \arrow[d, "h"] \\
C \arrow[r, "k"'] & D
\end{tikzcd}$$

For another example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A \amalg_{B} C} \ar[r]^{\text{Add}}_{\sigma} \ar[d]
    & {X} \ar[d]^{\psi}\\
    {Y} \ar[r]^{\alpha}_{\text{Lifted}} \ar[r]
    & {P\times_{Q}R}
  }
\end{displaymath}
```

you'll get

$$\begin{tikzcd}
A \amalg_B C \arrow[r, "\text{Add}", "\sigma"'] \arrow[d] & X \arrow[d, "\psi"] \\
Y \arrow[r, "\alpha", "\text{Lifted}"'] & P \times_Q R
\end{tikzcd}$$

but if you type

```
\begin{displaymath}
  \xymatrix{
    {A \amalg_{B} C} \ar[r]^-{\text{Add}}_-{\sigma} \ar[d]
    & {X} \ar[d]\\
    {Y} \ar[r]^-{\alpha}_-{\text{Lifted}}
    & {P\times_{Q}R}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \amalg_B C \arrow[r, "\text{Add}", "\sigma"'] \arrow[d] & X \arrow[d] \\
Y \arrow[r, "\alpha", "\text{Lifted}"'] & P \times_Q R
\end{tikzcd}$$

8.4.2. *Arbitrary placement of labels.* In addition to the possibility of centering a label along an arrow (see section 8.4.1), it is possible to place a label at an arbitrary point between the center of the source and the center of the target. If `a` is a number between 0 and 1, then you can place a label `a` of the way from the center of the source to the center of the target by typing `(a)` immediately following the `^` or `_`. For example, if you type

```
\begin{displaymath}
  \xymatrix@C=8em{
    {A} \ar[r]^(.3){f} \ar[d]
    & {B} \ar[d]\\
    {C} \ar[r]^{g} \ar[d]
    & {D} \ar[d]\\
    {E} \ar[r]_(.7){h}
    & {F}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}[column sep=8em]
A \arrow[r, "f" pos=0.3] \arrow[d] & B \arrow[d] \\
C \arrow[r, "g"] \arrow[d] & D \arrow[d] \\
E \arrow[r, "h"' pos=0.7] & F
\end{tikzcd}$$

For a more elaborate example, with labels along segments of segmented arrows, see diagram 8.4.

8.4.3. *Labeling each segment of a segmented arrow.* An arrow that uses the "arrows passing under" feature (see section 8.3) will be in several segments, and a label can be attached to each of the segments. For example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar ’[r]^{f} ’[rr]^{g} [rrr]^{h}
    & {B}
    & {}
    & {D}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \arrow[r, dash, "f"] & B \arrow[r, dash, "g"] & {} \arrow[r, "h"] & D
\end{tikzcd}$$

For a more elaborate example, if you type

```
\begin{equation}
  \label{diag:LblCube}
  \vcenter{
    \xymatrix@=2ex{
      {A} \ar[rr]^{f} \ar[dr] \ar[dd]_{i}
      && {B} \ar[dr] \ar’[d][dd]^(.3){j}\\
      & {A’} \ar[rr]^(.3){f’} \ar[dd]^(.25){i’}
      && {B’} \ar[dd]^{j’}\\
      {C} \ar’[r]^{g}[rr] \ar[dr]
      && {D} \ar[dr]\\
      & {C’} \ar[rr]_{g’}
      && {D’}
    }
  }
\end{equation}
```

then you’ll get

$$\begin{tikzcd}
A \arrow[rr, "f"] \arrow[dr] \arrow[dd, "i"'] & & B \arrow[dr] \arrow[dd, "j" near end] & \\
& A' \arrow[rr, "f'" near start] \arrow[dd, "i'" near start] & & B' \arrow[dd, "j'"] \\
C \arrow[rr, "g"] \arrow[dr] & & D \arrow[dr] & \\
& C' \arrow[rr, "g'"'] & & D'
\end{tikzcd} \tag{8.4}$$

8.4.4. *Breaking an arrow with a label.* Instead of placing a label to the side of an arrow, you can have the label “break” the arrow. For this, you use the vertical bar character `|` in place of either `^` or `_`.

For example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar[r]|{f}
    & {B} \ar[r]|{g}
    & {C}
  }
\end{displaymath}
```

then you’ll get

$$\begin{tikzcd}
A \arrow[r, "f" description] & B \arrow[r, "g" description] & C
\end{tikzcd}$$

A label that breaks an arrow can be positioned anywhere along the arrow, just as for labels placed alongside the arrow (see sections 8.4.1 and 8.4.2). For example, if you type

```
\begin{equation*}
  \xymatrix{
    {A \amalg_{B} C} \ar[r]|-{f} \ar[d]_{i}
    & {X} \ar[r]|(.7){m} \ar[d]^{p}
    & {Y} \ar[dl]|(.3)n\\
    {D} \ar[r]|-{g}
    & {W \times_{Z} Y}
  }
\end{equation*}
```

then you'll get

$$\begin{tikzcd}
A \amalg_B C \arrow[r, "f" description] \arrow[d, "i"'] & X \arrow[r, "m" description] \arrow[d, "p"] & Y \arrow[dl, "n" description] \\
D \arrow[r, "g" description] & W \times_Z Y &
\end{tikzcd}$$

8.5. **More crossing arrows.** In section 8.3 we presented a method of having an arrow pass over another arrow by having them cross at an empty node. We present here another method of having one arrow pass over another.

To have one arrow appear to cross over another, we create a small gap in the second arrow at the spot at which the first arrow will cross. We do this by breaking that second arrow with a label, as in section 8.4.4, using the special label `\hole`, which just leaves an empty hole in the arrow. For example, if you type

```
\begin{equation*}
  \xymatrix{
    {A} \ar[dr]|{\hole}
    & {B} \ar[dl]\\
    {C}
    & {D}
  }
\end{equation*}
```

then you'll get

$$\begin{tikzcd}
A \arrow[dr] & B \arrow[dl, crossing over] \\
C & D
\end{tikzcd}$$

For another example, if you type

```
\begin{equation*}
  \xymatrix{
    {A} \ar[dr] \ar[r] \ar[d]
    & {B} \ar[r]
    & {C}\\
    {D} \ar[urr]|(.33){\hole} \ar[r]
    & {E}
  }
\end{equation*}
```

then you'll get

$$\begin{tikzcd}
A \arrow[r] \arrow[d] & B \arrow[r] & C \\
D \arrow[urr] \arrow[r] & E \arrow[from=ul, crossing over] &
\end{tikzcd}$$

(see section 8.4.2). It took some trial and error to figure out that `.33` was the number to use in the arrow with the hole, but there's a much better way specify the location of the hole: Instead of inserting that number between 0 and 1 that specifies how far from the center of the source to the center of the target the break should appear, you can specify the place to put the break using the notation `!{[node1];[node2]}` to denote the point on the arrow at which a straight line from [node1] to [node2] would cross the arrow. (*Note:* This method only works for straight line arrows.) That is, if you type

```
\begin{equation*}
  \xymatrix{
    {A} \ar[dr] \ar[r] \ar[d]
    & {B} \ar[r]
    & {C}\\
    {D} \ar[urr]|!{[u];[r]}{\hole} \ar[r]
    & {E}
  }
\end{equation*}
```

then you'll get

$$\begin{tikzcd}
A \arrow[r] \arrow[d] & B \arrow[r] & C \\
D \arrow[urr] \arrow[r] \arrow[from=1-1, to=2-2, crossing over] & E &
\end{tikzcd}$$

The position of the hole is specified as !{[u];[r]} because the arrow with the hole begins at D and, relative to that node, the arrow that crosses over goes from the node [u] to the node [r].

