

Introduction to L^AT_EX

There are lots of good online introductions to L^AT_EX, for example <http://en.wikibooks.org/wiki/LaTeX/>, <http://ctan.tug.org/tex-archive/info/lshort/english/lshort.pdf>, and <ftp://ftp.ams.org/ams/doc/amsmath/short-math-guide.pdf>. In these notes, which freely borrow from all these introductions, I have tried to provide value-added by skipping stuff you don't see much in economics articles, and adding stuff you do.

1. INSTALLING L^AT_EX

If you have a Mac, go to <http://pages.uoregon.edu/koch/texshop/obtaining.html> and click on the `MacTeX.mpkg.zip` link halfway down the page. A huge file will download. Click on it and it will self-install. Done! You'll find a new folder called TeX inside your Applications folder, and inside that a file called `READ ME FIRST.pdf`. Do that (read it first) and you'll be up and running.

If you have a PC, I'm afraid you're on your own, because nothing seems to work reliably for every possible Windows flavor. Try <http://www.tug.org/texworks/>, or <http://www.xmlmath.net/texmaker/>, or <http://miktex.org/>. Ask around. Google.

One thing you should be aware of is that L^AT_EX comes in two parts. One is usually referred to as the L^AT_EX “distribution.” It consists of a huge number of folders and packages and fonts and style files that any of the above packages will install for you on your computer, and that normally will stay hidden from you. The other part is usually referred to as the L^AT_EX “front-end.” It is the application that you actually interact with when you use L^AT_EX, by typing in text and L^AT_EX commands, and then compiling the result into a `.pdf` file.

On the Mac, the above instructions will install both parts, and will give you a choice between using either TeXShop and TeXworks as your front-end. Try out both, and see which you like best.¹

2. THE SIMPLEST POSSIBLE L^AT_EX FILE

The three lines of a L^AT_EX file that are absolutely essential are

```
\documentclass{amsart}

\begin{document}

\end{document}
```

The `documentclass` line specifies the overall “look” of the document. Economics journals usually require a look that is close to that used by journals of the American Mathematical Society.²

Anything extra between the `\documentclass{amsart}` line and the `\begin{document}` line is called the “preamble.” This is where you can specify all kinds of options and special commands to make your document look differently, or to make typing up your document easier. I'll get into some of these options and commands in Section 3 below.

¹I don't use either: my preferred front-end is an editor called vim, with a plug-in called auctex. While I highly recommend vim—it's great for editing MATLAB, Stata, and other programs' files as well—there's a learning curve involved in switching to it, so I'd stick with simpler front-ends for now.

²The string “amsart” is short for “AMS article.”

The actual text of your document comes between the `\begin{document}` and `\end{document}` lines. So if you just add something between those lines, as shown below on the left, and then “compile” the file by hitting the compile button (or pressing the compile keyboard shortcut) on your front-end, it should produce a .pdf file (it’ll probably ask you to give it a name and save it somewhere) that looks like the box on the right (except of course that it’ll be the size of a full page):

```
\documentclass{amsart}
```

```
\begin{document}
```

```
This is my very first LaTeX document.
```

```
\end{document}
```

This is my very first LaTeX document.

That’s it! Your very first L^AT_EX document.

In fact, this is how most front-ends have you work with L^AT_EX: you type stuff in a window on the left side of your screen, hit a compile button, and then see the result on the right side of your screen. The contents of the left window are saved as a .tex file, those on the right as a .pdf file.³

This separation between what you type and the final output is a key difference between L^AT_EX and most other text editors. While it may seem a hassle at first, you’ll soon understand why it’s actually one of the big reasons for L^AT_EX’s popularity.

The other, related key difference is that formatting in L^AT_EX works not with menus and buttons and checkboxes, but with so-called “commands,” which are distinguished by the fact that they start with the backslash “\” character. Any word with a backslash in front of is interpreted by L^AT_EX not as regular text, but as an instruction telling L^AT_EX to do something: format text in a particular way, say, or apply a certain shortcut. In the example below, the text string LaTeX without a backslash is just treated as regular text, but the text string \LaTeX is treated as a command to print out the specially formatted string “L^AT_EX” to the .pdf:

```
LaTeX versus \LaTeX
```

LaTeX versus L^AT_EX

3. PREAMBLE

The preamble is used to make specify global changes to the default look of a document specified by the documentclass.

Incidentally, many economics journals now provide preambles that L^AT_EX users can simply download and paste at the top of their document in order to get their paper formatted exactly the way the journal wants it. For Word users, getting the formatting right can be a major pain.

Similarly, graduate-student L^AT_EX users can usually get hold of a preamble that formats their thesis exactly the way universities wants it. There’s one for UW theses too.

³A whole bunch of other files are created and saved automatically as well, with extensions like .aux, .bbl, and .log, but you rarely if ever need to worry about those. In fact, you can delete them at any point in time and L^AT_EX will just re-create them if and when necessary.

3.1. Change default font size

For the `amsart` documentclass, the default font size for the document as a whole is 10 points. The simplest way to change this is to specify a different font size in brackets just before the `documentclass` declaration, like this:

```
\documentclass[11pt]{amsart}
```

3.2. Change default equation numbering

While you're at it, you may as well also add the “`reqno`” option, which specifies that equation numbers should be placed on the right side of the page, rather than on `amsart`'s default left side.

```
\documentclass[11pt,reqno]{amsart}
```

3.3. Add packages

\LaTeX users have written hundreds of so-called “packages” that modify the default behavior \LaTeX , or add features that are not included in the standard \LaTeX setup. Many of these packages are installed on your computer when you first install \LaTeX , but are ignored by \LaTeX unless you specifically include them in the preamble. You do this by using the `\usepackage` command in the preamble.

One package that you'll almost always need in economics papers is the `graphicx` package, which allows you to import graphs or figures into your document (see Section 8 for details). To include it, simply add the line

```
\usepackage{graphicx}
```

to your preamble.

You'll normally also want to include the `amssymb` and `amsthm` packages, which add non-standard math symbols and useful shortcuts for writing Theorems, Propositions, and Lemma's and the like.

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

Another package you'll almost always want to include is the `natbib` package, which handles citations (see Section 10 for details).

```
\usepackage{natbib}
\bibpunct{(}{)}{;}{:}{,}{}
```

The `\bibpunct` command changes some of `natbib`'s default punctuation of citations, to make them conform to standard practice in economics journals.

Optionally, you can also include the `hyperref` package. It turns all the various references used in your paper—citations, URLs, as well as cross-references to the paper's tables, figures, sections, footnotes, equations, and pages—into “clickable” hyperlinks (see Section 11 for details).

```
\usepackage[colorlinks=true,citecolor=blue,urlcolor=blue,linkcolor=blue]{hyperref}
```

Often you can specify various options to a package's behavior in brackets before the package declaration. Above, I have specified that I want `hyperref` to color hyperlinks rather than indicating them by a box (the default), and that I want it to color the main link types—citations, URLs, and internal references—blue rather than the default variety of colors.

Detailed documentation for any L^AT_EX package can usually be found online. Just Google “latex” and then the name of the package.

3.4. Change page layout

By default, the `amsart` documentclass leaves very wide margins around the text. If you prefer narrower margins, you can specify that as an option in the L^AT_EX file’s preamble. For example, if you add

```
\setlength{\hoffset}{-0.75in}
\setlength{\voffset}{-0.75in}
\setlength{\textwidth}{6.5in}
\setlength{\textheight}{9in}
```

to your preamble, you get 1-inch margins on all four sides.

3.5. Change paragraph formatting

By default, the `amsart` documentclass leaves no space between paragraphs in the .pdf output, and indents each paragraph’s first line. If you prefer a different look, add for example

```
\setlength{\parskip}{0.5\baselineskip}
\setlength{\parindent}{0in}
```

to your preamble. This tells L^AT_EX to leave open (“skip”) half a line between paragraphs, and to indent the first line by zero inch, i.e., not at all.

3.6. Change default section numbering

By default, the `amsart` documentclass numbers all sections, subsections, sub-subsections and down. If you prefer to number only sections, but not subsections and down, then add this to your preamble:

```
\setcounter{secnumdepth}{1}
```

If you prefer to number sections and subsections, but not sub-subsections and down, change the `secnumdepth` to 2, etc. Setting the `secnumdepth` to 0 specifies that nothing should be numbered.

3.7. Change default page numbering

By default, the `amsart` documentclass places the page number for the very first page of a document at the bottom-center of the page, but page numbers for additional pages alternately at the top-left and top-right. If you prefer to have all page numbers at the bottom-center, add

```
\pagestyle{plain}
```

to your preamble. If you prefer no numbers at all, add

```
\pagestyle{empty}
```

4. LOCAL FORMATTING

Where the previous section focused on on global, document-wide formatting issues, this section focuses on more local formatting of a document’s text elements: its paragraphs, lines, words, and the spaces in between.

4.1. Paragraphs

L^AT_EX treats any space in the .tex file that is shorter than a full line (or equivalently, does not contain a carriage return) as if it is a single space. As a result, the .pdf output you get if you type, for example,

```
Once
upon
a
time
```

Once upon a time

is exactly the same as if you type

```
Once upon a time
```

Once upon a time

As you’ll see below, this feature can be very useful when you type out complicated math expressions.

On the other hand, any space in the .tex file that is a full line or longer is interpreted as a paragraph break. To start a new paragraph, you therefore just leave one or more blank lines in your typed text.

As mentioned above, the .pdf output will by default have no space between the paragraphs and will indent each paragraph’s first line. So

```
\noindent
```

```
The first paragraph of an academic article should tell the reader something
about the article’s topic that she already knows, or could reasonably
be expected to know.
```

```
The second paragraph introduces a ‘‘twist’’---something about the subject
that the reader does {\em not} know. This paragraph should set up a
tension in the reader, raise a question in her mind.
```

```
The third paragraph immediately relieves the tension, by telling the
reader how this article is going to address that question.
```

shows up as

The first paragraph of an academic article should tell the reader something about the article’s topic that she already knows, or could reasonably be expected to know.

The second paragraph introduces a “twist”—something about the subject that the reader does *not* know. This paragraph should set up a tension in the reader, raise a question in her mind.

The third paragraph immediately relieves the tension, by telling the reader how this article is going to address that question.

Notice that you can locally suppress the indentation of a paragraph's first line by preceding it by a `\noindent` command.

4.2. Justification

By default, \LaTeX fully justifies text, so both the left and right margins are flush. To instead center text (as you might want to do for the title of your paper), put it between `\begin{center}` and `\end{center}` commands:

```
\begin{center}
  My Masterwork
\end{center}
```

My Masterwork

The `\begin{}` and `\end{}` commands begin and end what is known as a \LaTeX “environment,” in this case the `center` environment. Think of an environment as a special region in the `.tex` file that is to be formatted or interpreted in a particular way.

To right-justify text, use the `flushright` environment:

```
\begin{flushright}
  \today
\end{flushright}
```

January 12, 2012

Notice that the command `\today` automatically enters the date at which you compile the `.pdf` file.

