1 Description

This package serves as a set of examples of user-defined \LaTeX commands. Everything here is something that I used or experimented with during my time as an undergraduate student. Some of the commands may not be useful at all. The primary purpose of most commands in this package is to reduce the amount of typing required.

2 Writing Your Own Commands

If you’re reading this, you’re presumably an undergraduate student who might be interested in using \LaTeX. In general, I think the best way to start using \LaTeX and then form a habit of it is to start early and to learn things little by little.

That being said, one of the most useful things you can do is to write commands that make things easier. I would recommend accumulating these commands in one file (package) as you go along, similarly to what I’ve done here with alounsburymacros.

Example. In Real Analysis, I found that I was having to type $\mathcal{R}(\alpha)$ over and over again. So, I wrote the command

\[ \text{newcommand}\{\ara}\{\mathcal{R}\{\alpha}\} \]

To avoid this. ‘ar’ is an onomatopoeic rendition of the letter ‘R’ spoken aloud, and the second ‘a’ stands for “alpha.” You have to get creative with naming these things.
3 Some Actual Documentation

Some of this information is similar to the formalities you might see in the documentation of other packages.

I’m not going to take the time to document every command I’ve made in alounsburymacros; however, I’ve made many comments throughout alounsburymacros.sty. If you want to get an idea of what’s included in the package, just open the sty file in your editor read through those comments.

3.1 License

This software is contributed to the public domain. In other words, I don’t care what you do with it. Add things to it, remove things from it, copy things from it to your own set of macros, distribute it, whatever.

3.2 Acknowledgements

Most of the LINEAR ALGEBRA section and perhaps a few commands elsewhere were provided to me in my freshman year by Dr. R. Ablamowicz, emeritus.

3.3 Dependencies

The alounsburymacros package loads forloop, mathrsfs, mathtools, nicefrac, xcolor, amsmath, amsfonts, amssymb, amsthm, cancel, calligra, enumitem, fancyvrb, graphicx, pagecolor, pifont, suppose, totcount, upgreek, verbatim, verse, and fontenc. Most packages will rely on nowhere near as many packages as this. It only loads this many because alounsburymacros is essentially my personal preamble file, so I put whatever I wanted there.

These dependencies may require or include other packages. For instance, the nicefrac package includes the units package. If you require one of these packages to be loaded with some option, consider loading it yourself before the alounsburymacros package or use, e.g.,

\PassOptionsToPackage{tight}{units}

3.4 Option b

Using the alounsburymacros package with the option b:
will use the \texttt{pagecolor} package to set the page color to black and the text color to white.

\textbf{3.5 Commands using mathtools}

The commands \texttt{\textbackslash Mid} and \texttt{\textbackslash set} are respectively renamed from the commands \texttt{\textbackslash given} and \texttt{\textbackslash Set} from page 27 of the \texttt{mathtools} package documentation as follows:

\begin{verbatim}
\providecommand\Mid{}
% can be useful to refer to this outside \set
% just to make sure it exists
\newcommand\SetSymbol[1][1]{\nonscript\:#1\vert\nonscript\:\allowbreak\mathopen{}}
\DeclaPe\Delim\set[1]\{\}
\renewcommand\Mid{\SetSymbol[\delimsize]}%
#1%
\end{verbatim}

The command \texttt{\set} is invoked as \texttt{\set\{\textit{argument}\}}:

\[
2\mathbb{Z} = \{x \in \mathbb{Z} \mid x \mod 2 = 0\},
\]

but adding a * makes everything extensible, in which case we use \texttt{\Mid} rather than \texttt{\mid}:

\[
\mathbb{Q} = \left\{ \frac{x}{y} \mid x, y \in \mathbb{Z} \right\}.
\]

\textbf{3.6 Proof Environments with the Technique Specified}

This package contains proof environments with the technique specified in the title of the proof. I had to create supplementary environments with \texttt{\newtheorem\{\ldots\}\{\ldots\}} in order to make the actual environments that have Q.E.D. symbols with \texttt{\newenvironment\{\ldots\}\{\ldots\}}. In other words, use
rather than
\begin{pfcombinatorial}
\end{pfcombinatorial}

Proof (Combinatorial). The Q.E.D. symbols in these environments function the same way that they do in the regular proof environment.

Proof by Induction. We can even push the Q.E.D. symbol into a display that ends the proof:
\[ \vec{f}_j = \sum_{i=1}^{\infty} \frac{\partial f_i}{\partial x_j} \vec{u}_j. \]

\section{For Whom it May Interest}

There are a couple of things in alounsburymacros that you may never use but may be curious about.

\subsection{Conditionals}

At the end of alounsburymacros.sty there are some experimental commands for typing ditto marks: "\n\n. They are underdeveloped and not very useful, but I’m leaving them as an example of the more dynamic things we can do with \LaTeX{}, such as for-from-to loops, for-each-in loops, while loops, etc.

One applications of conditionals is in drawing with \LaTeX{}. For examples of \texttt{\textbackslash foreach}, refer to this page.

Assuming I haven’t changed it since writing this, I’ve also used the conditional \texttt{\textbackslash ifnum \ldots \textbackslash fi} in my suppose package.

\subsection{\texttt{\textbackslash mspace\{\ldots\}}}  

The command \texttt{\textbackslash mspace\{\ldots\}}\footnote{I believe \texttt{\textbackslash mspace\{\ldots\}} is either short for “mu space” or short for “math space,” but I’m not sure which.} is a more precise way of inserting horizontal space in math mode. It accepts rational values of the unit \texttt{mu}. The conver-
sion is

\[ 18 \, \mu = 1 \, \text{quad} = 1 \, \text{em} = \text{the width of ‘M’ in the current font}. \]

\[
\begin{array}{cc}
  x & y \\
  x & y \\
  xMy \\
\end{array}
\]

\[ 1 \, \text{qquad} \text{is the width of ‘MM’}. \]

\[
\begin{array}{cc}
  x & y \\
  x & y \\
  xMMy \\
\end{array}
\]

Other spaces are defined to be a certain number of \( \mu \).
For instance, the thin space \( \, \), is 3 \( \mu \):

\[
\begin{array}{cc}
  x & y \\
  x & y \\
\end{array}
\]

The thick space \( \; \) is 6 \( \mu \):

\[
\begin{array}{cc}
  x & y \\
  x & y \\
\end{array}
\]

The negative thin space \( \! \) is \(-3 \, \mu\):

\[
\begin{array}{cc}
  xy \\
  xy \\
\end{array}
\]

Another typesetting unit is \( \text{ex} \), or x-height, which is the height of the lower-case ‘x’ character: \( \kappa \).

3.8 \( \text{\textbackslash DeclareMathOperator}\{\ldots\}\{\ldots\} \)
Commands line \texttt{\textbackslash sin, \textbackslash max, \textbackslash sup} are defined as math operators with the command \( \text{\textbackslash DeclareMathOperator}\{\text{\textbackslash commandname}\}\{\langle text\rangle\} \). It displays \( \langle text\rangle \) in
the \texttt{mathrm} font and places a space after \texttt{text} if a delimiter does not immediately follow, which, for instance, is how we write \texttt{sin} by hand when we omit the parentheses.

Compare the following commands that are defined at the beginning of this document:

\begin{center}
\begin{tabular}{c|c}
\verb|\DeclareMathOperator{...}{...}| & \verb|\newcommand{...}{...}| \\
\texttt{sin π} & \texttt{sin\pi}
\end{tabular}
\end{center}