For another example, if you type

```
\begin{equation*}
  \xymatrix{
    {A} \ar[d] \ar[dr]
    & {B} \ar[d] \ar[dr]
    & {C} \ar[d]\\
    {D}
    \ar[urr]|!{[u];[r]}{\hole}
            |!{[ur];[r]}{\hole}
            |!{[ur];[rr]}{\hole}
    & {E}
    & {F}
  }
\end{equation*}
```

then you'll get

$$\begin{tikzcd}
A \arrow[d] & B \arrow[d] & C \arrow[d] \\
D \arrow[urr] \arrow[from=1-1, to=2-2, crossing over] \arrow[from=1-2, to=2-2, crossing over] \arrow[from=1-2, to=2-3, crossing over] & E & F
\end{tikzcd}$$

8.6. **Making room for large labels.** Although XY-pic automatically leaves enough room for large nodes, it ignores the size of the labels on the arrows when constructing the diagram. Thus, if you have a particularly large label, it may overwrite the nodes if you don't do something to make the diagram larger. You can do that either by enlarging the entire diagram (see section 8.6.1) or by creating a row (or column) all of whose nodes are empty and have the arrows span the empty row (or column) (see section 8.6.2).

8.6.1. *Enlarging the entire diagram.* To enlarge the entire diagram, we use the spacing commands described in Section 8.1. For example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A\amalg B} \ar[d]
    \ar[r]^-{\operatorname{amalgamate}}
    & {X} \ar[r] \ar[d]
    & {P} \ar[d]\\
    {C \amalg D}
    \ar[r]_-{\operatorname{amalgamate}}
    & {Y} \ar[r]
    & {Q}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \amalg B \arrow[r, "\operatorname{amalgamate}"] \arrow[d] & X \arrow[r] \arrow[d] & P \arrow[d] \\
C \amalg D \arrow[r, "\operatorname{amalgamate}"'] & Y \arrow[r] & Q
\end{tikzcd}$$

but if you type

```
\begin{displaymath}
  \xymatrix@C=6em{
    {A\amalg B} \ar[d] \ar[r]^-{\operatorname{amalgamate}}
    & {X} \ar[r] \ar[d]
    & {P} \ar[d]\\
    {C \amalg D} \ar[r]_-{\operatorname{amalgamate}}
    & {Y} \ar[r]
    & {Q}
  }
\end{displaymath}
```

(see Section 8.1) then you'll get

$$\begin{tikzcd}[column sep=6em]
A \amalg B \arrow[r, "\operatorname{amalgamate}"] \arrow[d] & X \arrow[r] \arrow[d] & P \arrow[d] \\
C \amalg D \arrow[r, "\operatorname{amalgamate}"'] & Y \arrow[r] & Q
\end{tikzcd}$$

If you don't want to have to make the entire diagram larger, you can just create a column of empty nodes (see Section 8.6.2).

8.6.2. *Creating a column of empty nodes.* We continue the discussion of Section 8.6.1: Instead of enlarging the entire diagram, you can create an additional column in between the first two columns and have all the nodes in the new column be empty. For example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A\amalg B} \ar[d] \ar[rr]^-{\operatorname{amalgamate}}
    && {X} \ar[r] \ar[d]
    & {P} \ar[d]\\
    {C \amalg D} \ar[rr]_-{\operatorname{amalgamate}}
    && {Y} \ar[r]
    & {Q}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \amalg B \arrow[rr, "\operatorname{amalgamate}"] \arrow[d] & & X \arrow[r] \arrow[d] & P \arrow[d] \\
C \amalg D \arrow[rr, "\operatorname{amalgamate}"'] & & Y \arrow[r] & Q
\end{tikzcd}$$

Note that that diagram has four columns, but both of the nodes in the second column are empty, and the labelled arrows are created as `\ar[rr]`. If you'd like to create even more space for that large label, you can put an `\hspace` command into one of the nodes in the second column. For example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A\amalg B} \ar[d] \ar[rr]^-{\operatorname{amalgamate}}
    & {\hspace{3em}}
    & {X} \ar[r] \ar[d]
    & {P} \ar[d]\\
    {C \amalg D} \ar[rr]_-{\operatorname{amalgamate}}
    && {Y} \ar[r]
    & {Q}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \amalg B \arrow[rr, "\operatorname{amalgamate}"] \arrow[d] & \hspace{3em} & X \arrow[r] \arrow[d] & P \arrow[d] \\
C \amalg D \arrow[rr, "\operatorname{amalgamate}"'] & & Y \arrow[r] & Q
\end{tikzcd}$$

Note that we only put the `\hspace` command into the first row; there's no need to put it into every row.

8.7. **Different arrow styles.** It's possible to have arrows with tails, multiple heads, dotted, dashed, or multiple shafts, and any combination of these (see Table 1). You can even omit both the head and the tail, or omit the arrow entirely, which is useful for placing things into the diagram in places outside of the grid of nodes (see Diagram 8.5).

All of these arrows can point in whatever direction you choose; we used `[r]` in the table just for readability. For an example of the use of invisible arrows: If you type

```
\begin{equation}
  \label{diag:CentEq}
  \vcenter{
```

To produce:	Type:
$\cdots\cdots\rightarrow$	`\ar@{.>}[r]`
$-\ -\ \rightarrow$	`\ar@{-->}[r]`
$\Longrightarrow$	`\ar@{=>}[r]`
$\vdots\!\vdots\!\vdots\Rightarrow$	`\ar@{:>}[r]`
$\longrightarrow\!\!\!\!\twoheadrightarrow$	`\ar@{->>}[r]`
$\rightarrowtail\!\!\longrightarrow$	`\ar@{>->}[r]`
$\longleftrightarrow$	`\ar@{<->}[r]`
$-\ -\ \twoheadrightarrow$	`\ar@{-->>}[r]`
$\succ\!-\ -\ \rightarrow$	`\ar@{>-->}[r]`
$\leftarrow\ -\ \rightarrow$	`\ar@{<-->}[r]`
$\Longrightarrow\!\!\!\!\Rightarrow$	`\ar@{=>>}[r]`
$\Longleftrightarrow$	`\ar@{<=>}[r]`
$\vdots\!\vdots\!\vdots\Rightarrow\!\!\!\!\Rightarrow$	`\ar@{:>>}[r]`
$\Leftarrow\!\vdots\!\vdots\!\vdots\Rightarrow$	`\ar@{<:>}[r]`
$\relbar\joinrel\relbar\joinrel\relbar$	`\ar@{-}[r]`
$\cdots\cdots$	`\ar@{.}[r]`
$=\!=\!=$	`\ar@{=}[r]`
	`\ar@{}[r]`

TABLE 1. Arrow Styles

```
    \xymatrix{
      {A} \ar@{.>}[r] \ar@{.>}[d] \ar@{}[dr]|{=}
      & {B} \ar[d]\\
      {C} \ar[r]
      & {D}
    }
  }
\end{equation}
```

then you'll get

$$\begin{tikzcd}
A \arrow[r, dotted] \arrow[d, dotted] \arrow[dr, phantom, "="] & B \arrow[d] \\
C \arrow[r] & D
\end{tikzcd} \tag{8.5}$$

Note that the equals sign is inserted by having it break the middle of the invisible arrow from the upper left node to the lower right node.