4.3. Section headers

Section, subsection, sub-subsection headers, etc., are specified with `\section`, `\subsection`, `\subsubsection`, etc. commands:

```
\section{Roadmaps}
```

This section covers different ways of providing the reader with “roadmaps” of where you are going to take him/her next in the paper.

```
\subsection{Introductory paragraphs}
```

Start each section with a paragraph outlining what the section is going to cover, like I did just now.

```
\subsection{“Umbrella” sentences}
```

Start or end each paragraph with a sentence summarizing what the paragraph is about.

4.4. Footnotes

To generate footnotes such as the ones on page 1, use the `\footnote` command:

```
... American Mathematical Society.\footnote{The string ‘amsart’ is short for ‘AMS article.’} This look is ...
```

4.5. Font size

To increase the font size for just some text—your title, for example—use the commands `\large`, `\Large`, `\LARGE`, `\huge`, and `\Huge`:

3. ROADMAPS

This section covers different ways of providing the reader with “roadmaps” of where you are going to take him/her next in the paper.

3.1. Introductory paragraphs. Start each section with a paragraph outlining what the section is going to cover, like I did just now.

3.2. “Umbrella” sentences. Start or end each paragraph with a sentence summarizing what the paragraph is about.

Ever `{\large larger}`,
and `{\Large larger}`,
and `{\LARGE larger}`,
and `{\huge larger}`,
and `{\Huge larger}`.

Ever larger,
and larger,
and larger,
and larger,
and larger.

To decrease the font size, use `\small`, `\footnotesize`, `\scriptsize`, and `\tiny`:

Ever `{\small smaller}`,
and `{\footnotesize smaller}`,
and `{\scriptsize smaller}`,
and `{\tiny smaller}`.

Ever smaller,
and smaller,
and smaller,
and smaller.

4.6. Bolding and italicizing

To bold text, surround it with braces, and then put the command `\bf` (short for “bold font”) at the beginning, like this:

To `{\bf boldly}` go where no
man has gone before.

To **boldly** go where no man has gone before.

To italicize text, do the same, but use the command `\it` or `\em` (short for “emphasized”):

Do `{\it not}` overuse `{\em italics}`!

Do *not* overuse *italics*!

4.7. Line and page breaks

To force a line break use two backslashes:

The old pond;\\
A frog jumps in ---\
The sound of the water.

{\em Matsuo Basho}

The old pond;
A frog jumps in —
The sound of the water.

Matsuo Basho

Notice also that three consecutive dashes, “---” produce an em-dash.

Use the command `\newpage` to force a page break:

4.8. Modifying spacing

To locally reduce or increase the horizontal or vertical space between text elements like words, lines, paragraphs, or (as discussed later on) tables and figures, use the commands `\hspace*{}` and `\vspace*{}`, with the size of the space specified between the braces. You can use a variety of units for that: `pt` (points, equal to 1/72 of an inch, which is the standard unit to measure font size) `cm` (centimeter), `in` (inch), and also `ex` (the height of a letter “x” in the current font) and `em` (the width of the letter “M” in the current font). The advantage of using the latter units is that the space will automatically scale up or down with any changes you might later make to the font or fontsize of your entire document.

up

```
\vspace*{0.2in}\hspace*{3em}and
```

```
\vspace*{-0.5in}\hspace*{1.3in}down
```

```
left\hspace*{2in}and\hspace*{-2ex}right
```

up		down
left	and	right

4.9. Lists

To generate a bulleted list, use the `itemize` environment, and precede each item of the list by the `\item` command:

```
\begin{itemize}
  \item Exogenous variables
  \item Endogenous decision variables
  \item Endogenous “other” variables
\end{itemize}
```

<ul style="list-style-type: none">• Exogenous variables• Endogenous decision variables• Endogenous “other” variables
--

You can also nest lists:

```
\begin{itemize}
  \item Exogenous variables
  \item Endogenous variables
    \begin{itemize}
      \item decision
      \item “other”
    \end{itemize}
\end{itemize}
```

<ul style="list-style-type: none">• Exogenous variables• Endogenous variables<ul style="list-style-type: none">– decision– “other”
--

Anything you write in brackets immediately following `\item` replaces the default label:

```
\begin{itemize}
  \item Exogenous variables
  \item Endogenous variables
    \begin{itemize}
      \item[ $\circ$ ] decision
      \item[ $\circ$ ] “other”
    \end{itemize}
\end{itemize}
```

<ul style="list-style-type: none">• Exogenous variables• Endogenous variables<ul style="list-style-type: none">◦ decision◦ “other”
--

But if you're unhappy with L^AT_EX's defaults for these labels, you can also redefine them in the preamble. The L^AT_EX commands `\labelitemi`, `\labelitemii`, and `\labelitemiii` define the formats of respectively the top-, second-, and third-level itemize labels, so adding for example

```
\renewcommand{\labelitemii}{\circ}
```

to your preamble redefines the second-level label to “o” throughout your document.

To generate a numbered list, use the `enumerate` environment:

```
\begin{enumerate}
  \item Inequality constraints
    \begin{enumerate}
      \item binding
      \item non-binding \label{itm:nb}
    \end{enumerate}
  \item Equality constraints
\end{enumerate}
```

<pre>(1) Inequality constraints (a) binding (b) non-binding (2) Equality constraints</pre>
--

(Ignore the `\label` command for now.) Again, if you're unhappy with L^AT_EX's default label formats, you can either change them locally by using brackets

```
\begin{enumerate}
  \item[1.] Inequality constraints
    \begin{enumerate}
      \item[(i)] binding
      \item[(ii)] non-binding
    \end{enumerate}
  \item[2.] Equality constraints
\end{enumerate}
```

<pre>1. Inequality constraints (i) binding (ii) non-binding 2. Equality constraints</pre>

or change them globally by redefining the the L^AT_EX commands `\labelenumi`, `\labelenumii`, and `\labelenumiii` in the preamble. In the above case, you would specify

```
\renewcommand{\labelenumi}{\arabic{enumi}.}
\renewcommand{\labelenumii}{\roman{enumii}}
```

4.10. International accents and characters

Below is a list of all the international accents and characters you're likely to ever need. In general you generate accents by putting a backslash followed by something that looks like the accent before the letter you want to apply the accent to. With the first five accents, you could, but don't need to, put braces around that letter. Notice also that `\i` generates the letter “i” without its dot.

<code>\'a</code>	\acute{a}	<code>\'{\i}</code>	\acute{i}	<code>\={a}</code>	\bar{a}	<code>\c{c}</code>	\c{c}	<code>\aa</code>	\AA
<code>\'a</code>	\grave{a}	<code>\'{\i}</code>	\grave{i}	<code>\.a</code>	\grave{a}	<code>\ae</code>	\ae	<code>\AA</code>	\AA
<code>\^a</code>	\hat{a}	<code>\^{\i}</code>	\hat{i}	<code>\v{a}</code>	\check{a}	<code>\oe</code>	\oe	<code>\o</code>	\o
<code>\~a</code>	\tilde{a}	<code>\~{\i}</code>	\tilde{i}	<code>\u{a}</code>	$\text{\u{a}}$	<code>\AE</code>	\AE	<code>\O</code>	\O
<code>\"a</code>	\aa	<code>\"{\i}</code>	\i	<code>\H{a}</code>	$\text{\H{a}}$	<code>\OE</code>	\OE	<code>\ss</code>	\ss

4.11. Special characters and verbatim printing

The ten characters `#`, `$`, `%`, `&`, `{`, `}`, `_`, `^`, `~`, and `\` are called “special characters,” because L^AT_EX by default interprets them as being part of a command. Paradoxically, when you want to print out

these characters as is, you generally need to put a backslash in front of them to make \LaTeX treat them as regular characters. This is called “escaping” the characters.

For the last three characters, though, that doesn’t work either, because their escaped versions have special meanings in \LaTeX as well: as shown above, \^ and \~ are used to generate hat and tilde accents, respectively, and \ is used to generate a line break.

To print out these characters as is, you’ll need to use the \verb (short for “verbatim”) command, although for a literal backslash you can also use $\text{\$}\text{\backslash}\text{\$}$. The \verb command tells \LaTeX to look at the single character that immediately follows the command and then to print whatever comes next, up to the closest second instance of that same character “verbatim,” i.e., literally. In the second column below, I could therefore also have used, for example, \verb\#\# or \verb\&\& or \verb\@\@ to print a literal hat, but $\text{\verb\{}$ would cause \LaTeX to complain.

\#	\#	\verb\^ 	\^
$\text{\$}$	$\text{\$}$	\verb\~ 	\~
\%	\%	\verb\\ 	\
\&	\&	$\text{\$}\text{\backslash}\text{\$}$	\
$\text{\{}$	$\text{\{}$		
$\text{\}}$	$\text{\}}$		
_	_		

\LaTeX also has a `verbatim` environment that can be used to print out larger chunks of text verbatim, without having \LaTeX interpret that text in any way. In fact, I’m using that environment throughout these notes to show you \LaTeX code as is:

```
\begin{verbatim}
  but \verb|\verb{^}| would cause
  \LaTeX\ to complain.
\end{verbatim}
```

but $\text{\verb|\verb{^}|}$ would cause \LaTeX to complain.

A final special character is the “literal space.” As I explained above, \LaTeX does its own thing with the spaces that you type into your .tex file, sometimes treating multiple spaces as if you had typed a single space, sometimes treating them as a paragraph break, and sometimes ignoring them altogether. If you want \LaTeX to treat a space as literally a space and not mess with it, just escape the space with a backslash in front, so type “ \ ”.

This is useful, for example, whenever you use an abbreviation followed by a period in the middle of a sentence. \LaTeX will by default interpret that period as ending the sentence, and print two spaces after it in the .pdf file, regardless of how many spaces you type after it in the .tex file. To avoid this, escape the space:

```
Differentiating w.r.t.  $\text{\$L}$  gives ...
Differentiating w.r.t.\  $\text{\$L}$  gives ...
The f.o.c. is ...
The f.o.c.\ is ...
```

Differentiating w.r.t. L gives ...
 Differentiating w.r.t. L gives ...
 The f.o.c. is ...
 The f.o.c. is ...

5. COMMENTING

So far, I've only shown you how to do things in L^AT_EX that you can do at least as easily by pointing and clicking your way through various menus in, say, Word. Why then bother with L^AT_EX? The remainder of this introduction will show you why, by showing you various things that you can't do as easily in Word, or can't do at all.

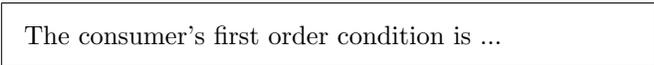
One of these is adding comments to your paper. Yes, you can add comments in Word too, but only in limited ways. For example, you can't insert math in Word comments, nor can you attach comments to footnotes. Also, if you have a lot of comments (as I usually do on student papers), the margin quickly becomes impossibly cluttered.

