Typesetting Mathematics in LaTeX Workshop #3

David Arnold

Friday, March 19, 1999

– Typeset by FoilTEX –

In-line Versus Displayed Mathematics

Enter the following source code, save as `workshop3.tex`, LaTeX, and preview.

```
\documentclass{article}
\begin{document}
Summation notation, as in $\sum_{k=1}^n 2^k$, looks
slightly different when it occurs within a line of
text (in-line). Contrast this appearance with the
display
    $$
    \sum_{k=1}^n 2^k.
    $$
\end{document}
```

Note that in-line mathematics is delimited with a pair of *single* dollar signs, as in $. . . $. Displayed mathematics is delimited with a pair of *double* dollar signs, as in $$. . . $$.

Note also that the limits appear above and below the summation symbol in displayed mathematics, but alongside the summation symbol within a line of text. This typifies the contrast between displayed and in-line mathematics.

In-line mathematics may also be delimited with `\(` . . . `\)`, displayed mathematics with `\[` . . . `\]`. Adjust the source, `workshop3.tex`, as follows. LaTeX and preview.

```
\documentclass{article}
\begin{document}
Integration, as in \(f(x)=\int_1^x f(t)\,dt\), looks
slightly different when it occurs within a line of
text. Contrast this appearance with the display
    \[
    f(x)=\int_1^x f(t)\,dt.
    \]
\end{document}
```

One can also delimit in-line mathematics with `\begin{math}` . . . `\end{math}` and displayed mathematics with `\begin{displaymath}` . . . `\end{displaymath}`, but this is probably the least favorite technique among TeXnichal typists.

Note that _ and ^ are used for subscripts and superscripts, respectively. If more than one character appears as a subscript or superscript, use grouping symbols to contain them, as in `$x_{i,j}$`. Finally, `\,` generates a thinspace between $f(t)$ and dt.

Crafting New Commands

You can create your own LaTeX commands with \newcommand. For example, if an expression is repeated numerous times in your document, you might think about crafting a command to save typing the expression repeatedly.

```
\documentclass{article}
\newcommand{\be}{\begin{enumerate}}
\newcommand{\ee}{\end{enumerate}}
\begin{document}
I have to complete the following chores today.
\be
\item Mow the lawn.
\item Rake the leaves.
\item Wash the car.
\ee
\end{document}
```

However, if you have a editor with the power of WinEdt, this is not the optimal solution. Select Insert→Environment→Enumerate from the WinEdt menu bar and watch what happens. Fill in the first item, then use Ctrl+Spacebar to jump to the next item.

Commands With Parameters

This command \rowvec defined below saves keystrokes.

```
\documentclass{article}
\newcommand{\rowvec}{(x_1,x_2,\ldots,x_n)}
\begin{document}
Consider the vector $\rowvec$.
\end{document}
```

But what if you want to use the variable α for the components of your vector? Try this instead.

```
\documentclass{article}
\newcommand{\rowvec}[1]{({#1}_1,{#1}_2,\ldots,{#1}_n)}
\begin{document}
Consider the vectors $\rowvec{\alpha}$,
$\rowvec{\beta}$, and $\rowvec{\gamma}$.
\end{document}
```

The [1] in \newcommand{\rowvec}[1] signals LaTeX that the command \rowvec expects one parameter. Thus, the command $\rowvec{\alpha}$ grabs the parameter \alpha and inserts it in place of #1 in {({#1}_1,{#1}_2,\ldots,{#1}_n)}.

Pushing this example further, what if you don't want the last component to be the nth component of your vector?

```
\documentclass{article}
\newcommand{\rowvec}[2]{({#1}_1,{#1}_2,\ldots,{#1}_{#2})}
\begin{document}
Consider the vectors $\rowvec{\alpha}{2k+1}$
and $\rowvec{\beta}{s}$.
\end{document}
```

The [2] in \newcommand{\rowvec}[2] signals LaTeX that the command \rowvec expects two parameters. Thus, the command $\rowvec{\alpha}{2k+1}$ grabs the parameters \alpha and 2k+1 and inserts them in place of #1 and #2, respectively, in {({#1}_1,{#1}_2,\ldots,{#1}_{#2})}.

Changing Type Styles

There are three font *families*. The commands `\textrm`, `\textsf`, and `\texttt` invoke the Roman, Sans Serif, and `Typewriter` families, respectively, with Roman being the default. Save the following as `workshop3.tex`, LaTeX, and preview.

```
\documentclass{article}
\begin{document}
Here is a sample of the \textsf{Sans Serif} family,
and here is a sample of the \texttt{Typewriter}
family.
\end{document}
```

There are two font *series*. The commands `\textmd` and `\textbf` invoke the Medium and **Boldface** series, respectively, with Medium being the default series.

```
\documentclass{article}
\begin{document}
This is \textbf{bold}, while \textbf{this is
\textmd{medium} series}.
\end{document}
```

There are four font *shapes*. The commands \textup, \textit, \textsl, and \textsc invoke the Upright, *Italic*, *Slant*, and SMALL CAPS shapes, respectively, with Upright begin the default shape.

```
\documentclass{article}

\begin{document}

At 3\,\textsc{pm}, you can easily tell the difference
between \textit{the italic font shape} and
\textsl{the slant font shape}. \textit{If the text
is already italic, emphasis is provided by \textup{the
upright shape} in the italicized sentence}.

You can also combine various text commands, as
\textsl{in \textbf{this} phrase}.

In most circumstances, if you want to emphasize
\emph{a word or phase}, use the \verb+\emph+ command.

\end{document}
```

Note: In WinEdt, place the cursor in the command \verb above, then hit Ctrl+F1 to get help on the \verb command.