8.8. **Curved arrows.** It's possible to have arrows curve, either by specifying the amount that they curve or by specifying the direction in which they leave their source and the direction from which they arrive at their target.

8.8.1. *Specifying the amount of the curve.* To have an arrow curve in the up direction (where "up" means when the paper is oriented so that the arrow goes left to right) by a default amount you follow the `\ar` with `@/^/`; that is, you type `\ar@/^/[target]`. To have it curve in the down direction by a default amount you follow the `\ar` with `@/_/`; that is, you type `\ar@/_/[target]` (or, if you want to label the arrow, `\ar@/_/[target]^{label}`). Thus, if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar@/^/[drr]^{p} \ar@{.>}[dr]|{\exists!} \ar@/_/[ddr]_{q}\\
    & {B} \ar[r] \ar[d]
    & {C} \ar[d]\\
    & {D} \ar[r]
    & {E}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \arrow[rrd, bend left, "p"] \arrow[rd, dotted, "\exists!" description] \arrow[rdd, bend right, "q"'] & & \\
& B \arrow[r] \arrow[d] & C \arrow[d] \\
& D \arrow[r] & E
\end{tikzcd}$$

If you'd like to specify the amount of curve, you can specify a dimension following the `^` or the `_`, as in `\ar@/^1ex/[target]`. (Although you can use whatever units you like for `dimen` (e.g., inches (`in`), centimeters (`cm`), points (`pt`), etc.), it's a good idea to measure vertical distances in `ex` (roughly the height of an upper case "X") and horizontal distances in `em` (roughly the width of an upper case "M"), so that the distance will be roughly correct even if you change font sizes.) For example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar@/^3ex/[drr]^{p} \ar@{.>}[dr]|{\exists!}
        \ar@/_3ex/[ddr]_{q}\\
    & {B} \ar[r] \ar[d]
    & {C} \ar[d]\\
    & {D} \ar[r]
    & {E}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \arrow[rrd, bend left=45, "p"] \arrow[rd, dotted, "\exists!" description] \arrow[rdd, bend right=45, "q"'] & & \\
& B \arrow[r] \arrow[d] & C \arrow[d] \\
& D \arrow[r] & E
\end{tikzcd}$$

and if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar@{>->}[r]^{i}
    & {B} \ar@{->>}[r]^{p}
    & {C} \ar@{.>}@/_4ex/[l]_{s}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \arrow[r, tail, "i"] & B \arrow[r, two heads, "p"] & C \arrow[l, dotted, bend right, "s"']
\end{tikzcd}$$

8.8.2. *Specifying the start and end directions.* To specify the start and end directions of a curved arrow you type `\ar@(start,end)[target]` where "`start`" is the direction towards which the arrow begins and "`end`" is the direction from which the arrow ends. For example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar[dr]|{p} \ar@(r,u)[dr]^{f} \ar@(d,l)[dr]_{g}\\
    & {B}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \arrow[dr, "p" description] \arrow[dr, out=0, in=90, "f"] \arrow[dr, out=270, in=180, "g"'] & \\
& B
\end{tikzcd}$$

and if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar@(ur,ul)[rr]^{1_{A}} \ar[r]_{f}
    & {B} \ar[r]_{g} \ar@(dr,dl)[rr]_{1_{B}}
    & {A} \ar[r]_{f}
    & {B}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \arrow[r, "f"'] \arrow[rr, bend left, "1_A"] & B \arrow[r, "g"'] \arrow[rr, bend right, "1_B"'] & A \arrow[r, "f"'] & B
\end{tikzcd}$$

Since the current node is denoted `[]` (i.e., a pair of square brackets with nothing inside), if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar@(ur,dr)[]
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \arrow[loop right]
\end{tikzcd}$$

8.9. **Sliding arrows sideways; multiple arrows.** It's possible to slide arrows sideways, so that you can have more than one straight line arrow between a single pair of nodes. To do this, you type `\ar@<dimen>` to move the arrow "upwards" by `dimen` (where "upwards" means when the paper is held so that the arrow goes from left to right). (A negative `dimen` will move the arrow "downwards".) Although you can use whatever units you like for `dimen` (e.g., inches (`in`), centimeters (`cm`), points (`pt`), etc.), it's a good idea to measure vertical distances in `ex` (roughly the height of an upper case "X") and horizontal distances in `em` (roughly the width of an upper case "M"), so that the distance will be roughly correct even if you change font sizes. For example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar@<1ex>[r]
    & {B} \ar[r]
    & {C} \ar@<-1ex>[r]
    & {D}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd} A \arrow[r, shift left] & B \arrow[r] & C \arrow[r, shift right] & D \end{tikzcd}$$

For another example, if you type

```
\begin{displaymath}
  \xymatrix{
    {\coprod_{i \in I} A_{i}\quad}
      \ar@<+.7ex>[r]^-{\phi} \ar@<-.7ex>[r]_-{\psi}
    & {\quad\coprod_{i \in I} B_{i}}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd} \coprod_{i \in I} A_{i}\quad \arrow[r, shift left, "\phi"] \arrow[r, shift right, "\psi"'] & \quad\coprod_{i \in I} B_{i} \end{tikzcd}$$

It's also possible to slide curved arrows sideways. For example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar@/^ 1ex/[r] \ar@/^ 1ex/@<1ex>[r]
    & {B}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd} A \arrow[r, bend left=10] \arrow[r, bend left=10, shift left] & B \end{tikzcd}$$

8.10. **Specifying both the source and the target of an arrow.** Up until now, all of our examples of arrows have gone from the current node to a node specified in the code for the arrow. It's possible, however, to specify both the source and the target of an arrow, rather than letting the source be the current node.

To draw an arrow who's source is not at the current node, you type

```
[source];[target]
```

where you would have typed just `[target]`. Since the current node is denoted `[]`, typing "`\ar[target]`" is the same as typing "`\ar[];[target]`".

For example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A}
    & {B}\\
    {C}
    & {D}
    \ar^{f}[ul];[u]
    \ar_{g}[l];[]
    \ar_{h}[ul];[l]
    \ar^{k}[u];[]
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "h"'] & B \arrow[d, "k"] \\
C \arrow[r, "g"'] & D
\end{tikzcd}$$

Note that when we drew the arrows the current node was D, and so the target of, e.g., the arrow labelled g was specified as `[]` (which denotes the current node).

8.11. **Text objects.**

You can insert a text object into a diagram by typing

`*\txt{Some text}` or `*\txt<width>{Some text}`

In the first form you create linebreaks with a double backslash "`\\`" and in the second form the lines are broken automatically. Text objects by default don't have any blank space around them, but you can add a bit of space by typing them as `*+\txt{Some text}`, or even more space by typing them as `*++\txt{Some text}`.

For example, if you type

```
\begin{displaymath}
  \xymatrix{
    *+\txt{Roses are red,\\ Violets are blue.\\
      This is a contrived example,\\ And the rhyme is bad too.}
    \ar@{=>}[r]
    & *\txt<2in>{Four score and seven years ago our fathers brought
      forth on this continent a new nation, conceived in liberty,
      and dedicated to the proposition that all men are created
      equal.}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
\begin{array}{c}\text{Roses are red,}\\ \text{Violets are blue.}\\ \text{This is a contrived example,}\\ \text{And the rhyme is bad too.}\end{array} \arrow[r, Rightarrow] & \begin{array}{c}\text{Four score and seven years ago}\\ \text{our fathers brought forth on}\\ \text{this continent a new nation,}\\ \text{conceived in liberty, and}\\ \text{dedicated to the proposition}\\ \text{that all men are created equal.}\end{array}
\end{tikzcd}$$

We typed that first one as `*+\txt` so that the arrow wouldn't start too close to the text.

