Math I: Super/Subscripts and Common Commands

Introduction

This tutorial will show you how to do some basic typesetting of math symbols, equations and matrices. Although LaTeX in its basic form (i.e. when you start to write it in WinEdt) can do almost everything in math that the normal user will want to do, we can make it more **user-friendly** by telling it to load certain *packages* into its operating memory.

- open WinEdt and write the following:

  ```
  \documentclass{article}
  \usepackage{amsmath, amssymb}
  \begin{document}

  \end{document}
  ```

  Note that this is just like last time, except that we have added

  `\usepackage{amsmath, amssymb}`

  in our *preamble*. These packages gives LaTeX access to some more symbols and some more structures that we will find useful and will make our lives a bit easier today. You don't have to load them every-time you want to write math, but some of our commands and symbols today (or in the second math typesetting session)—specifically those connected to matrices—will work *only* if you do. Note that `amsmath` and `amssymb` are different packages that need to be separated by a comma.

Reminder: In Line and Display

- when you want some math in the same line as your text, you use the dollar signs. So, `$4x-6y=67$` will produce the expression in the text, as $4x - 6y = 67$. Here is another sentence to convince you it was genuinely *in text*.
- when you want some math displayed on its own below your text, you use `\[ \]`. So, `\[x+37=56\]` gives

$$x + 37 = 56.$$

- everything I say below about writing math in dollar signs will apply to the math display mode, `\[  \]`, too. Sometimes it will look slightly different, because LaTeX adjusts heights of characters and their spacing to look more reasonable given it is in text. For example, $\sum_{i=1}^{n} f(x)$ looks different to

$$\sum_{i=1}^{n} f(x),$$

but they were produced using the same commands (but in-line and displayed, respectively)

- note that numbers like 1, 2, 67, 3459 are always in math mode. There seems to be a UK/US difference when writing large numbers with commas. Americans often don't leave a space between the comma and the next number, but many other nationalities do. So, Brits write `$100,000$` to write $100,000$, whereas Americans write `100,000` (without the `$ $` signs) to write 100,000.
- note that letters in math mode look very different compared to those in text. So `$x$` gives x, while in text, we would have x
- Be careful with punctuation in math mode. Suppose you wanted to write

  $$\frac{\partial y}{\partial x}=\frac{1}{|x|},$$

  then you need to be sure that you put the comma (or semi-colon, or whatever) in the right place. This displayed formula was produced with the following code:

  ```
  \[\frac{\partial y}{\partial x}=\frac{1}{|x|},\]
  ```

  Note that the comma that comes at the end of the formula occurs before the command `\]`. If we put the comma *after* the `\]` command it would appear on the next line (incorrectly). That said, when we write *in line* math —recall with dollar signs— make sure you put the comma *after* the closing dollar sign.

Greek Letters

In math mode, you sometimes need Greek letters, like γ, ϵ, ϕ and so on.

- generally, the command structure is `$\letter-name$`. So, to get a theta you type `$\theta$`. That gives you θ.
- note that I used a *lower-case* t at the start of theta to get a lower case theta in math mode. To get an upper case one, you use a capital T. So, `$\Theta$` gives Θ

- this upper/lower case distinction is not the case for all Greek letters: for example, there is only one type of alpha and beta LaTeX will produce (and they both take the lower case): `$\alpha$` gives you α and `$\beta$` gives you β, whilst `$\Beta$` will give you an 'undefined command' message

- if you can't remember all the names of the letters, WinEdt can help you out. Look at the top of your tool bar, and you will see a sum button $\sum$ (this is actually a capital epsilon, but is produced using `$\sum$`. To the left of it is a picture of a video camera and to the right is an umlauted E. Pressing the $\sum$ button gives you all the math and Greek characters you will need (and a lot of other useful symbols beside). Pick one, and press its button. Notice that it just writes the command for that symbol: **you** need to put it in dollar signs otherwise LaTeX won't understand what you want to do.

- when you write a greek letter command (or any other math command), you **must** leave a space between the end of the command and the next nonoperation—like $+, -, \div, \times$ —character. So, it won't know what to do with `$\alpha5$`, but if you give it `$\alpha 5$` it will write $\alpha 5$ (note how it deletes the space you left between the characters).

Subscripts and Superscripts

- to subscript an expression, like P_i you use an underscore in the following way: `$P_i$`. If there an expression of *several* symbols that are to be a subscript, you need to use braces. If you don't use an opening brace after the underscore, only the *first* character will be subscripted. So, `$R_i\beta$` gives you $R_i\beta$, but `$R_{i\beta}$` gives you $R_{i\beta}$.

- to superscript an expression, like F^x you use the hat symbol in the following way: `$F^x$`. If there an expression of *several* symbols that are to be a superscript, you need to use braces. If you don't use an opening brace after the hat, only the *first* character will be superscripted. So, `$G^2\alpha$` gives you $G^2\alpha$, but `$G^{2\alpha}$` gives you $G^{2\alpha}$.

- sometimes symbols have a subscript *and* a superscript. Here is a John Duggan favorite: x_i^m which was produced by writing `$x_i^m$`: you could also use `$x^m_i$` since LaTeX understands either order of superscripts and subscripts. What was said about braces (above) still applies.