In L^AT_EX, you can add comments by using the fact that L^AT_EX ignores anything in a line that comes after a percent “%” sign. So if your co-author writes

```
The consumer's first order condition is ...
```

you can simply add a comment to the .tex file by adding, say

```
The consumer's first order condition
% Shouldn't first-order be hyphenated?
is ...
```

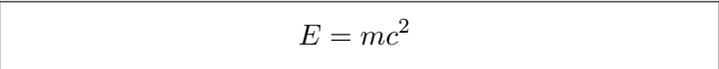


without having your comment show up in the .pdf. Most front-ends also helpfully color any such comments in your .tex file, to make them stand out.⁴ So the above actually shows up as

```
The consumer's first order condition
% Shouldn't first-order be hyphenated?
is ...
```

Because of the way L^AT_EX treats spaces in the .tex file, you can also add comments mid-line for example to remind yourself of what parts of an equation represent:

```
\begin{equation*}
E % Energy
=
m % mass
c^2 % speed of light squared
\end{equation*}
```


$$E = mc^2$$

If you want to add longer comments, add the following lines to your preamble:

```
\let\IG\iffalse
\let\ENDIG\fi
```

These lines define two new L^AT_EX commands, \IG (short for “ignore”) and \ENDIG (short for “end ignore”) that, if you put them respectively before and after any chunk of your text, cause L^AT_EX to skip over that chunk when it compiles the .pdf, just as it does with lines (or parts of lines) that begin with %.

You can use this, for example, to include detailed derivations that are too long to make it into the final version of a paper submitted to a journal, but that you want to have in there in case a referee

⁴In TeXworks, choose Syntax Coloring from the Format menu. It will then also color all L^AT_EX commands blue and all environment delimiters green.

or reader asks for them. The derivation will be sitting there in your .tex file right before the derived result:

```

\begin{equation*}
  Y = K^{\alpha}L^{\beta}
\end{equation*}
implies
\IG
(divide through by  $L^{\beta}$  to get
\begin{equation*}
  \frac{Y}{L^{\beta}} = K^{\alpha}
\end{equation*}
and then raise both sides to
the power  $1/\alpha$ )
\ENDIG
\begin{equation*}
  K = \left(
    \frac{Y}{L^{\beta}}
  \right)^{\frac{1}{\alpha}}
\end{equation*}

```

implies

$$Y = K^{\alpha}L^{\beta}$$

$$K = \left(\frac{Y}{L^{\beta}}\right)^{\frac{1}{\alpha}}$$

and whenever you *do* want to see it in the .pdf, all you need to do is comment out the \IG and \ENDIG lines (in effect telling L^AT_EX to ignore your ignore commands!):

```

\begin{equation*}
  Y = K^{\alpha}L^{\beta}
\end{equation*}
implies
%\IG
(divide through by  $L^{\beta}$  to get
\begin{equation*}
  \frac{Y}{L^{\beta}} = K^{\alpha}
\end{equation*}
and then raise both sides to
the power  $1/\alpha$ )
%\ENDIG
\begin{equation*}
  K = \left(
    \frac{Y}{L^{\beta}}
  \right)^{\frac{1}{\alpha}}
\end{equation*}

```

implies (divide through by L^{β} to get

$$\frac{Y}{L^{\beta}} = K^{\alpha}$$

and then raise both sides to the power $1/\alpha$)

$$K = \left(\frac{Y}{L^{\beta}}\right)^{\frac{1}{\alpha}}$$

The \IG and \ENDIG commands are also very useful for debugging purposes. As you'll find out, L^AT_EX is amazingly powerful, but also occasionally very frustrating. One source of frustration is that it sometimes refuses to compile a document to a .pdf just because there's some tiny error in the way you've written a command or a mathematical expression. This can be as simple as typing the opening brace { of a command, but accidentally forgetting to type the closing brace }.

If you're lucky, L^AT_EX will show you a message explaining exactly what the problem was, on what line of your typed text. In other cases, though, its error message is not helpful at all, and it can then be hard to locate the bug. One way to track it down is to put \IG and \ENDIG around suspect parts of the .tex file. If the .pdf compiles successfully after you do this, you know the problem must

lie somewhere between the `\IG` and `\ENDIG` lines. By gradually shrinking the commented-out part, you'll eventually zoom in on the line that is causing L^AT_EX to complain.

6. MATH

The main reason why L^AT_EX is the standard text editor used by engineers, mathematicians, and statisticians is that it has amazing math typesetting capabilities. As with everything, there is a bit of a learning curve initially, but once you get the hang of it, you'll never want to go back to using any other editor. In this section I've tried to collect all the math commands that you'll commonly use in economics.

6.1. Inline vs. displayed math

An important distinction to be aware of is that between “inline” math and “displayed” math. Inline math is math embedded inside regular lines of text. It is represented in L^AT_EX by surrounding dollar signs:

The production function is `$q = f(L)$`,
where `q` denotes output and `L` labor.

The production function is $q = f(L)$, where q denotes output and L labor.

Displayed math is math that is typeset on a separate line by itself. It is represented in L^AT_EX by a variety of L^AT_EX environments, introduced in the next two subsections.

6.2. Single equations

The simplest environment, used for single equations without any equation numbers is the `equation*` environment:

The production function is
`\begin{equation*}`
`q = f(L),`
`\end{equation*}`
where `q` denotes output and `L` labor.

The production function is
$$q = f(L),$$

where q denotes output and L labor.

If you do want an equation number to be generated automatically, use the `equation` environment instead, i.e., simply drop the asterisk in the environment's name:

The production function is
`\begin{equation}`
`q = f(L),`
`\end{equation}`
where `q` denotes output and `L` labor.

The production function is
$$q = f(L), \tag{1}$$

where q denotes output and L labor.

Notice (I'm a stickler for this) that it is good practice to punctuate displayed math *as if* it were part of the sentence surrounding it, in this case by following the displayed equation by a comma. Think of it as a courtesy to the reader, helping him/her follow your derivations more easily.

Sometimes you'll have an equation that is too long to fit on a single line. In that case, use the `\multline*` or `multline` environment (depending again on whether you want the equation to be numbered) and insert two backslashes at the point where you want to break up the equation:

```

\begin{multline}
dQ =
(\alpha K^\rho + \beta L^\rho)
^{\frac{1}{\rho}-1}
\alpha K^{\rho-1} dK \\
+
(\alpha K^\rho + \beta L^\rho)
^{\frac{1}{\rho}-1}
\beta L^{\rho-1} dL.
\end{multline}

```

$$dQ = (\alpha K^\rho + \beta L^\rho)^{\frac{1}{\rho}-1} \alpha K^{\rho-1} dK + (\alpha K^\rho + \beta L^\rho)^{\frac{1}{\rho}-1} \beta L^{\rho-1} dL. \quad (2)$$

6.3. Multiple equations

For a collection of miscellaneous equations of varying lengths, use the `gather*` or `gather` environment, and use double backslashes to separate the equations:

```

\begin{gather*}
p - \alpha\theta - c(n) = 0 \\
p = \gamma(n)f(\theta) \\
\gamma'(n)f(\theta) - c'(n) = 0
\end{gather*}

```

$$\begin{aligned}
p - \alpha\theta - c(n) &= 0 \\
p &= \gamma(n)f(\theta) \\
\gamma'(n)f(\theta) - c'(n) &= 0
\end{aligned}$$

More commonly, you'll have a collection of related equations such as first-order conditions, in which case you'll typically want to align them at the beginning, at the end, or at the equals signs. For this, use the `align*` or `align` environment, and use an ampersand `&` to indicate the place in each equation where you want it to be aligned with that same place in the other equations:

```

The first-order conditions are
\begin{align}
\frac{\partial \pi}{\partial K}
&= \alpha K^{\alpha-1} L^\beta - r = 0, \label{eqn:focK} \\
\frac{\partial \pi}{\partial L}
&= \beta K^\alpha L^{\beta-1} - w = 0. \label{eqn:focL}
\end{align}

```

The first-order conditions are

$$\frac{\partial \pi}{\partial K} = \alpha K^{\alpha-1} L^\beta - r = 0, \quad (3)$$

$$\frac{\partial \pi}{\partial L} = \beta K^\alpha L^{\beta-1} - w = 0. \quad (4)$$

(For now again ignore the `\label` commands.) You'll also want to use this environment when you want to show how an expression can be rewritten in a series of steps to simplify it:

```

\begin{align}
\frac{\partial^2 \pi}{\partial K \partial L}
&= \alpha\beta K^{\alpha-1} L^{\beta-1} \\
&= \frac{\alpha\beta K^\alpha L^\beta}{KL} \\
&= \alpha\beta \frac{Q}{KL}.
\end{align}

```

$$\begin{aligned}
\frac{\partial^2 \pi}{\partial K \partial L} &= \alpha\beta K^{\alpha-1} L^{\beta-1} \\
&= \frac{\alpha\beta K^\alpha L^\beta}{KL} \\
&= \alpha\beta \frac{Q}{KL}. \quad (5)
\end{aligned}$$

In such cases, you'll usually want only the last step to be numbered. As the example shows, you can achieve this by adding the command `\notag` before the double backslashes ending the steps that you *don't* want to be numbered (or "tagged").

When rewrites involve not just the right-hand side of an expression, but rather the expression as a whole, you'll want to use the `alignat*` or `alignat` environment. You'll also need the `\Leftrightarrow` command to indicate equivalence of the successive steps:

```
\begin{alignat}{2}
&&
Q + (p - m)\frac{dQ}{dp}
&= 0\notag\
&\Leftrightarrow\quad&
\frac{Q}{p} + \frac{p - m}{p}\frac{dQ}{dp}
&= 0\notag\
&\Leftrightarrow\quad&
\frac{p - m}{p}
&= -\frac{dp}{dQ}\frac{Q}{p}
&= |\epsilon|.
\end{alignat}
```

$$\begin{aligned}
 & Q + (p - m)\frac{dQ}{dp} = 0 \\
 \Leftrightarrow & \frac{Q}{p} + \frac{p - m}{p}\frac{dQ}{dp} = 0 \\
 \Leftrightarrow & \frac{p - m}{p} = -\frac{dp}{dQ}\frac{Q}{p} = |\epsilon|. \quad (6)
 \end{aligned}$$

The `{2}` after the `\begin{alignat}` indicates that each equation is to be aligned at two different points, indicated by the first and last of three ampersands in each equation. The first alignment point is indicated by the first ampersand, and in this case aligns the \Leftrightarrow symbols. The second alignment point is indicated by the last ampersand, in this case aligns the $=$ signs. The middle ampersand in each equation indicates the point at which the equation is to be split into two chunks for alignment purposes: everything to the left of that ampersand is to be aligned at the first alignment point; everything to right at the second alignment point.

I realize this is confusing; the `alignat` environment is easily the most complex one in L^AT_EX. Fortunately, the only time that you're likely to use it is precisely for the above type of derivation, which always has the same structure in the `.tex` file:

```
&&
LHS1 &= RHS1 \notag\
&\Leftrightarrow\quad&
LHS2 &= RHS2 \notag\
...
&\Leftrightarrow\quad&
LHSn &= RHSn.
```

You can therefore simply copy and paste the above code into your document and replace only the left-hand and right-hand side chunks `LHSi` and `RHSi` by the appropriate expressions for your derivation.