You can also shift a text object to the left (so that arrows to or from the object will go to its right end instead of its center) by typing it as `*+[l]\txt{Some text}` or to the right (so that arrows to or from the object will go to its left end instead of its center) by typing it as `*+[r]\txt{Some text}`. For example, if you type

```
\begin{displaymath}
  \xymatrix{
    *+[l]\txt{Interior} \ar[r] \ar[d]
    & *+[r]\txt{Exterior} \ar[d]\\
    {A} \ar[r]
    & {B}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
\text{Interior} \arrow[r] \arrow[d] & \text{Exterior} \arrow[d] \\
A \arrow[r] & B
\end{tikzcd}$$

Text objects are mostly used for adding annotations to a diagram (see section 8.13).

8.12. Arrows off of the grid.

It's possible to draw arrows that begin or end at points other than nodes. To do this, you specify the point by choosing a node and then adding a displacement vector "`+<Dx,Dy>`" from that node (where `Dx` is the horizontal displacement and `Dy` is the vertical displacement). (Although you can use whatever units you like for the displacements (e.g., inches (`in`), centimeters (`cm`), points (`pt`), etc.), it's a good idea to measure vertical distances in `ex` (roughly the height of an upper case "X") and horizontal distances in `em` (roughly the width of an upper case "M"), so that the displacement will be roughly correct even if you change font sizes.) For example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar[d]+<4em,0ex>\\
    {B}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \arrow[dr] & \\
B & {}
\end{tikzcd}$$

since the arrow is pointing to the spot 4em to the right of the B.

For another example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A}
    & {B}\\
    {C}
    & {D}
    \ar[ul]+<-2em,0ex>;[]+<2em,0ex>
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
{} \arrow[drrr] & A & B & \\
& C & D & {}
\end{tikzcd}$$

(see section 8.10) because we specified the source as being `[ul]+<-2em,0ex>`, i.e., 2em to the left of the node that's one up and one to the left of the current node (D) and the target as being `[]+<2em,0ex>`, i.e., 2em to the right of the current node (D).

For an example that combines displacements of source and target along with curving arrows and labelled arrows: if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar[r]+<0em,-3ex>^{f}
    & {B} \ar@(ur,ur)[d]+<2em,0ex>^{g}\\
    {C}
    & {D} \ar@/^ 2ex/[l];[]^{h}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd}
A \arrow[r, "f"] & B \arrow[d, bend left=90, "g"] \\
C \arrow[r, bend left, "h"] & D
\end{tikzcd}$$

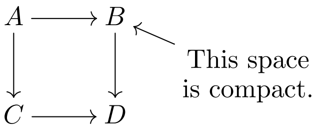
Realistic examples of arrows off of the grid can be found in section 8.13, where we show how to drop objects into a diagram at either the source or the target of an arrow.

8.13. **Dropping objects into a diagram; annotations.** You can drop an object at an arbitrary point of a diagram (i.e., not necessarily at a node) by drawing an arrow to or from that point (see section 8.12) and dropping the object at one of the ends of the arrow. The object can be either mathematics or a text object (see section 8.11).

For example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar[r] \ar[d]
    & {B} \ar[d]
    \ar[]+<5em,-5ex>*+\txt{This space\\is compact.};[]\\
    {C} \ar[r]
    & {D}
  }
\end{displaymath}
```

then you'll get



Note that the text and the arrow from it was typed by first creating the arrow

```
\ar[]+<5em,-5ex>;[]
```

whose source is displaced from the current node (see sections 8.10 and 8.12) and whose target is the current node, and then dropping the text object (see section 8.11) at the source by inserting

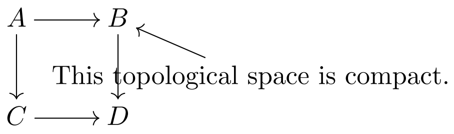
```
*+\txt{This space\\is compact.}
```

immediately after the specification of the source.

Note that the arrows to or from a text object by default go to its center, and if a text object is dropped at a specified point by default it's the center of the text object that lands at that point. Thus, if you're typing a wide text object, you should be sure either to leave enough room so that it doesn't overlap other parts of the diagram or to shift the text object sideways (as in section 8.2). For example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar[r] \ar[d]
    & {B} \ar[d]
    \ar[]+<5em,-5ex>*+\txt{This topological space is compact.};[]\\
    {C} \ar[r]
    & {D}
  }
\end{displaymath}
```

then you'll get



but if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar[r] \ar[d]
    & {B} \ar[d]
    \ar[]+<10em,-5ex>*+\txt{This topological space is compact.};[]\\
    {C} \ar[r]
    & {D}
  }
\end{displaymath}
```

(in which we increased the horizontal displacement to 10em) then you'll get

$$\begin{tikzcd}
A \arrow[r] \arrow[d] & B \arrow[d] & & \\
C \arrow[r] & D & & \text{This topological space is compact.} \arrow[llu]
\end{tikzcd}$$

and if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar[r] \ar[d]
    & {B} \ar[d]
    \ar[]+<5em,-5ex>*+[r]\txt{This topological space is compact.};[]\\
    {C} \ar[r]
    & {D}
  }
\end{displaymath}
```

(in which we typed

```
*+[r]\txt{This topological space is compact.}
```

to create the text object, i.e., we added "`[r]`" to shift the text object to the right; see section 8.2) then you'll get

$$\begin{tikzcd}
A \arrow[r] \arrow[d] & B \arrow[d] & \\
C \arrow[r] & D & \text{This topological space is compact.} \arrow[lu]
\end{tikzcd}$$

If you want to drop an object onto a spot other than at a node without printing an arrow to or from that spot, you can draw an invisible arrow going to that spot and drop the object at the end of the invisible arrow. For example, if you type

```
\begin{displaymath}
  \xymatrix{
    {A} \ar[r] \ar[d]_{f}
    & {B} \ar[d]^{g}
    \ar@{}[]+<8em,-4ex>*+\txt{Both $f$ and $g$\\are fibrations.}\\
    {C} \ar[r]
    & {D}
  }
\end{displaymath}
```

then you'll get

$$\begin{tikzcd} A \arrow[r] \arrow[d, "f"'] & B \arrow[d, "g"] & \text{Both f and g} \\ C \arrow[r] & D & \text{are fibrations.} \end{tikzcd}$$

The arrow to the text object is invisible because we typed it starting with `\ar@{}` (see section 8.7 and Diagram 8.5). After that, we typed