Logical Markup

Suppose that you are writing a biology paper, and you want to the names of genera to appear in slant shape throughout your paper. Don't use `\textsl` throughout the text. Use the following approach instead.

```
\documentclass{article}
\newcommand{\genus}[1]{\textsl{#1}}
\begin{document}
\genus{Connochaetes} seems to pop up \ldots
\end{document}
```

Later, should your editor or publisher request that you change the genus to italic shape, you only need to change one command, not the hundreds of occurrences of genera in slant shaped font.

This example clearly marks the advantage of a logical markup language like LaTeX over WYSIWYG (what you see is what you get) wordprocessors. And *please* don't even try to rebut this argument with a reference to global search and replace, a **very dangerous** operation to perform on a 500 page manuscript.

Roots and Fractions

Save the following as `workshop3.tex`, LaTeX, and preview. The output is self-explanatory.

```
\documentclass{article}
\begin{document}
It's probably best to use in-line notation for
fractions appearing in a line of text, as in
$x/(x+3/x)$. A fraction such as $\frac{x}{x+\frac3x}$
makes a mess of interline spacing and is hard to
read, leading to a document that does not quite
achieve the professional look you are striving for
when using \LaTeX. However, the same fraction, when
displayed as
    $$
    \frac{x}{x+\frac3x},
    $$
looks great! So go for it! These types of fraction
decisions come up all the time. For example, which
of the following look better? Which is easier to
read?
    $$
    x^\frac12 \quad\mbox{or}\quad x^{1/2}
    $$
Finally, displaying roots is easy in \LaTeX.
    $$
    \sqrt{\frac{x}{x+\frac3x}}
    \quad\mbox{or}\quad
    \sqrt[3]{\frac{x}{x+\frac3x}}
    $$
\end{document}
```

Delimiters and Arrays

LaTeX offers a wide variety of delimiters.

```
\documentclass{article}
\begin{document}
This leads to the property
    $$
    \|c \vec v\|=|c|\|\vec v\|.
    $$
\end{document}
```

Delimiters will automatically expand to the correct height in mathematical expressions if you proceed them with the commands `\left` and `\right`.

```
\documentclass{article}
\begin{document}
This leads to the result
    $$
    \left\{ \frac{x}{x-1} \right\}^{1/p}.
    $$
\end{document}
```

However, you must *pair* each `\left` with a `\right` or you will get an error when you LaTeX your document.

Array syntax is similar to the `tabular` environment. Separate row entries with ampersands (&), and end each row (except the last) with a newline (\\).

```
\documentclass{article}
\begin{document}
This leads to the result
    $$
    \left(
        \begin{array}{cc}
        a_{11} & a_{12}\\
        a_{21} & a_{22}
        \end{array}
    \right)
    \left(
        \begin{array}{c}
        x_1\\
        x_2
        \end{array}
    \right)
    =\left(
        \begin{array}{c}
        b_1\\
        b_2
        \end{array}
    \right).
    $$
\end{document}
```

Equations

You can easily typeset automatically numbered equations in LaTeX.

```
\documentclass{article}
\begin{document}
All things considered, our previous discussion
leads to the following result.
\begin{equation}
s=\int_0^t \sqrt{\|\vec v(\tau)\|}\,d\tau.
\end{equation}
\end{document}
```

You can also typeset automatically numbered arrays of equations.

```
\documentclass{article}
\begin{document}
All things considered, our previous discussion
leads to the following result.
\begin{eqnarray}
s&=&\int_0^t \sqrt{\|\vec v(\tau)\|}\,d\tau\\
s&=&\int_0^t \sqrt{\left(\frac{dx}{d\tau}\right)^2
    +\left(\frac{dy}{d\tau}\right)^2}\,d\tau
\end{eqnarray}
\end{document}
```

You can suppress all numbering of equations with \eqnarray*.

```
\documentclass{article}
\begin{document}
All things considered, our previous discussion
leads to the following result.
\begin{eqnarray*}
s&=&\int_0^t \sqrt{\|\vec v(\tau)\|}\,d\tau\\
s&=&\int_0^t \sqrt{\left(\frac{dx}{d\tau}\right)^2
    +\left(\frac{dy}{d\tau}\right)^2}\,d\tau
\end{eqnarray*}
\end{document}
```

You can suppress the numbering of any individual equation with \eqnarray and \nonumber.

```
\documentclass{article}
\begin{document}
All things considered, our previous discussion
leads to the following result.
\begin{eqnarray}
s&=&\int_0^t \sqrt{\|\vec v(\tau)\|}\,d\tau\nonumber \\
s&=&\int_0^t \sqrt{\left(\frac{dx}{d\tau}\right)^2
    +\left(\frac{dy}{d\tau}\right)^2}\,d\tau
\end{eqnarray}
\end{document}
```

Function Names

Most of the important functions in mathematics are entered in a special way in LaTeX.

```
\documentclass{article}
\begin{document}
Note that
    $$
    sinx
    $$
doesn't look right. The spacing is all wrong. Note
that
    $$
    sin\,x,
    $$
although somewhat improved, still doesn't look right.
However,
    $$
    \sin x
    $$
has the proper spacing and sets the math name in an
upright roman font. Just what the math doctor ordered!
\end{document}
```

In a similar manner, LaTeX provides `\log`, `\sinh`, etc. In WinEdt, right-click the menu bar, then select Show GUI Page Control. Click the `\ln(x)` tab to get a complete listing of function names recognized by LaTeX.