Sometimes, you'll want to insert explanatory text between derivation steps, without messing up the alignment. Use the `\intertext` command for this, which works in all multi-equation environments:

```
\begin{align}
Q + (p - m)\dfrac{dQ}{dp}
&= 0.\notag\
\intertext{Equivalently,
dividing through by $p$,}
\dfrac{Q}{p} +
\dfrac{p - m}{p}\dfrac{dQ}{dp}
&= 0,\notag\
\intertext{or, rearranging and
using the definition of $\epsilon$,}
\dfrac{p - m}{p}
&= -\dfrac{dp}{dQ}\dfrac{Q}{p}
= |\epsilon|.
\end{align}
```

$Q + (p - m)\frac{dQ}{dp} = 0.$
<p style="text-align: center;">Equivalently, dividing through by p,</p>
$\frac{Q}{p} + \frac{p - m}{p}\frac{dQ}{dp} = 0,$
<p style="text-align: center;">or, rearranging and using the definition of ϵ,</p>
$\frac{p - m}{p} = -\frac{dp}{dQ}\frac{Q}{p} = \epsilon . \quad (7)$

6.4. Greek letters

For lowercase Greek letters, just use a backslash with the name of the letter in lowercase:

<code>\alpha</code>	α	<code>\eta</code>	η	<code>\nu</code>	ν	<code>\upsilon</code>	υ
<code>\beta</code>	β	<code>\theta</code>	θ	<code>\xi</code>	ξ	<code>\phi</code>	ϕ
<code>\gamma</code>	γ	<code>\vartheta</code>	ϑ	<code>\pi</code>	π	<code>\varphi</code>	φ
<code>\delta</code>	δ	<code>\iota</code>	ι	<code>\rho</code>	ρ	<code>\chi</code>	χ
<code>\epsilon</code>	ϵ	<code>\kappa</code>	κ	<code>\varrho</code>	ϱ	<code>\psi</code>	ψ
<code>\varepsilon</code>	ε	<code>\lambda</code>	λ	<code>\sigma</code>	σ	<code>\omega</code>	ω
<code>\zeta</code>	ζ	<code>\mu</code>	μ	<code>\tau</code>	τ		

Notice that some lowercase letters have fairly common variant typesettings, indicated by “`var`” before the name. In particular, ϵ is commonly used to denote elasticities, but ε to denote the error term in econometrics.

For uppercase Greek letters, use a backslash and the name with the first letter capitalized:⁵

<code>\Gamma</code>	Γ	<code>\Theta</code>	Θ	<code>\Xi</code>	Ξ	<code>\Upsilon</code>	Υ
<code>\Delta</code>	Δ	<code>\Lambda</code>	Λ	<code>\Pi</code>	Π	<code>\Phi</code>	Φ
				<code>\Sigma</code>	Σ	<code>\Psi</code>	Ψ
						<code>\Omega</code>	Ω

Three other “Greek-derived” symbols are⁶

<code>\partial</code>	∂	<code>\nabla</code>	∇	<code>\infty</code>	∞
-----------------------	------------	---------------------	----------	---------------------	----------

⁵Not all Greek letters have uppercase versions that are distinct from their Roman counterparts: capital α , for example, is just A, and capital β is just B.

⁶The ∇ symbol is usually pronounced as “del,” and is used to denote vector gradients. Some people think the ∞ symbol is derived from the Greek letter ω .

6.5. Binary operators

<code>a + b</code>	$a + b$	<code>a \cdot b</code>	$a \cdot b$
<code>a - b</code>	$a - b$	<code>a / b</code>	a / b
<code>a \pm b</code>	$a \pm b$	<code>a^b</code>	a^b
<code>a \times b</code>	$a \times b$		

6.6. Relations

<code>a = b</code>	$a = b$	<code>a \succ b</code>	$a > b$	<code>a \succeq b</code>	$a \geq b$
<code>a \neq b</code>	$a \neq b$	<code>a \nsucc b</code>	$a \not> b$	<code>a \nsucceq b</code>	$a \not\geq b$
<code>a \equiv b</code>	$a \equiv b$	<code>a < b</code>	$a < b$	<code>a \gg b</code>	$a \gg b$
<code>a \approx b</code>	$a \approx b$	<code>a \nless b</code>	$a \not< b$	<code>a \leq b</code>	$a \leq b$
<code>a \cong b</code>	$a \cong b$	<code>a \prec b</code>	$a \prec b$	<code>a \nleq b</code>	$a \not\leq b$
<code>a \sim b</code>	$a \sim b$	<code>a \nprec b</code>	$a \not\prec b$	<code>a \ll b</code>	$a \ll b$
<code>a > b</code>	$a > b$	<code>a \geq b</code>	$a \geq b$	<code>a \gtreqless b</code>	$a \gtrless b$
<code>a \ngtr b</code>	$a \not> b$	<code>a \ngeq b</code>	$a \not\geq b$	<code>a \lesseqgtr b</code>	$a \lesseqgtr b$

Most of the relation symbols with “eq” in their name also have a variant with “eqq” in the name that prints them with a double rather than a single equality sign:

<code>a \geqq b</code>	$a \geqq b$	<code>a \leqq b</code>	$a \leqq b$	<code>a \gtreqqless b</code>	$a \gtreqqless b$
<code>a \ngeqq b</code>	$a \not\geqq b$	<code>a \nleqq b</code>	$a \not\leqq b$	<code>a \lesseqqgtr b</code>	$a \lesseqqgtr b$

6.7. Fractions

To write fractions, use the `\frac` command with two sets of braces—one for the numerator and one for the denominator. If necessary, you can nest `\frac` commands:

```
\begin{equation*}
\frac{dx}{dy}
= -\frac{\frac{\partial F}{\partial y}}{\frac{\partial F}{\partial y}}
= -\frac{\frac{1}{2}xy^{-\frac{1}{2}}}{y^{\frac{1}{2}}}
= -\frac{1}{2}\frac{x}{y}
\end{equation*}
```

$$\frac{dx}{dy} = -\frac{\frac{\partial F}{\partial y}}{\frac{\partial F}{\partial y}} = -\frac{\frac{1}{2}xy^{-\frac{1}{2}}}{y^{\frac{1}{2}}} = -\frac{1}{2}\frac{x}{y}$$

Notice that L^AT_EX automatically adjusts the fractions’ size depending on the context. If you disagree with those adjustments, use the alternative `\dfrac` command to guarantee that a fraction is displayed at normal (“display”) size, or the `\tfrac` command to guarantee that it is displayed at small (“text”) size:

```

\begin{equation*}
\frac{dx}{dy}
= -\frac{\frac{\partial F}{\partial x}}{\frac{\partial F}{\partial y}}
= -\frac{\frac{1}{2}xy^{-\frac{1}{2}}}{y^{\frac{1}{2}}}
= -\frac{1}{2}\frac{x}{y}
\end{equation*}

```

$$\frac{dx}{dy} = -\frac{\frac{\partial F}{\partial x}}{\frac{\partial F}{\partial y}} = -\frac{\frac{1}{2}xy^{-\frac{1}{2}}}{y^{\frac{1}{2}}} = -\frac{1}{2}\frac{x}{y}$$

6.8. Sub- and superscripts

Use `_` for subscripts and `^` for superscripts. If you need both, use both `_` and `^`, in either order. Also, if the sub- or superscript is a more complex expression, surround it by braces:

```

K_t^\alpha      K_t^\alpha
L^\beta_s      L_s^\beta
K^{\frac{1-\alpha}{\alpha}}  K^{\frac{1-\alpha}{\alpha}}
L_{i,t+1}      L_{i,t+1}

```

6.9. Delimiters

```

(a + b)  (a + b)  [a + b]  [a + b]  \{a + b\}  {a + b}  |a + b|  |a + b|

```

Notice that you need to escape (i.e., use a backslash before) braces, because \LaTeX will otherwise ignore them.

If you want bigger versions of the delimiters, do this:

```

\bigl(\frac{a}{b}\bigr)  (\frac{a}{b})
\Bigl(\frac{a}{b}\Bigr)  (\frac{a}{b})
\biggl(\frac{a}{b}\biggr)  (\frac{a}{b})
\Biggl(\frac{a}{b}\Biggr)  (\frac{a}{b})

```

This works for brackets, braces, and vertical bars as well (use, for example, `\Bigl\{` and `\Bigr\}` to get large-ish braces).

Alternatively, you can have \LaTeX adjust the size of the delimiters automatically to the size of whatever is between them, by using the `\left` and `\right` commands:

```

\left(a + b\right)
\left(\dfrac{a}{b}\right)
\left(\dfrac{K}{L}\right)^{1-\alpha}
\left(\dfrac{\dfrac{L}{K}}{\dfrac{f_K}{f_L}}\right)

```

$$\left(\begin{array}{c} (a + b) \\ \left(\frac{a}{b}\right) \\ \left(\frac{K}{L}\right)^{1-\alpha} \\ \left(\frac{\frac{L}{K}}{\frac{f_K}{f_L}}\right) \end{array} \right)$$

Importantly, the `\bigl`, `\bigr`, etc., commands work across breaks in equations, but the `\left`, `\right` commands don't. So you can write, for example

```

\begin{multline*}
\dfrac{dL}{dp} = \dfrac{1}{|H|}
\Bigl\{
-p\alpha^2(\alpha-1)\beta
K^{2\alpha-2}L^{2\beta-1}
+ p\alpha^2\beta
K^{2\alpha-2}L^{2\beta-2}
\Bigr\}
\end{multline*}

```

$$\frac{dL}{dp} = \frac{1}{|H|} \left\{ -p\alpha^2(\alpha-1)\beta K^{2\alpha-2}L^{2\beta-1} + p\alpha^2\beta K^{2\alpha-2}L^{2\beta-2} \right\}$$

but \LaTeX would complain with an error message if you tried this with `\left\{` and `\right\}`.