```
[]+<8em,-4ex>*+\txt{Both $f$ and $g$\\are fibrations.}
```

so that the arrow would go to the point 8em to the right and 4ex below the current node, and we dropped the text object at that point.

9. Including graphics

If you have graphics files that you want to include in your document, then you should load the `graphicx` package by putting the command

```
\usepackage{graphicx}
```

after your `\documentclass` command. (Note: The spelling `graphicx` is correct; the package graphicx.sty is the updated version of the older package graphics.sty.) You can then include graphics using the `\includegraphics` command.

The standard practice is to include graphics inside of a `figure` environment, so that it can float to the next page if there isn't enough room for it on the current page. This also allows you to give a caption (printed below the graphics) and a label (for use in a `\ref` command). Thus, if you want to include the contents of `mygraphic.eps`, you might type

```
\begin{figure}[h]
  \centering
  \includegraphics{mygraphics.eps}
  \caption{A pretty picture}
  \label{fig:pretty}
\end{figure}
```

and you can then refer to it as `figure~\ref{fig:pretty}`. (The `\centering` command centers the graphics in the figure; this is preferable to using a `center` environment, which would add additional vertical space.) It's important to note that the `\label` command must come *after* the `\caption` command, since it's the `\caption` command that creates the number that the `\label` command labels. You can also put the `\label` command *inside* the caption, as in

```
\caption{A pretty picture\label{fig:pretty}}
```

and some people find that this helps prevent errors.

9.1. Graphic formats: Postscript or pdf.

One problem to watch out for is that different versions of LaTeX require different types of graphics files:

- Standard `latex` (which produces dvi files) requires encapsulationed postscript graphics files, while
- `pdflatex` (which directly produces pdf files) requires pdf graphics files.

This can be confusing, because some installations of LaTeX give you a `latex` command that is actually running `pdflatex`. In particular, the standard installation of TeXShop for Mac OS X gives you a `latex` command that actually runs `pdflatex` and produces a pdf file. TeXShop can be configured to run standard `latex` to produce dvi files and then convert them to pdf; for this, see the list of frequently asked questions at

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

and search that page for "texshop".

On most systems, if you need to include encapsulated postscript graphics and produce a pdf file, you can use plain old `latex` to produce a dvi file and then use `dvipdf` to create the pdf file. Another option is to use the `epstopdf` program to create a postscript file from your eps file.

10. Macro definitions, a.k.a. \newcommand

LaTeX allows you to use the same `\def` command that you use in plain TeX, but it's considered bad style. Instead, LaTeX has the `\newcommand` and `\renewcommand` commands, which do a little error checking for you. In plain TeX, you might use the command

```
\def\tensor{\otimes}
```

but in LaTeX the preferred form is

```
\newcommand{\tensor}{\otimes}
```

The advantage of this is that LaTeX will check to see if there already is a command with the name `\tensor` and give you an error message if there is. If you know that there is a previous definition of `\tensor` but you *want* to override it, then you use the command

```
\renewcommand{\tensor}{\otimes}
```

If you want to use macros with replaceable parameters, both `\newcommand` and `\renewcommand` allow this. For the equivalent of the plain TeX command

```
\def\pushout#1#2#3{#1\cup_{#2}#3}
```

you use the LaTeX command

```
\newcommand{\pushout}[3]{#1\cup_{#2}#3}
```

i.e., the command name is enclosed in braces, and the number of parameters is enclosed in square brackets.

11. Lists: itemize, enumerate, and description

There are three list making environments:

- `itemize`,
- `enumerate`, and
- `description`.

The `itemize` environment just lists the items with a marker in front of each one. If you type

```
\begin{itemize}
\item This is the first item in the list, which runs on long enough
  to spill over onto a second line.
\item This is the second item in the list, which is a bit shorter.
\item This is the last item.
\end{itemize}
```

then you'll get

- This is the first item in the list, which runs on long enough to spill over onto a second line.
- This is the second item in the list, which is a bit shorter.
- This is the last item.

The `enumerate` environment looks the same, except that the items in the list are numbered. If you type

```
\begin{enumerate}
\item This is the first item in the list, which runs on long enough
  to spill over onto a second line.
\item This is the second item in the list, which is a bit shorter.
\item This is the last item.
\end{enumerate}
```

then you'll get

(1) This is the first item in the list, which runs on long enough to spill over onto a second line.
(2) This is the second item in the list, which is a bit shorter.
(3) This is the last item.

The `description` environment requires an argument for each `\item` command, which will be printed at the beginning of the item. If you type

```
\begin{description}
\item[sedge] A green plant, found in both wetlands and uplands.
  Sedges are often confused with grasses and rushes.
\item[grass] A green plant, found in both wetlands and uplands.
  Grasses are often confused with sedges and rushes.
\item[rush] A green plant, found in both wetlands and uplands.
  Rushes are often confused with sedges and grasses
\end{description}
```

you'll get

sedge: A green plant, found in both wetlands and uplands. Sedges are often confused with grasses and rushes.

grass: A green plant, found in both wetlands and uplands. Grasses are often confused with sedges and rushes.

rush: A green plant, found in both wetlands and uplands. Rushes are often confused with sedges and grasses

These environments can be inserted within each other, and the `enumerate` environment keeps track of what level it's at, and numbers its items accordingly. If you type

```
\begin{enumerate}
\item I went to the dry cleaners.
\item I went to the supermarket.  I bought
  \begin{enumerate}
  \item bread,
  \item cheese, and
  \item Tabasco sauce.
  \end{enumerate}
\item I went to the bank.
\end{enumerate}
```

you'll get

(1) I went to the dry cleaners.
(2) I went to the supermarket. I bought
  (a) bread,
  (b) cheese, and
  (c) Tabasco sauce.
(3) I went to the bank.

11.1. **Referring to enumerated items by number.** If you want to refer to an item in an enumerated list by item number, then you need to label the item using the `\label` command and then refer to it using the `\ref` command, just as you do for sections, subsections, theorems, etc. (see section 5). For example, if you type

```
\begin{enumerate}
\item \label{step:trans} Put the transmission into drive.
\item \label{step:brake} Release the parking brake.
\item \label{step:move} Start moving.
\end{enumerate}
You must complete steps \ref{step:trans} and \ref{step:brake}
before beginning step~\ref{step:move}.
```

then you'll get

(1) Put the transmission into drive.
(2) Release the parking brake.
(3) Start moving.

You must complete steps 1 and 2 before beginning step 3.

12. The bibliography

The recommended way to create a bibliography is to use the `amsrefs` package (see section 12.1). This makes it simple to type the bibliography items and have them correctly formatted and automatically numbered, and makes it possible to change the style of the bibliography without retyping the items. At the same time, it allows you to preserve the structure of the information contained in the items, and (optionally) to create a database of possible references that can be reused in other documents.

The `amsrefs` package is fairly new. The classical method of creating a bibliography from a database of possible references uses BibTeX, which is a separate program that must be run in between the runs of LaTeX. If you already have one or more BiBTeX databases, `amsrefs` can make use of them through a special BibTeX style file (see section 12.5).

If for some reason you want to avoid the use of `amsrefs` and of BibTeX, you can type and format the bibliography items any way you choose by creating a `thebibliography` environment (see section 12.7).

12.1. **Using `amsrefs`.** To use `amsrefs`, you load it by putting the command