- subscripts and superscripts can themselves have subscripts and superscripts. To get e^{k_i}, you use `$e^{k_i}$`. Note that the `i` used for the subscript of the `k` is smaller than the e and the k.

- for things like integrations and sums, we use exactly the same principles. To show integration between x and ∞ for example, we type `$\int^\infty_x$`

which yields $\int_x^\infty$, or

$$\int_x^\infty$$

in display mode.

Summing between $i = 1$ and n over x_i is written as `$\sum^{n}_{1} x_i$` which yields $\sum_{i=1}^n x_i$, or

$$\sum_{i=1}^n x_i$$

in display mode.

Inequalites, equalities, set notation etc.

LaTeX has lots of symbols available for use. You can find many of these if you press the $\sum$ key on the tool-bar as we discussed above. We will not list all the possible symbols (you can find them in any LaTeX guide book), but some of the common ones are in the table below (once again, all of this must be in the math mode). Some of the are commands are intuitive, some aren't. You (may) need to use the packages we specified above to have access to of all of them.

Notice that some of them allow you to write subscripts/superscripts and limits in the way we discussed before. For example, you could write

```
\[\bigcap^\infty_{i=1}(i-1,i)=\emptyset\]
```

if the mood so took you. This gives:

$$\bigcap_{i=1}^\infty (i-1,i) = \emptyset.$$

(If that is correct, credit to me. If it is wrong, blame Tuḡba Güvenç. She never did give us solution sets. . .)

output	input	output	input
$=$	`=`	$\ne$	`\ne`
$<$	`<`	$>$	`>`
$\leq$	`\leq`	$\geq$	`\geq`
$\in$	`\in`	$\ni$	`\ni`
$\notin$	`\notin`	$\forall$	`\forall`
$\exists$	`\exists`	$\nexists$	`\nexists`
$\emptyset$	`\emptyset`	$\mathbb{R}$	`\mathbb{R}`
$\prime$	`\prime`	$\blacksquare$	`\blacksquare`
∂	`\partial`	$\div$	`\div`
$\cup$	`\cup`	$\cap$	`\cap`
$\bigcup$	`\bigcup`	$\bigcap$	`\bigcap`
$\subset$	`\subset`	$\supset$	`\supset`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleftrightarrow`
$\Leftarrow$	`\Leftarrow`	$\Rightarrow$	`\Rightarrow`
$\times$	`\times`	$\approx$	`\approx`
$x\|y$	`x|y`	$\therefore$	`\therefore`
∞	`\infty`	$\prod$	`\prod`
$\vee$	`\vee`	$\land$	`\land`

Words: lim, det, sin. . .

In math you often use small words like cos to mean cosine, max to mean maximum and so on. If you just write the words in math mode, it won't look correct. For example, if we write `$sin\theta$` we get $sin\theta$, which is not what we want. If we write `$\sin\theta$`, though, it reads as: $\sin\theta$. The commands for these operators are generally intuitive, and we give a brief list of some common ones here.

output	input	output	input
$\det$	`\det`	$\inf$	`\inf`
$\lim$	`\lim`	$\liminf$	`\liminf`
$\limsup$	`\limsup`	$\sup$	`\sup`
$\max$	`\max`	$\min$	`\min`
$\Pr$	`\Pr`	$\arg$	`\arg`
$\cos$	`\cos`	$\exp$	`\exp`
$\sin$	`\sin`	$\cos$	`\cos`
$\tan$	`\tan`	$\deg$	`\deg`
$\varinjlim$	`\varinjlim`	$\varprojlim$	`\varprojlim`

Brackets

Examples include parentheses like (a, b) which was produced using `(a,b)` or curly braces like $\{a, b\}$ produced using `\{a,b\}`.

- if you type the commands above, they give you only a certain (regulated) size of delimiter. But that doesn't look correct in some circumstances. For example:

  $$(\sum_{i=1}^n),$$

  produced by `\[(\sum_{i=1}^n)\]` looks strange. You can correct for this by using variable size delimiters which have the general format:

  `\left`*open delimiter*   *math bit*   `\right`*close delimiter.*

  So, the following code `\[\left(\sum_{i=1}^n\right),\]` gives you

  $$\left(\sum_{i=1}^n\right).$$

  You could substitute `(` and `)` with `[` and `]` or `{` and `}` or `\langle` and `\rangle` as desired.

Accents

Suppose you want $\hat{h}$: you will use `$\hat{h}$`. Notice that you have to be **in math mode** for this to work. For example, `\hat{h}` without the dollar signs won't be understood by LaTeX . Here is a useful list for guidance (from Diller, p102):

output	input	output	input
$\acute{a}$	`\acute{a}`	$\dot{f}$	`\dot{f}`
$\bar{b}$	`\bar{b}`	$\grave{g}$	`\grave{g}`
$\breve{c}$	`\breve{c}`	$\hat{h}$	`\hat{h}`
$\check{d}$	`\check{d}`	$\tilde{\imath}$	`\tilde{\imath}`
$\ddot{e}$	`\ddot{e}`	$\vec{\jmath}$	`\vec{jmath}`

If you want to put any of the accents over *more* than a single symbol LaTeX will just center the accent over the whole expression. However, for the `\hat` and `\tilde` command there are 'wide versions': so, `$\widehat{npq}$` gives $\widehat{npq}$, and `$\widetilde{mvp}$` gives $\widetilde{mvp}$.