A special case is the vertical-bar notation used to represent evaluating a function at particular values of its arguments. For some reason, you need to use the combination `\left.` (note the period) and `\right|` (note the vertical bar) to have \LaTeX stretch the vertical bar to the appropriate height:

```

\begin{equation*}
\left.\dfrac{dW(q)}{dq}\right|_{q=q_o}
= N[u'(q_o) - c'(q_o)] = 0
\end{equation*}

```

$$\left.\frac{dW(q)}{dq}\right|_{q=q_o} = N[u'(q_o) - c'(q_o)] = 0$$

6.10. Math accents

```

\hat{a}
\widehat{M}
\tilde{a}
\widetilde{w}
\bar{a}
\overline{a}
\underline{a}
\check{a}
a^*
a^{**}
\dot{a}
\ddot{a}
a^*
a^{**}
\dot{a}
\ddot{a}

```

6.11. Arrows

Use double arrows do denote logical implications, single ones for things like limits:

```

A \Rightarrow B
A \nrightarrow B
A \Leftarrow B
A \nLeftarrow B
A \Rightarrow B
A \Leftarrow B
A \Leftrightarrow B
A \nLeftrightarrow B
A \Rightarrow B
A \Leftarrow B
A \Leftrightarrow B
A \nLeftrightarrow B
x \rightarrow \infty
x \uparrow b
x \downarrow 0

```

6.12. Products, summations, and integrals

<code>\prod_{i=1}^N a_i</code>	$\prod_{i=1}^N a_i$	<code>\sum_{t=1}^T \frac{x}{(1+r)^t}</code>	$\sum_{t=1}^T \frac{x}{(1+r)^t}$
<code>\prod\limits_{i=1}^N a_i</code>	$\prod_{i=1}^N a_i$	<code>\sum\limits_{t=1}^T \frac{x}{(1+r)^t}</code>	$\sum_{t=1}^T \frac{x}{(1+r)^t}$

Notice that you use sub- and superscripts to indicate the limits over which a product, summation, or integral operates, and you add the command `\limits` if you want those limits to be typeset under and above the operators rather than next to them. The same is true for integrals as well:

<code>\int_{t=0}^{\infty} e^{-rt}x \, dt</code>	$\int_{t=0}^{\infty} e^{-rt}x \, dt$
<code>\int\limits_{t=0}^{\infty} e^{-rt}x \, dt</code>	$\int_{t=0}^{\infty} e^{-rt}x \, dt$

6.13. Special functions

L^AT_EX recognizes the following commands as math functions, and does not italicize them:⁷

<code>\ln(x)</code>	$\ln(x)$	<code>\arg\max(x)</code>	$\arg \max(x)$
<code>\log(x)</code>	$\log(x)$	<code>\arg\min(x)</code>	$\arg \min(x)$
<code>\exp(x)</code>	$\exp(x)$	<code>\lim(x)</code>	$\lim(x)$
<code>\min(x)</code>	$\min(x)$	<code>\det(x)</code>	$\det(x)$
<code>\max(x)</code>	$\max(x)$	<code>\Pr(x)</code>	$\Pr(x)$

In economics, the `\min` and `\max` functions are often used to describe optimization problems, and it is standard practice to then write the choice variables w.r.t. which the objective function is optimized (as well as sometimes the choice set) immediately below the min and max. In L^AT_EX, you can do this using subscript expressions:

<code>\begin{equation*}</code> <code>\max_{K,L} \pi = pf(K,L) - wL - rK</code> <code>\end{equation*}</code>	$\max_{K,L} \pi = pf(K,L) - wL - rK$
---	--------------------------------------

<code>\begin{equation*}</code> <code>\max_{m \in \{0,1\}} U =</code> <code>(1-m)u(H,I) + m\overline{u}</code> <code>\end{equation*}</code>	$\max_{m \in \{0,1\}} U = (1 - m)u(H, I) + m\bar{u}$
---	--

Similarly, use subscript expressions to indicate what variable is taken to what limit:

<code>\begin{equation*}</code> <code>e \equiv \lim_{n \rightarrow \infty} \left[1 + \frac{1}{n} \right]^n</code> <code>\end{equation*}</code>	$e \equiv \lim_{n \rightarrow \infty} \left[1 + \frac{1}{n} \right]^n$
--	---

⁷It recognizes trigonometric functions as well, but those are rarely used in economics.

6.14. Boldface math

For matrix algebra, you sometimes want to use boldface math. Unfortunately, the standard L^AT_EX command `\mathbf` (short for “math bold font”) has the drawback that it doesn’t bold Greek letters:

```
\begin{equation*}
\mathbf{y} = X\beta + \varepsilon
\end{equation*}
```

$$y = X\beta + \varepsilon$$

To get around this, include the `bm` (short for “bold math”) package in your preamble,

```
\usepackage{bm}
```

after which you’ll be able to use the `\bm` command instead. Unfortunately, *that* command has the drawback that it leaves bold variables in italics:

```
\begin{equation*}
\bm{y} = X\beta + \varepsilon
\end{equation*}
```

$$y = X\beta + \varepsilon$$

To get the best of both worlds (and the standard formatting used in econometrics), you’ll have to use a mix of both commands:

```
\begin{equation*}
\mathbf{y} = \mathbf{X}\bm{\beta}
+ \bm{\varepsilon}
\end{equation*}
```

$$y = X\beta + \varepsilon$$

See Section 7 below for how this can be made much less painful than it seems, by defining shortcuts.

6.15. Matrices

Use the `\bmatrix` command to generate matrices. Matrix columns are separated by ampersands and matrix rows by double backslashes:

```
\begin{equation*}
\begin{bmatrix}
a & b \\
c & d
\end{bmatrix}
\end{equation*}
```

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Here’s an extended, more realistic example, the structure of which you’ll very frequently need to use in applied microeconomics:

Totally differentiating the first-order conditions

```
\begin{align*}
p\alpha K^{\alpha-1}L^{\beta} - r &= 0 \\
p\beta K^{\alpha}L^{\beta-1} - w &= 0
\end{align*}
```

w.r.t. K , L , and p gives

```
\begin{equation*}
\begin{bmatrix}
p\alpha(\alpha-1)K^{\alpha-2}L^{\beta} & p\alpha\beta K^{\alpha-1}L^{\beta-1} \\
\end{bmatrix}
\end{equation*}
```

```

p\alpha\beta K^{\alpha-1}L^{\beta-1} & p\beta(\beta-1) K^{\alpha}L^{\beta-2}
\end{bmatrix}
\begin{bmatrix}
dK \\
dL
\end{bmatrix}
=
\begin{bmatrix}
-\alpha K^{\alpha-1}L^{\beta} \\
-\beta K^{\alpha}L^{\beta-1}
\end{bmatrix}
dp.
\end{equation*}

```

Totally differentiating the first-order conditions

$$p\alpha K^{\alpha-1}L^{\beta} - r = 0$$

$$p\beta K^{\alpha}L^{\beta-1} - w = 0$$

w.r.t. K , L , and p gives

$$\begin{bmatrix} p\alpha(\alpha-1)K^{\alpha-2}L^{\beta} & p\alpha\beta K^{\alpha-1}L^{\beta-1} \\ p\alpha\beta K^{\alpha-1}L^{\beta-1} & p\beta(\beta-1)K^{\alpha}L^{\beta-2} \end{bmatrix} \begin{bmatrix} dK \\ dL \end{bmatrix} = \begin{bmatrix} -\alpha K^{\alpha-1}L^{\beta} \\ -\beta K^{\alpha}L^{\beta-1} \end{bmatrix} dp.$$

Use the `\vmatrix` command for determinants:

```
By Cramer's Rule,
\begin{equation*}
\frac{dL}{dp} = \frac{\begin{vmatrix} p^\alpha(\alpha-1)K^{\alpha-2}L^\beta & -\alpha K^{\alpha-1}L^\beta \\ p^\alpha\beta K^{\alpha-1}L^{\beta-1} & -\beta K^\alpha L^{\beta-1} \end{vmatrix}}{\begin{vmatrix} p^\alpha(\alpha-1)K^{\alpha-2}L^\beta & p^\alpha\beta K^{\alpha-1}L^{\beta-1} \\ p^\alpha\beta K^{\alpha-1}L^{\beta-1} & p^\beta(\beta-1)K^\alpha L^{\beta-2} \end{vmatrix}}.
\end{equation*}
```

By Cramer's Rule,

$$\frac{dL}{dp} = \frac{\begin{vmatrix} p^\alpha(\alpha-1)K^{\alpha-2}L^\beta & -\alpha K^{\alpha-1}L^\beta \\ p^\alpha\beta K^{\alpha-1}L^{\beta-1} & -\beta K^\alpha L^{\beta-1} \end{vmatrix}}{\begin{vmatrix} p^\alpha(\alpha-1)K^{\alpha-2}L^\beta & p^\alpha\beta K^{\alpha-1}L^{\beta-1} \\ p^\alpha\beta K^{\alpha-1}L^{\beta-1} & p^\beta(\beta-1)K^\alpha L^{\beta-2} \end{vmatrix}}.$$

6.16. Dots

Technically, I should call these “ellipses.” L^AT_EX has “low” `\ldots` and “centered” `\cdots`

```
\begin{equation*}
\frac{\partial U}{\partial x_i} = u'(x_i) - p_i,
\quad i = 1, 2, \ldots, n
\end{equation*}
```

$$\frac{\partial U}{\partial x_i} = u'(x_i) - p_i, \quad i = 1, 2, \dots, n$$

```
\begin{equation*}
\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k]
\end{equation*}
```

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k]$$

as well as “vertical” `\vdots`, and “diagonal” `\ddots`:

```
\begin{equation*}
H \equiv
\begin{bmatrix}
\pi_{11} & \pi_{12} & \cdots & \pi_{1m} \\
\pi_{21} & \pi_{22} & \cdots & \pi_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
\pi_{m1} & \pi_{m2} & \cdots & \pi_{mm}
\end{bmatrix}
\end{equation*}
```

$$H \equiv \begin{bmatrix} \pi_{11} & \pi_{12} & \cdots & \pi_{1m} \\ \pi_{21} & \pi_{22} & \cdots & \pi_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{m1} & \pi_{m2} & \cdots & \pi_{mm} \end{bmatrix}$$

6.17. Over- and underbraces

In applied microeconomics, you’ll very often find yourself having to sign an expression. In such cases, you’ll find the \LaTeX commands `\overbrace` and `\underbrace` useful:

```
\begin{equation*}
\frac{dI}{dW_0} = -\frac{
\overbrace{p'U''(A)(1-q)}^{(-)}
- \overbrace{(1-p)U''(B)q}^{(-)}
}{
\underbrace{D}_{(-)}
}
\gtreqless 0.
\end{equation*}
```

$$\frac{dI}{dW_0} = -\frac{\overbrace{pU''(A)(1-q)}^{(-)} - \overbrace{(1-p)U''(B)q}^{(-)}}{\underbrace{D}_{(-)}} \leq 0.$$

Underbraces are also useful to indicate what various parts of your equations represent:

```
\begin{equation*}
\frac{\partial EU}{\partial x} =
\underbrace{
\overbrace{p'(x)[U(A) - U(A)]}^{0}
}_{\text{MB of self-protection}}
\quad - \quad
\underbrace{
\overbrace{U'(A)}^{(+)}
}_{\text{MC of self-protection}}
\end{equation*}
```

$$\frac{\partial EU}{\partial x} = \underbrace{p'(x)[U(A) - U(A)]}_{\text{MB of self-protection}} - \underbrace{U'(A)}_{\text{MC of self-protection}}$$

Notice how you can combine braces by nesting them.