```
\usepackage[lite]{amsrefs}
```

into the preamble of your document, i.e., after the `\documentclass` command and before the `\begin{document}` command. You can then

(1) begin the bibliography with the commands

```
\begin{bibdiv}
  \begin{biblist}
```

(2) define the bibliography items using a `\bib` command for each item (see section 12.2), and then

(3) end the bibliography with the commands

```
  \end{biblist}
\end{bibdiv}
```

The reason there are two different environments (the outer environment `bibdiv` containing the inner environment `biblist`) is that the `bibdiv` environment produces the chapter or section heading for the bibliography (depending on what's appropriate for the type of your document) and the `biblist` environment produces the actual list of references. You can also type material in between the `\begin{bibdiv}` and `begin{biblist}` commands and it will be printed in between the heading for the list and the actual list.

12.2. **The `\bib` command.** The information for each bibliography item inside the `bibdiv` environment (see section 12.1) is entered using a `\bib` command. The format of a `\bib` command is

```
\bib{ReferenceKey}{ReferenceType}{
  comma separated list of keyword={value} statements
}
```

where `ReferenceKey` is the key that will be used to refer to the item with a `\cite` command (see section 5.3) and `ReferenceType` is either

`article`, `book`, `misc`, `report`, or `thesis`.

For example, the bibliography of these notes contains the following `\bib` commands:

```
\bib{yellowmonster}{book}{
  author={Bousfield, A.K.},
  author={Kan, D.M.},
  title={Homotopy Limits, Completions and Localizations},
  date={1972},
  series={Lecture Notes in Mathematics},
  volume={304},
  publisher={Springer-Verlag},
  address={Berlin-New York}
}

\bib{quil:rht}{article}{
  author={Quillen, Daniel G.},
  title={Rational Homotopy Theory},
  journal={Ann. of Math. (2)},
  volume={90},
  date={1969},
  pages={205--295}
}
```

If those `\bib` commands are in your `biblist` environment, you can type

```
see \cite{yellowmonster} and \cite{quil:rht}
```

and it will be typeset as “see [4] and [10]”.

Some of the rules governing the `\bib` command are:

- The keywords must all be typed in lower case. (This is different from in BibTEX databases, in which the case of field names is ignored.)
- The value in each “`keyword={value}`” statement must always be enclosed in braces. (This is different from in BibTEX databases, in which the braces can sometimes be omitted.)
- Author names must be typed in the form “von Last, First, Jr.”, as in “`author={Jones, John Paul}`”, or “`author={van Beethoven, Ludwig}`”, or “`author={Ford, Henry, Jr.}`”. This ensures there won't be errors in reversing the first and last names, or when `amsrefs` replaces the first and middle names with initials (when the `initials` option is used; see section 12.6), or when `amsrefs` creates a label based on the last name (when the `alphabetic` option is used; see section 12.6).
- Multiple authors must each be listed in a separate `author` field.
- The capitalization in the title should be exactly as you want it to appear in the bibliography. (This is different from in BibTEX databases, in which capitalization is often changed to fit a chosen style.)
- The date must be written in the form `date={1776}`, or `date={1776-07}`, or `date={1776-07-04}`.

The full list of simple fields (i.e., fields that can appear at most once) is

> `address`, `booktitle`, `date`, `edition`, `eprint`, `hyphenation`, `journal`, `label`, `language`, `note`, `number`, `organization`, `pages`, `part`, `publisher`, `series`, `status`, `subtitle`, `title`, `type`, `volume`, and `xref`

The full list of repeatable fields is

> `author`, `editor`, `translator`, `isbn`, `issn`, and `review`.

The full list of compound fields (for which the value of each is a comma separated list of `keyword={value}` statements) is

> `book`, `conference`, `contribution`, `partial`, `reprint`, and `translation`.

A description of the less obvious ones can be found in [6, section 5.2].

12.3. **Obtaining bibliographic information from MathSciNet.** If you have access to MathSciNet (at `http://www.ams.org/mathscinet`), then you can obtain a completely filled out `\bib` command (see section 12.2) for any of the publications listed there. To do this:

- Use MathSciNet to find the Mathematical Reviews review of the publication.
- Look near the upper left corner of the page for a pulldown menu titled “Select Alternative Format”.
- Select “AMSRefs” from that menu; you'll get a new page showing the `\bib` command for the publication.
- Copy that `\bib` command and paste it into your LATEX file.

12.4. **Using a database of possible references.** If your list of `\bib` commands (see section 12.2) grows so long that you don't want to include it in your LATEX file, or if you want to reuse a collection of `\bib` commands from other papers, you can put all of your `\bib` commands into a file and have `amsrefs` extract only the ones used in your current paper. To do this, you create a file whose name ends in

.ltb, for example: myrefs.ltb, and put all of your \bib commands into that file. In such a file, it's important that each \bib command begins on a new line, and that that line contains the first two arguments and the following open brace, as in

```
\bib{ReferenceKey}{ReferenceType}{
```

You then delete all of the \bib commands from your LaTeX file and replace them with the single line

```
\bibselect{myrefs}
```

and amsrefs will automatically read myrefs.ltb and extract only those items that are cited in the current paper. Thus, if you have the \bib commands for all of your possible references in the file myrefs.ltb, and you've loaded amsrefs with the command \usepackage[lite]{amsrefs} in your preamble (see section 12.1), then you would create the bibliography with the lines

```
\begin{bibdiv}
  \begin{biblist}
    \bibselect{myrefs}
  \end{biblist}
\end{bibdiv}
```

If your LaTeX file contains the above lines and your file is named mypaper.tex, then LaTeX will create the file mypaper.bbl which will contain the \bib commands for only those references cited in your paper, and mypaper.bbl will automatically be read and used in your LaTeX file.

If your \bib commands are spread over several .ltb files, you can either list them all, separated by commas, in the argument of the \bibselect command, as in

```
\bibselect{myfirstrefs,mysecondrefs}
```

or just use multiple \bibselect commands, as in

```
\bibselect{myfirstrefs}
\bibselect{mysecondrefs}
```

As described above, a \bibselect command uses only the references actually cited in your document. To list *all* the references in an .ltb file, whether or not they're cited in your document, use the command

```
\bibselect*{myrefs}
```

12.5. **Using amsrefs with BibTeX.** If you already have a BibTeX database of possible references, you can use that by combining BibTeX with amsrefs. If your references are in the BibTeX file myrefs.bib, then you would load the amsrefs package by putting the command \usepackage[lite]{amsrefs} in your preamble (in the same way that you would load it if you were using amsrefs without BibTeX; see section 12.1) and put the single line

```
\bibliography{myrefs}
```

where you want the bibliography to appear. (That is, you don't create the bibdiv or biblist environments.) You then run BibTeX on your file in the normal way, and BibTeX will create the .bbl file. That .bbl file will contain the bibdiv and biblist environments, and LaTeX will read it the next time that you run LaTeX and create the bibliography.

If you use this method, you don't use the \bibliographystyle command (and any such command in your LaTeX file will be ignored).

12.6. **Options for the `amsrefs` package.** There are a number of optional arguments that you can use when loading the `amsrefs` package (see section 12.1), most of which affect the formatting of the bibliography. You use them by listing them as optional arguments to the `\usepackage` command (see section 12.1). For example, we suggested that you always use the optional argument `lite`, as in

```
\usepackage[lite]{amsrefs}
```

We'll describe the most important options here; the full list can be found in [6, section 6].

lite: We suggest that you always use this option in order to avoid a conflict between `amsrefs` and the Xy-pic package (see section 8). If you don't use the `lite` option, then `amsrefs` automatically loads the packages `mathscinet` (which defines a number of special characters and accents that are sometimes encountered when downloading data from MathSciNet) and `txtcmds` (which provides shorthand commands for a number of characters that are usually specified via ligatures). The problem is that both Xy-pic and the `txtcmds` package define the command `\cir` (they define it to be entirely different things). If you load `txtcmds` *before* loading Xy-pic then Xy-pic will redefine `\cir` as it pleases while putting a warning into your `.log` file, but if you load `txtcmds` *after* loading Xy-pic then LaTeX will stop with an error, since the `txtcmds` package uses `\newcommand` to define `\cir`. Unfortunately, if you need to use the `mathscinet` package and you try to load it by putting the command `\usepackage{mathscinet}` after your `\documentclass` command, it will also load the `txtcmds` package, so this doesn't avoid the conflict.

If you need to use both Xy-pic and the definitions in the `mathscinet` and `txtcmds` packages, you should load `amsrefs` *before* Xy-pic, as in

```
\usepackage{amsrefs}
\usepackage[all,cmtip]{xy}
```

to keep LaTeX from generating an error when loading `txtcmds`. (Of course, the command `\cir` as defined in `textcmds` will not be available.)

initials: If you use this option, then all authors, editors, and translators will have their first and middle names replaced by their initials.

alphabetic: If you use this option, as in

```
\usepackage[alphabetic,lite]{amsrefs}
```

then instead of numbers being used to label the bibliography entries, you will get alphabetic labels similar to the `alpha` style used by BibTeX, consisting of the first letter(s) of each author name plus the year of publication.

shortalphabetic: This is similar to the `alphabetic` option, except that you'll get a shorter alphabetic label using only the first letter of each author name.

y2k: If you use the `alphabetic` option, only the last two digits of the year are normally used in the label. If you also use the `y2k` option, as in

```
\usepackage[y2k,alphabetic,lite]{amsrefs}
```

then the full year will be used.

author-year: If you use this option, then bibliography items will not be labelled at all, and references to them will be in the author-year format similar to that described in `The Chicago Manual of Style`. If you do use this option, you may sometimes want to use the `\ycite` and `\ocite` commands when referring to bibliography items (see section 5.3.2).

12.7. Avoiding both `amsrefs` and BibTEX.

If for some reason you don't want to use `amsrefs` or BibTEX, you can type each bibliography item exactly as you want it to appear and have the items numbered automatically. To do this, you

(1) begin the bibliography with the command

```
\begin{thebibliography}{number}
```

where *number* is any number that, when printed, is as wide as the widest number of any item in the bibliography,

(2) define the bibliography items using a `\bibitem` command for each item (see section 12.7.1), and then

(3) end the bibliography with the command

```
\end{thebibliography}
```

The only use made of the `number` in `\begin{thebibliography}{number}` is that LATEX assumes that its width when printed is at least as large as the width of any number of an item in the bibliography. For example, if the bibliography will contain between 10 and 99 items, you can use `\begin{thebibliography}{99}`.

12.7.1. *Bibliography items.* When using the commands `\begin{thebibliography}` and `\end{thebibliography}` to create the bibliography (see section 12.7), each item is begun with a `\bibitem` command. The format is

```
\bibitem{ReferenceKey}Item entry
```

For example, had we not been using `amsrefs`, the bibliography in these instructions might have contained the entry

```
\bibitem{yellowmonster}
A. K. Bousfield and D. M. Kan, \emph{Homotopy Limits, Completions
and Localizations,} Lecture Notes in Mathematics number 304,
Springer-Verlag, New York, 1972.
```

The above entry would allow us to type

```
Homotopy inverse limits are discussed
in~\cite[Chapter 11]{yellowmonster}.
```

and have it print as "Homotopy inverse limits are discussed in [4, Chapter 11]." For more on the `\cite` command, see section 5.3.

13. The template file

The following is the text of the file `template.tex`.

```
%%% template.tex
%%% This is a template for making up an AMS-LaTeX file
%%% Version of February 12, 2011
%%%---------------------------------------------------------
%%% The following command chooses the default 10 point type.
%%% To choose 12 point, change it to
%%% \documentclass[12pt]{amsart}
\documentclass{amsart}

%%% The following command loads the amsrefs package, which will be
%%% used to create the bibliography:
\usepackage[lite]{amsrefs}

%%% The following command defines the standard names for all of the
%%% special symbols in the AMSfonts package, listed in
%%% http://www.ctan.org/tex-archive/info/symbols/math/symbols.pdf
\usepackage{amssymb}

%%% The following commands allow you to use \Xy-pic to draw
%%% commutative diagrams.  (You can omit the second line if you want
%%% the default style of the nodes to be \textstyle.)
\usepackage[all,cmtip]{xy}
\let\objectstyle=\displaystyle

%%% If you'll be importing any graphics, uncomment the following
%%% line.  (Note: The spelling is correct; the package graphicx.sty is
%%% the updated version of the older graphics.sty.)
% \usepackage{graphicx}



%%% This part of the file (after the \documentclass command,
%%% but before the \begin{document}) is called the ``preamble''.
%%% This is where we put our macro definitions.

%%% Comment out (or delete) any of these that you don't want to use.
\newcommand{\tensor}{\otimes}
\newcommand{\homotopic}{\simeq}
\newcommand{\homeq}{\cong}
\newcommand{\iso}{\approx}

\DeclareMathOperator{\ho}{Ho}
\DeclareMathOperator*{\colim}{colim}

\newcommand{\R}{\mathbb{R}}
\newcommand{\C}{\mathbb{C}}
\newcommand{\Z}{\mathbb{Z}}

\newcommand{\M}{\mathcal{M}}
\newcommand{\W}{\mathcal{W}}

\newcommand{\itilde}{\tilde{\imath}}
\newcommand{\jtilde}{\tilde{\jmath}}
\newcommand{\ihat}{\hat{\imath}}
\newcommand{\jhat}{\hat{\jmath}}
```

```
%%%-----------------------------------------------------------------
%%%-----------------------------------------------------------------
%%% The Theorem environments:
%%%
%%%
%%% The following commands set it up so that:
%%%
%%% All Theorems, Corollaries, Lemmas, Propositions, Definitions,
%%% Remarks, Examples, Notations, and Terminologies  will be numbered
%%% in a single sequence, and the numbering will be within each
%%% section.  Displayed equations will be numbered in the same
%%% sequence.
%%%
%%%
%%% Theorems, Propositions, Lemmas, and Corollaries will have the most
%%% formal typesetting.
%%%
%%% Definitions will have the next level of formality.
%%%
%%% Remarks, Examples, Notations, and Terminologies will be the least
%%% formal.
%%%
%%% Theorem:
%%% \begin{thm}
%%%
%%% \end{thm}
%%%
%%% Corollary:
%%% \begin{cor}
%%%
%%% \end{cor}
%%%
%%% Lemma:
%%% \begin{lem}
%%%
%%% \end{lem}
%%%
%%% Proposition:
%%% \begin{prop}
%%%
%%% \end{prop}
%%%
%%% Definition:
%%% \begin{defn}
%%%
%%% \end{defn}
%%%
%%% Remark:
```

```
%%% \begin{rem}
%%%
%%% \end{rem}
%%%
%%% Example:
%%% \begin{ex}
%%%
%%% \end{ex}
%%%
%%% Notation:
%%% \begin{notation}
%%%
%%% \end{notation}
%%%
%%% Terminology:
%%% \begin{terminology}
%%%
%%% \end{terminology}
%%%
%%%       Theorem environments

% The following causes equations to be numbered within sections
\numberwithin{equation}{section}

% We'll use the equation counter for all our theorem environments, so
% that everything will be numbered in the same sequence.