6.18. Math spacing

As you may have noticed already in previous examples, sometimes \LaTeX needs a little help in order to get the spacing of math elements right. The `\hspace*` and `\vspace*` commands can be used for this, but there are a variety of other spacing commands that are specific to math environments:

(no space) ab	ab	(no space) ab	ab
a \thinspace b or a \, b	$a b$	a \negthinspace b	$a b$
a \medspace b or a \: b	$a b$	a \negmedspace b or a \! b	$a b$
a \thickspace b or a \; b	$a b$	a \negthickspace b	$a b$
a \quad b	$a \quad b$		
a \qquad b	$a \qquad b$		

Integral expressions, for example, look better if you add a little space between the integrand and the dx part, where x is the dummy variable of integration. Usually a “\,” will do the trick:

$\int_{t=0}^{\infty} e^{-rt} x dt$	$\int_{t=0}^{\infty} e^{-rt} x dt$
$\int_{t=0}^{\infty} e^{-rt} x \, dt$	$\int_{t=0}^{\infty} e^{-rt} x dt$

6.19. Text in math

Sometimes you’ll want to use some text inside a mathematical expression. In such cases, wrap the text in braces and use the `\text` command. Usually, you’ll need to add space before and after:

```

The consumer’s optimization problem is
\begin{equation*}
\max_{x,y} U = U(x,y)
\quad \text{subject to} \quad
I = p_x x + p_y y.
\end{equation*}

```

<p>The consumer’s optimization problem is</p> $\max_{x,y} U = U(x,y) \quad \text{subject to} \quad I = p_x x + p_y y.$
--

6.20. Cases

The `cases` environment useful for describing functions with a discontinuity:

```

\begin{equation*}
t_i =
\begin{cases}
1 & \text{if } p \geq x_i, \\
0 & \text{otherwise.}
\end{cases}
\end{equation*}

```

$t_i = \begin{cases} 1 & \text{if } p \geq x_i, \\ 0 & \text{otherwise.} \end{cases}$

Notice the use of ampersands to align the text strings.

6.21. Roots

Use `\sqrt` for square roots, `\sqrt[3]` for cubic roots, `\sqrt[4]` for quartic roots, etc.:

$\sqrt{a + b}$	$\sqrt{a + b}$
$\sqrt[3]{a + b}$	$\sqrt[3]{a + b}$
$\sqrt[4]{a + b}$	$\sqrt[4]{a + b}$

6.22. Sets

The `\mathbb` (“math blackboard font”) command is often used to denote the special sets of the natural numbers, integers, rational numbers, and reals. For other, arbitrary sets, the `\mathcal` (“math calligraphic font”) command is often used:

<code>\mathbb{N}</code>	\mathbb{N}	<code>\mathcal{A}</code>	\mathcal{A}
<code>\mathbb{Z}</code>	\mathbb{Z}	<code>\mathcal{B}</code>	\mathcal{B}
<code>\mathbb{Q}</code>	\mathbb{Q}	<code>\mathcal{C}</code>	\mathcal{C}
<code>\mathbb{R}</code>	\mathbb{R}		

Here are some more or less random statements to illustrate other set-related symbols:

```
\newcommand{\R}{\rightarrow}
```

```
\varnothing = \{\}
```

```
x \in S
```

```
x \notin T
```

```
S \cap T = \{x: x \in S \wedge x \in T\}
```

```
S \cup T = \{x: x \in S \vee x \in T\}
```

```
S \setminus T = \{x: x \in S \wedge x \notin T\}
```

```
S \subseteq T \R \forall x: x \in S \R x \in T
```

```
S \not\subseteq T \R \exists x: x \in S \wedge x \notin T
```

```
S \subset T \R S \subseteq T \wedge S \neq T
```

```
S \not\subset T \R \not\exists x: x \notin S \wedge x \in T
```

$$\emptyset = \{\}$$

$$x \in S$$

$$x \notin T$$

$$S \cap T = \{x : x \in S \wedge x \in T\}$$

$$S \cup T = \{x : x \in S \vee x \in T\}$$

$$S \setminus T = \{x : x \in S \wedge x \notin T\}$$

$$S \subseteq T \Rightarrow \forall x : x \in S \Rightarrow x \in T$$

$$S \not\subseteq T \Rightarrow \exists x : x \in S \wedge x \notin T$$

$$S \subset T \Rightarrow S \subseteq T \wedge S \neq T$$

$$S \not\subset T \Rightarrow \exists x : x \notin S \wedge x \in T$$

7. SHORTCUTS

Whenever you find yourself repeatedly typing a complicated or lengthy string into your `.tex` file, you can save yourself time and effort by defining a \LaTeX shortcut.

Notice how I did this in the previous example: to avoid having to type `\rightarrow` all the time, I simply typed

```
\newcommand{\R}{\rightarrow}
```

once before the list of example statements, after which I could use the abbreviation `\R` instead. More generally, you’ll want to define new commands in the preamble, to make sure that they’re available throughout the document. This also makes it easy to copy your favorite shortcuts over to other documents, since you can simply copy over the preamble.

In the preamble of this document, for example, I defined the shortcut

```
\renewcommand{\L}{\LaTeX}
```

to avoid having to type `\LaTeX` all the time. Notice that I used `\renewcommand` rather than `\newcommand`. The reason is that when I tried using `\newcommand`, I got an error message from \LaTeX saying that the command `\L` was already defined. This just means that somewhere in the zillions of files that make up the standard \LaTeX distribution, or perhaps in one of the packages that I included in this document’s preamble, some programmer already used `\newcommand` do

define `\L` as something. In such cases, you can either use a different shortcut string, or you can override the existing command by using `\renewcommand`. The latter method is slightly risky, because by overriding the current definition of `\L` you might “break” something, causing L^AT_EX to do unexpected things when you try compiling this document.

Something else to be aware of is that L^AT_EX by default ignores any spaces that follow a command. As a result, if I were to type either “`\LaTeX`” or (now that I have defined the shortcut) “`\L`”, the resulting L^AT_EX string will be smushed together with whatever word follows it. To avoid this, you have to “escape” the space after the command by putting a backslash in front:

```
\L by default ignores any
spaces that follow a command
```

```
LATEXby default ignores any spaces that follow a
command
```

```
\L\ by default ignores any
spaces that follow a command
```

```
LATEX\ by default ignores any spaces that follow a
command
```

Another example of a shortcut that I use frequently is

```
\newcommand{\CO}{CO$_{}_2$}
```

It allows me to type `\CO` instead of the cumbersome `\CO$_{}_2$` to generate the string “CO₂” in my papers on CO₂-enhanced oil recovery and global warming.

To abbreviate various total and partial derivative expressions I have the following shortcuts defined in all my preambles:⁸

```
\newcommand{\td}[2]{\dfrac{\partial #1}{\partial #2}}
\newcommand{\std}[2]{\dfrac{d^2 #1}{d #2^2}}
\newcommand{\ctd}[3]{\dfrac{d^2 #1}{d #2 d #3}}

\newcommand{\pd}[2]{\dfrac{\partial #1}{\partial #2}}
\newcommand{\spd}[2]{\dfrac{\partial^2 #1}{\partial #2^2}}
\newcommand{\cpd}[3]{\dfrac{\partial^2 #1}{\partial #2 \partial #3}}
```

This allows me to, for example, write

```
\begin{align*}
  \dfrac{\partial \pi}{\partial K} &= \cdots \\
  \dfrac{\partial^2 \pi}{\partial K \partial L} &= \cdots
\end{align*}
```

$$\frac{\partial \pi}{\partial K} = \cdots$$

$$\frac{\partial^2 \pi}{\partial K \partial L} = \cdots$$

much more simply as

```
\begin{align*}
  \pd{\pi}{K} &= \cdots \\
  \cpd{\pi}{K}{L} &= \cdots
\end{align*}
```

$$\frac{\partial \pi}{\partial K} = \cdots$$

$$\frac{\partial^2 \pi}{\partial K \partial L} = \cdots$$

⁸A “cross-total derivative” such as $d^2 f/dx dy$ may seem a contradiction in terms, but in practice it’s sometimes useful to distinguish between the direct and the indirect effect of, say, y on the derivate of $f(x, y)$ w.r.t. x , when x is itself a function of y .

Notice that you can define a shortcut to have variable arguments, by writing the number of arguments in brackets after the abbreviation, and then referring to the arguments as #1, #2, etc., in the expanded string. An abbreviation with n arguments will be used with n sets of braces after it.

Some frequently used symbols that I define in my preamble with shortcuts are

```
\newcommand{\Lg}{\mathcal{L}}
```

for the Lagrangian symbol \mathcal{L} ,

```
\newcommand{\half}{\tfrac{1}{2}}
```

for the fraction $\frac{1}{2}$,

```
\newcommand{\LR}{\Leftrightarrow}
```

for the equivalency symbol \Leftrightarrow , and

```
\newcommand{\eqs}{\buildrel s \over =}
```

do denote equality of sign as $\stackrel{s}{=}$. This latter symbol is non-standard: I find it very useful, but in papers I make sure to explain it to the reader whenever I use it for the first time:

```
\begin{align*}
&\td{I}{W_0} \stackrel{s}{=} -\dfrac{
pU''(A)(1-q) - (1-p)U''(B)q
}{
\underbrace{pU''(A)(1-q)^2
+ (1-p)U''(B)q^2}_{(-)}
} \\
&\&\eqs \\
&pU''(A)(1-q) - (1-p)U''(B)q,
\end{align*}
where “ $\stackrel{s}{=}$ ” denotes equality of sign.
```

$$\frac{dI}{dW_0} = -\frac{pU''(A)(1-q) - (1-p)U''(B)q}{\underbrace{pU''(A)(1-q)^2 + (1-p)U''(B)q^2}_{(-)}} \stackrel{s}{=} pU''(A)(1-q) - (1-p)U''(B)q,$$

where “ $\stackrel{s}{=}$ ” denotes equality of sign.

When I use bold math for matrix algebra, I avoid having to repeatedly surround vector symbols with either $\mathbf{\}$ or $\mathbf{\}$ depending on whether the symbol is a Roman or a Greek letter (see Subsection 6.14) by simply creating abbreviations for all vectors in the preamble. Defining, say

```
\newcommand{\vy}{\mathbf{y}}
\newcommand{\vX}{\mathbf{X}}
\newcommand{\vbeta}{\mathbf{\beta}}
\newcommand{\veps}{\mathbf{\varepsilon}}
```

allows me to subsequently type

```
\begin{equation*}
\vy = \vX\vbeta + \veps
\end{equation*}
```

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

An important gotcha with shortcuts is that L^AT_EX does not allow the abbreviation to contain any numbers. It will complain with a very cryptic error message if you try to define, say

```
\newcommand{\f12}{\dfrac{\partial^2 f}{x_1x_2}}
```

even though that would seem to be a perfectly reasonable thing to do.

8. GRAPHICS

As mentioned above, in Section 3, to easily import graphics into your document you should include the `graphicx` package in your preamble. To then, say, import a graphics file called `delta_func.pdf`, use the `figure` environment and the `includegraphics` command in the following way

```
\begin{figure}[h!]  
  \centering\includegraphics[width=0.5\textwidth]{delta_func}  
  \caption{Initial oil production and \CO sequestration as a function of  
    the initial \CO injection rate.\label{fig:delta_func}}  
\end{figure}
```

This generates

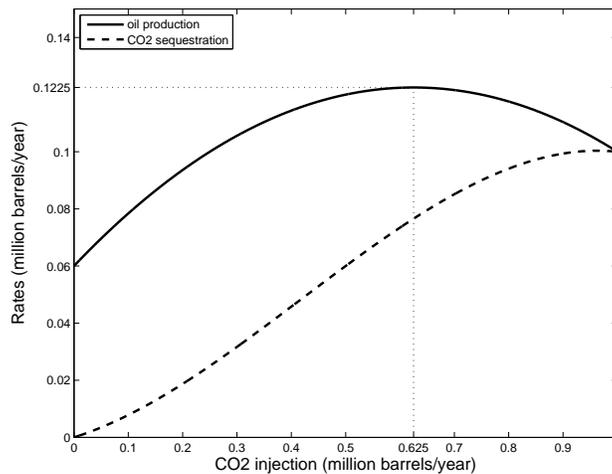


Figure 1: Initial oil production and CO₂ sequestration as a function of the initial CO₂ injection rate.

The `[h!]` option after the `\begin{figure}` command tells L^AT_EX to place the figure right “here!,” exactly where you put the `figure` environment relative to the rest of the text. If instead you want L^AT_EX to place the figure at the top of the page, replace `[h!]` by `[t]`.

The `\centering` command tells L^AT_EX to center the figure relative to the page margins.

The `[width=0.5\textwidth]` option to the `\includegraphics` command tells L^AT_EX to scale the figure so that its width is exactly half of the width of the text. If you want to scale the figure up or down, just change the number 0.5 accordingly.

In order to keep my folders from getting too cluttered, I often put figure files in a subfolder called `figures`. In that case, I would change the the `\includegraphics` command to

```
\includegraphics[width=0.5\textwidth]{figures/delta_func}
```

and L^AT_EX would know to look for the `delta_func.pdf` file in that subfolder.

Incidentally, graphics files don’t have to be in `.pdf` format. They can also be in `.jpg`, `.png`, or `.tif` format, and for those, too, you should just leave off the `.xxx` extension in the argument to the `\includegraphics` command.

The `\caption` command obviously specifies the figure caption, and the `\label` command creates a label that you can use to refer to the figure in the text (see Section 11 below).

If you include the `subfigure` package in your preamble,

```
\usepackage{subfigure}
```

then you can also do fancier stuff like

```
\setcounter{subfigure}{0}
\begin{figure}[h!]
  \centering
  \subfigure[] {
    \includegraphics[width=0.4\textwidth]{co2oil}}
  \hspace*{0.2in}
  \subfigure[] {
    \includegraphics[width=0.4\textwidth]{shadow}}
  \caption{\CO injection, production, and sequestration, and oil production
    (a) and the shadow price (b) over time.\label{fig:co2oil_shadow}}
\end{figure}
```

to get figures with subfigures:

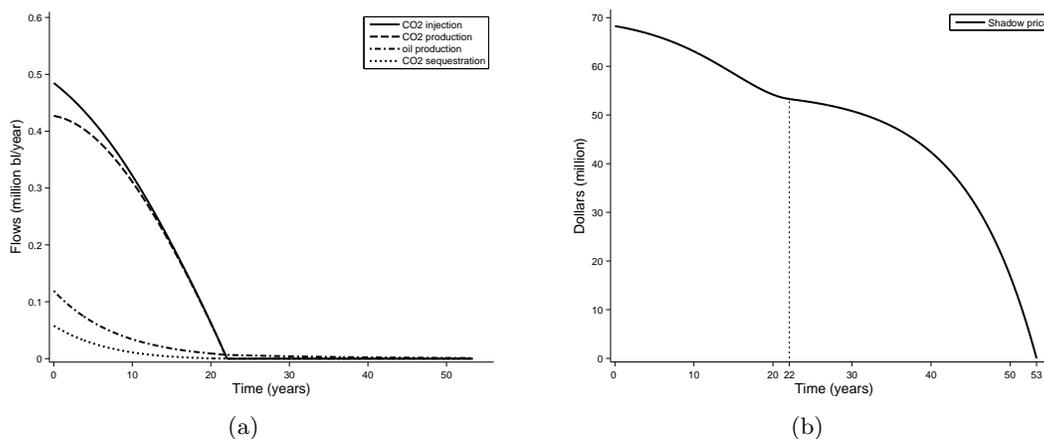


Figure 2: CO₂ injection, production, and sequestration, and oil production (a) and the shadow price (b) over time.

The `\setcounter{subfigure}{0}` line just tells \LaTeX to start over labeling the subfigures as (a), (b), etc., each time you do include subfigures in the document, rather than labeling them (c), (d), etc., next time.

Importantly, because \LaTeX freshly pulls in figure files every time the `.tex` file is compiled, any changes you make to the figures in whatever program you use to create them (MATLAB, Stata, ...) will automatically be reflected in your \LaTeX `.pdf`.

9. TABLES

Use the `tabular` environment to generate tables in \LaTeX . As with matrices (see Subsection 6.15), the columns of the table are separated by ampersands, while the rows are separated by double backslashes. Different from matrices, however, the `tabular` environment expects the entries to be regular text, so if you want to enter any math expressions, you have to surround them with `$` signs. Another difference is that you have to indicate immediately after the `\begin{tabular}` command how you want your columns to be aligned, and what, if anything, you want in between.

Here's a very simple example that lists the possible permutations of two multipliers for a Kuhn-Tucker problem. The following code

```
\begin{center}
\begin{tabular}{|c|c|c|}\hline
Case &  $\lambda$  &  $\mu$ \\\hline
1 & + & +\\
2 & + & 0\\
3 & 0 & +\\
4 & 0 & 0\\\hline
\end{tabular}
\end{center}
```

generates this table:

Case	λ	μ
1	+	+
2	+	0
3	0	+
4	0	0

The `|c|c|c|` string tells \LaTeX that you want three centered columns separated by vertical separator lines, and you want vertical lines along the left and right margins of the table as well. To get left- or right-aligned columns, use `l` or `r` instead of `c`.

The three `\hline` commands tells \LaTeX to draw horizontal lines at the top of the table, after the first row, and after the last row of the table.

Changing these specifications for example to

```
\begin{center}
\begin{tabular}{r|@{\quad}c c}
Case &  $\lambda$  &  $\mu$ \\\hline
1 & + & +\\
2 & + & 0\\
3 & 0 & +\\
4 & 0 & 0\\
\end{tabular}
\end{center}
```

instead generates this table:

Case	λ	μ
1	+	+
2	+	0
3	0	+
4	0	0

In the new alignment specification `r|@{\quad}c c`, the `@{\quad}` string tells \LaTeX to add exactly one `\quad`'s worth of space between the first and second columns, immediately following the vertical separator line. More generally, you can replace the default space between any two table columns with any string or alternative space you want, by inserting a `@` symbol between the alignment specifiers for those columns, and then specifying whatever replacement you want inside braces following that symbol. If you just insert `@{}`, then that tells \LaTeX to replace the default space between columns by nothing, implying that the columns will be smushed together.

Here's a much more involved example, which reports regression results:

```

\begin{table}[h!]
\begin{small}
\begin{center}
\begin{tabular}{l|*{2}{r@{}}l}
\hline
Estimation
method      & \multicolumn{2}{c}{PROBIT}& \multicolumn{2}{c}{OLS} \\
Dependent
variable    & \multicolumn{2}{c}{P.FWD.TR}& \multicolumn{2}{c}{L.FWD.TR} \\
\hline
N.HHMEM    & 0.112***& & -0.052* & \\
L.LAND     & 1.920** & & 0.337***& \\
L.LAND.SQ  & -0.116** & & & \\
L.BIO      & 1.340* & & -0.129** & \\
L.BIO.SQ   & -0.088** & & & \\
PRICE.FWD  & -0.448 & & 0.511 & \\
WAGE.LS.M  & -0.070** & & -0.034 & \\
CONSTANT   & -15.281***& & 1.804 & \\
\hline
No. of obs. & 539 & & 539 & \\
$R^2$       & & & 0.28 & \\
\hline
\end{tabular}
\end{center}
\vspace*{1em}
\caption{Ownership of private fuelwood trees.\label{tbl:fuel}}
\end{small}
\end{table}

```

It generates this table:

Estimation method	PROBIT	OLS
Dependent variable	P.FWD.TR	L.FWD.TR
N.HHMEM	0.112***	-0.052*
L.LAND	1.920**	0.337***
L.LAND.SQ	-0.116**	
L.BIO	1.340*	-0.129**
L.BIO.SQ	-0.088**	
PRICE.FWD	-0.448	0.511
WAGE.LS.M	-0.070**	-0.034
CONSTANT	-15.281***	1.804
No. of obs.	539	539
R^2		0.28

TABLE 1. Ownership of private fuelwood trees.

The `table` environment added on the outside is there just as a “wrapper,” very similar to the `figure` environment. As with the `figure` environment, having it allows you to specify the placement option `[h!]` or `[t]`, and also allows you to add a `\caption` with a `\label`.

The alignment specification `l|*{2}{r@{}}1` tells \LaTeX to generate a left-aligned column followed by a vertical line (`|`) and to then twice (`*{2}`) repeat the specification `r@{}1`, i.e., generate a right-aligned column followed by a left-aligned one, with no space in between. This nifty trick allows you to put the coefficient estimates in the right-aligned columns, and the “significance stars” (`*`, `**`, or `***`, if the estimate is significant at respectively the 10%, 5%, or 1% level) in the left-aligned columns.

The `\multicolumn` commands merge cells of adjacent columns. The `\multicolumn{2}{c}{PROBIT}` command, for example, merges the two (`{2}`) cells in the second and third column of the top row into a single, centered (`{c}`) cell containing the text `PROBIT`.

Very importantly, this table was not typed in by hand! It was generated directly by Stata, using its `estout` command for generating \LaTeX tables.⁹ The command put the code shown above in a file called `fuel.tex`, which I then incorporated into this `.tex` file by simply writing¹⁰

```
\input{fuel}
```

When you start running your own regressions in your econometrics classes, you’ll soon realize how incredibly convenient this is. Any time you change anything in your regression specification—which in practice you do tens or hundreds of times—the updates are automatically included in your \LaTeX file, without you having to do any laborious (and error-prone!) copying over of new results.

10. CITATIONS

\LaTeX makes it very easy to manage literature citations in your paper. If you have any such citations, you need to add two lines like the following to the end of your paper, at the point where you want your References section to appear:

```
\bibliographystyle{dcu}
\bibliography{eor}
```

The `\bibliographystyle` command specifies a particular style of citing papers in the list of references. Generally, `dcu` is a good choice for economics journals, but some journals have their own \LaTeX bibliography style files that you can download from their website. In that case, simply replace `dcu` by the title of their style file, leaving off the `.bst` (short for “bibliography style”) filename extension. Put the `.bst` file in the same folder as your `.tex` file, and \LaTeX will automatically find it and apply the appropriate formatting.