%       Theorem environments

\theoremstyle{plain} %% This is the default, anyway
\newtheorem{thm}[equation]{Theorem}
\newtheorem{cor}[equation]{Corollary}
\newtheorem{lem}[equation]{Lemma}
\newtheorem{prop}[equation]{Proposition}

\theoremstyle{definition}
\newtheorem{defn}[equation]{Definition}

\theoremstyle{remark}
\newtheorem{rem}[equation]{Remark}
\newtheorem{ex}[equation]{Example}
\newtheorem{notation}[equation]{Notation}
\newtheorem{terminology}[equation]{Terminology}

%%%-------------------------------------------------------------------
%%%-------------------------------------------------------------------
%%%-------------------------------------------------------------------
%%%-------------------------------------------------------------------
```

```
%%%-------------------------------------------------------------------
%%%-------------------------------------------------------------------
%%%-------------------------------------------------------------------
\begin{document}

%%% In the title, use a double backslash "\\" to show a linebreak:
%%% Use one of the following two forms:
%%% \title{Text of the title}
%%% or
%%% \title[Short form for the running head]{Text of the title}
\title{}


%%% If there are multiple authors, they're described one at a time:
%%% First author: \author{} \address{} \curraddr{} \email{} \thanks{}
%%% Second author: \author{} \address{} \curraddr{} \email{} \thanks{}
%%% Third author: \author{} \address{} \curraddr{} \email{} \thanks{}
\author{}

%%% In the address, show linebreaks with double backslashes:
\address{}

%%% Current address is optional.
% \curraddr{}

%%% Email address is optional.
% \email{}


%%% If there's a second author:
% \author{}
% \address{}
% \curraddr{}
% \email{}


%%% To have the current date inserted, use \date{\today}:
\date{}

%%% To include an abstract, uncomment the following two lines and type
%%% the abstract in between them:
% \begin{abstract}
% \end{abstract}


\maketitle

%%% To include a table of contents, uncomment the following line:
```

```
% \tableofcontents


%%%-------------------------------------------------------------------
%%%-------------------------------------------------------------------
%%% Start the body of the paper here!  E.G., maybe use:
%%% \section{Introduction}
%%% \label{sec:intro}

%%% For a numbered display, use
%%% \begin{equation}
%%%   \label{something}
%%%   The display goes here
%%% \end{equation}
%%% and you can refer to it as \eqref{something}.

%%% For an unnumbered display, use
%%% \begin{equation*}
%%%   The display goes here
%%% \end{equation*}

%%% To import a graphics file, you must have said
%%% \usepackage{graphicx}
%%% in the preamble (i.e., before the \begin{document}).
%%% Putting it into a figure environment enables it to float to the
%%% next page if there isn't enough room for it on the current page.
%%% The \label command must come after the \caption command.
% \begin{figure}[h]
%   \includegraphics{filename}
%   \caption{Some caption}
%   \label{somelabel}
% \end{figure}



%%% -------------------------------------------------------------------
%%% -------------------------------------------------------------------
%%% This is where we create the bibliography.

\begin{bibdiv}
  \begin{biblist}

%%% The format of bibliography items is as in the following examples:
%%%
%%% \bib{yellowmonster}{book}{
%%%   author={Bousfield, A.K.},
%%%   author={Kan, D.M.},
%%%   title={Homotopy Limits, Completions and Localizations},
%%%   date={1972},
%%%   series={Lecture Notes in Mathematics},
%%%   volume={304},
%%%   publisher={Springer-Verlag},
%%%   address={Berlin-New York}
%%% }

%%% \bib{HA}{book}{
%%%   author={Quillen, Daniel G.},
%%%   title={Homotopical Algebra},
%%%   series={Lecture Notes in Mathematics},
%%%   volume={43},
%%%   publisher={Springer-Verlag},
%%%   address={Berlin-New York},
%%%   date={1967}
%%% }

%%% \bib{serre:shfs}{article}{
%%%   author={Serre, Jean-Pierre},
%%%   title={Homologie Singuli\`ere des Espaces Fibr\'es.  Applications},
%%%   journal={Ann. of Math. (2)},
%%%   date={1951},
%%%   volume={54},
%%%   pages={425--505}
%%% }






  \end{biblist}
\end{bibdiv}

\end{document}
```

References

[1] American Mathematical Society, *User's Guide for the* `amsmath` *Package*, version 2.0, February 25, 2002, ftp://ftp.ams.org/pub/tex/doc/amsmath/amsldoc.pdf. also available from CTAN (the Comprehensive TEX Archive Network) at http://www.ctan.org/tex-archive/macros/latex/required/amslatex/math/amsldoc.pdf.

[2] ———, *Instructions for Preparation of Papers and Monographs: $\mathcal{A}\mathcal{M}\mathcal{S}$-LATEX*, August 2004, ftp://ftp.ams.org/pub/tex/doc/amscls/instr-l.pdf.

[3] ———, *Sample Paper for the* `amsmath` *Package*, November 1996.

[4] A.K. Bousfield and D.M. Kan, *Homotopy Limits, Completions and Localizations*, Lecture Notes in Mathematics, vol. 304, Springer-Verlag, Berlin-New York, 1972.

[5] Michael Downes, *Short Math Guide for LATEX*, version 1.09, ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf.

[6] David M. Jones, *User's Guide to the* `amsrefs` *Package*, October 16, 2007, ftp://ftp.ams.org/pub/tex/amsrefs/amsrdoc.pdf.

[7] Leslie Lamport, *LATEX: A Document Preparation System*, 2nd ed., Addison-Wesley, 1994.

[8] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl, *The Not So Short Introduction to LATEX 2ε*, version 4.24, http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf.

[9] Daniel G. Quillen, *Homotopical Algebra*, Lecture Notes in Mathematics, vol. 43, Springer-Verlag, Berlin-New York, 1967.

[10] ———, *Rational Homotopy Theory*, Ann. of Math. (2) **90** (1969), 205–295.

[11] Kristoffer H. Rose, *XY-pic User's Guide*, version 3.85, http://www.ctan.org/tex-archive/macros/generic/diagrams/xypic/xy-3.8.5/doc/xyguide.pdf.

[12] Kristoffer H. Rose and Ross Moore, *XY-pic Reference Manual*, version 3.85, http://www.ctan.org/tex-archive/macros/generic/diagrams/xypic/xy-3.8.5/doc/xyrefer.pdf.

Department of Mathematics, Wellesley College, Wellesley, Massachusetts 02481
E-mail address: psh@math.mit.edu