The `\bibliography` command specifies the name of a file that must also be sitting in the same folder as your `.tex` file, with filename extension `.bib`. This is the file in which you put all the information for any papers, books, websites, etc., that you *might* (!) cite in your paper.

Here’s what part of my `eor.bib` file for literature related to enhanced oil recovery looks like:

```
@techreport{guo_etal:06,
  author =      "Guo, X. and Du, Z. and Sun, L. and Fu, Y.",
  title =       "Optimization of Tertiary Water-Alternate- $\text{CO}_2$  Flood in
                {Jilin} Oil Field of {China}: Laboratory and Simulation Studies",
  institution = "Society of Petroleum Engineers (SPE)",
  type =        "Working Paper",
  year =        "2006",
```

⁹A number of other Stata commands can do the same: `outreg2`, for example.

¹⁰To be honest, I did delete a few rows from the original table generated for a paper I wrote, in order to make the example fit on one page.

```

    number =      "99616"
}

@article{jessen_etal:05,
  author =      "Jessen, Kristian and Kovscek, Anthony R. and Orr Jr., Franklin M.",
  title =      "Increasing {CO}$_{2}$ storage in oil recovery",
  journal =    "Energy Conversion and Management",
  year =      "2005",
  volume =    "46",
  pages =     "293--311",
}

@book{ipcc:05,
  author =      "IPCC",
  note =      "{Intergovernmental Panel on Climate Change}",
  title =      "{IPCC} Special Report: Carbon Dioxide Capture and Storage",
  publisher =  "Cambridge University Press",
  address =    "Cambridge, UK",
  year =      "2005"
}

@phdthesis{mccoy:08,
  author =     "{McCoy}, Sean T.",
  title =     "The Economics of {CO}$_{2}$ Transport by Pipeline and
              Storage in Saline Aquifers and Oil Reservoirs",
  school =   "Carnegie Mellon University",
  year =    "2008",
  type =    "{Ph.D.} Thesis",
  address = "Pittsburgh, PA"
}

@mastersthesis{cakici:03,
  author =     "Cakici, M. D.",
  title =     "Co-optimization of Oil Recovery and Carbon Dioxide Storage",
  school =   "Department of Petroleum Engineering, Stanford University",
  year =    "2003",
  type =    "Engineer Thesis",
}

@incollection{babadagli:06,
  author =     "Babadagli, Tayfun",
  editor =     "Lombardi, S. and Altunina, L. K. and Beaubien, S. E.",
  title =     "Optimization of {CO}$_{2}$ Injection for Sequestration/Enhanced
              Oil Recovery and Current Status in {Canada}",
  booktitle = "Advances in the Geological Storage of Carbon Dioxide:
              International Approaches to Reduce Anthropogenic Greenhouse
              Gas Emissions",
  publisher = "Springer",
  address =   "Dordrecht, The Netherlands",
  year =     "2006",
  pages =    "261--270",
  subtitle = "{NATO} Science Series: {IV}: Earth and Environmental Sciences"
}

```

```

}

@inproceedings{zhou_etal:05,
  author = "Zhou, W. and Stenhouse, M. J. and Arthur, R. and
           Whittaker, S. and Law, D. H.-S. and Chalaturnyk, R. and
           Jazrawi, W.",
  title = "The {IEA Weyburn} {CO}_2$ Monitoring and Storage
           Project: Modeling of the Long-Term Migration of {CO}_2$ From
           {Weyburn}",
  booktitle = "Proceedings of the 7th {International Conference on
              Greenhouse Gas Control Technologies (GHGT-7)}, September
              5-9, 2004, Vancouver, Canada}",
  year = "2005",
  volume = "1",
  pages = "721--730",
  series = "Peer-Reviewed Papers and Plenary Presentations",
  address = "Oxford, {UK}",
  publisher = "Elsevier",
}

```

```

@misc{epa:08,
  author = "{EPA}",
  title = "Inventory of {U.S.} greenhouse gas emissions and sinks:
           1990-2006",
  year = "2008",
  note = "{U.S. Environmental Protection Agency, Washington, DC. EPA
           430-R-08-005}"
}

```

If you now type, for example

```
The cost parameters of the model are based on data in \citet{mccoy:08}.
```

then L^AT_EX will produce line

The cost parameters of the model are based on data in McCoy (2008).

in the body of your .pdf file, as well as entry

McCoy, S. T. (2008). *The Economics of CO₂ Transport by Pipeline and Storage in Saline Aquifers and Oil Reservoirs*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA.

in a References section that will automatically be generated at the end of your .pdf.

Similarly, if you type

```
The estimated sequestration capacity of 12 Gt\CO\ amounts to just
two years' worth of US \CO\ emissions \citep{ipcc:05,epa:08}.
```

then L^AT_EX will produce line

The estimated sequestration capacity of 12 GtCO₂ amounts to just two years' worth of US CO₂ emissions (IPCC, 2005; EPA, 2008).

in the body of your .pdf (note that the citations are in parentheses), as well as entries

EPA (2008). Inventory of U.S. greenhouse gas emissions and sinks: 1990-2006. U.S. Environmental Protection Agency, Washington, DC. EPA 430-R-08-005.

IPCC (2005). *IPCC Special Report: Carbon Dioxide Capture and Storage*, Cambridge University Press, Cambridge, UK. Intergovernmental Panel on Climate Change.

in your References section.

How exactly it does this varies a bit across L^AT_EX front-ends. In TeXworks, you have to follow the following four steps:

- (1) Compile the L^AT_EX file once by selecting “pdfLaTeX” from the pull-down menu at the top and pressing the green arrow button. This will cause L^AT_EX to create an auxiliary file with extension .aux in the same folder as your .tex file, in which all citations it finds in the text are listed.
- (2) Compile the L^AT_EX file a second time, this time selecting “BibTeX” from the pull-down menu and again pressing the green arrow button. This will cause L^AT_EX to compare the citations in the text with the entries in the .bib file that have the same label, and create another auxiliary file with extension .bbl that lists the collated results.
- (3) Compile the L^AT_EX file a third time, this time going back to selecting “pdfLaTeX” from the pull-down menu and once again pressing the green arrow button. This will cause L^AT_EX to add the references section at the end of the .pdf, but will not yet add the corresponding citations in the body of the .pdf. Instead, you’ll see bold double question marks (??) everywhere where the .tex file has a `\citet` or `\citep` command.
- (4) Compile the L^AT_EX file a fourth time, keeping the selection at “pdfLaTeX.” This will finally cause L^AT_EX to replace the double question marks with proper citations.

In TeXShop, the process is similar, except that in steps (1), (3), and (4) the pull-down menu entry is just called “LaTeX,” and you press a “TypeSet” button rather than a green arrow.¹¹

While this may seem pretty cumbersome, keep in mind that you only have to do steps (2) through (4) once, when you’re completely done with your paper. Until that point, L^AT_EX will still compile fine: you just won’t see any Reference section at the end, and only the double question marks will appear wherever you have citations in your text.

The big advantage of L^AT_EX’s approach is that if, say, you later decide to delete the line cite referring to McCoy’s thesis (as well as any other references to it elsewhere in your text), the bibliography entry for it will automatically be deleted from the References section. If you later change your mind again and add the cite back, then the bibliography entry will be added back again (provided that in the meantime you didn’t delete it from your .bib file, of course). You never have to keep track of anything! Moreover, you can re-use your .bib file for any other papers you might write.

By the way, several online journal databases (such as JSTOR and the Web of Knowledge databases offered through the Coe Library website) give you the option of downloading citations in BibTeX format, so you can just paste them into your .bib file.

11. INTERNAL REFERENCES

Just as L^AT_EX automatically keeps track of citations, it automatically keeps track of internal references to the paper’s sections, list items, equations, figures, tables, and pages as well. All it takes is to add `\label{xxx}` commands at any points in the paper that you might want to refer to (whether

¹¹One reason why I like to use vim as my front-end is that it does all four steps in one fell swoop.

earlier in the paper or later). You can then refer to them by using the `\ref{xxx}` command or for pages the `\pageref{xxx}` command, where `xxx` is an arbitrary but unique string.

For example, because I added a label under the section header for the section on Graphics

```
\section{Graphics}
\label{sec:graphics}
```

and also added labels `\label{fig:delta_func}` and `\label{fig:co2oil_shadow}` to the captions of the figures in that section, I can at any point in paper write, say

```
see Figure~\ref{fig:co2oil_shadow} in Section~\ref{sec:graphics}
```

to have L^AT_EX generate line

```
see Figure 2 in Section 8
```

Moreover, if I've included the `hyperref` package, both the 2 and the 8 will be hyperlinks to those places in the paper (try it, if you're reading this as a .pdf on your computer).

Similarly, because I added a label `\label{pg:nabla}` right before the point in the paper where I added a footnote on the ∇ symbol, and also added a label `\label{fn:nabla}` inside that footnote, I can write

```
see footnote~\ref{fn:nabla} on page~\pageref{pg:nabla}
```

to have L^AT_EX generate line

```
see footnote 6 on page 17
```

As another example, because I added label `\label{itm:nb}` to the second sub-item of my earlier example list of constraint types, I can now write

```
item~\ref{itm:nb}
```

to refer to item 1b of that list. And finally, because I added labels `\label{eqn:focK}` and `\label{eqn:focL}` to the first-order conditions used earlier to illustrate the `align` environment, I can now write

```
first-order conditions \eqref{eqn:focK} and \eqref{eqn:focL}
```

to refer to first-order conditions (3) and (4).

Using `\eqref` rather than `\ref` in this final example conveniently adds the parentheses that you usually want around equation references; I could also have written `(\ref{eqn:focK})` and `(\ref{eqn:focL})`, adding the parentheses myself.

By the way, the tildes before the `\ref` command in the other examples merely create a so-called “non-breaking space” between for example the word “footnote” and the number 6. This prevents L^AT_EX from inserting an ugly line break between the two.

Also, the `sec:`, `fig:`, `fn:`, etc., strings at the beginning of the label examples have no special meaning. I just like adding them to minimize the chance of accidentally writing something like `footnote~\ref{nabla}` where `\ref{nabla}` refers, say, to a page or an equation rather than to a footnote.

Very importantly, all these internal references are renumbered fully automatically when I add, delete, or reorder any sections, footnotes, equations, etc. (provided you don't delete the labels that they refer to, of course).

As with citations, this does take multiple compilation runs, in this case just two (and just selecting “LaTeX” or “pdfLaTeX” from the pull-down menu—you don’t need to mess with “BibTeX”). The first run creates or updates an auxiliary file with filename extension .aux that lists all the labels L^AT_EX finds in the .tex file. In the .pdf file, it creates numbers at all points where a `\label` command appears in the .tex file (separately incrementing numbers that refer to figures, tables, equations, etc.), but prints double question marks at all points where `\ref` commands appear. It is only during the second run that these double questionmarks get replaced by the appropriate numbers.