LaTeX Workshop 2, Spring 2021

1 Labels and references

We have seen that many elements of a LaTeX document are automatically numbered, including sections, pages, etc. While we have thus far been using unnumbered equations with `\[ ... \]` or `$$ ... $$`, you can also typeset numbered equations with the construct

```
\begin{equation}
a = b
\end{equation}
```

Note: unnumbered sections, equations and so on are typically defined using a starred version of the corresponding macro, as in `\section*{Unnumbered section}` and `\begin{equation*} ... \end{equation*}`, the latter being equivalent to `\[ ... \]`.

Any of these automatically numbered elements may be given a label via the `\label{}` macro, as in

```
\begin{equation}
\label{E:myequation}
a = b
\end{equation}
```

or

```
\section{Introduction} \label{S:myintro}
```

Each label should be a unique name that you have not previously used for another label, or LaTeX will complain. Note that the `E:` and `S:` are not strictly necessary, you can use any name you want such as `\label{mywhatever}` for sections, equations, theorems, and so on, but it is good practice to separate

the namespaces for your equations from your sections and so on, so that you could have, for example a section labeled `S:pythagoras` and an equation labeled `E:pythagoras` without these conflicting.

To reference your numbered environment, use the `\ref{}` macro, as in

```
As discussed in section \ref{S:myintro} above,
the equation (\ref{E:myequation}) leads to the following formula.
```

When referencing equations, the macro `\eqref{}` (provided by the `amsmath` package) is preferred as it automatically puts the parentheses in for you, so the formulation

```
As discussed in section \ref{S:myintro} above,
the equation \eqref{E:myequation} leads to the following formula.
```

is equivalent to the above, but less error prone.

Problem 1. Make a simple document with several sections and numbered equations with labels. Then add some text which references some of your elements. Note what happens when you add additional sections or equations, causing the numbering to change.

2 Theorem environments

To typeset theorems, lemmas, corollaries, etc., the `amsthm`[1] package provides a way to define associated environments. Putting the line

```
\newtheorem{thm}{Theorem}
```

in your preamble (meaning the content before the `\begin{document}` line) defines a new environment called `thm` which can be used as in the following:

```
\begin{thm}
\label{T:pythagorean}
Let $a$ and $b$ be the lengths of the orthogonal sides of a
right triangle with hypotenuse of length $c$. Then
\[
a^2 + b^2 = c^2.
\]
\end{thm}
```

[1] the `amsmath` and `amsthm` packages should basically always be used in your documents.

This produces something like

Theorem 1. *Let a and b be the lengths of the orthogonal sides of a right triangle with hypotenuse of length c. Then*

$$a^2 + b^2 = c^2.$$

The theorem is automatically numbered, and we gave it a label as above, so it can now be referenced as in

```
According to Theorem \ref{T:pythagoras}, ...
```

Additional environments can be created this way, for lemmas, corollaries, definitions, etc. Proofs are defined using the (already defined) `proof` environment, as in

```
\begin{proof}
The proof follows from the law of cosines.
\end{proof}
```

Problem 2. Add environments for theorems, lemmas and definitions to your document and create at least one of each. Add a proof for one or two of your theorems or lemmas. (The content does not have to be sensible mathematics!) In the command `\newtheorem{thm}{Theorem}`, determine what the second argument (the `Theorem`) does.

3 Theorem appearance and numbering

By default, theorem-like environments are typeset with a bold heading and emphasized (italicized) text. This can be changed with `\theoremstyle{}` as in

```
\theoremstyle{definition}
\newtheorem{thm2}{Theorem}
```

Problem 3. Try this and see what a `thm2` environment looks like.

There are three settings:

- `\theoremstyle{plain}` gives the default style

- `\theoremstyle{definition}` gives bold headings but non-emphasized text, and
- `\theoremstyle{remark}` gives italicized headings and regular text.

As the name suggests, these are usually mixed and matched, as in

```
\theoremstyle{plain}
\newtheorem{thm}{Theorem}
\newtheorem{cor}{Corollary}
\newtheorem{lem}{Lemma}
\theoremstyle{definition}
\newtheorem{def}{Definition}
\theoremstyle{remark}
\newtheorem*{rmk}{Remark}
```

Note that in the last line, `\newtheorem*{rmk}{Remark}` the use of the asterisk defines an environment which will not be numbered.

Note that instances of each type of environment will be numbered consecutively according to that environment, so we get for example, Theorem 1, Theorem 2, Lemma 1, Theorem 3, etc. You can make a theorem environment share numbering with another one via an optional argument as follows:

```
\theoremstyle{theorem}
\newtheorem{thm}{Theorem}
\newtheorem{lem}[thm]{Lemma}
```

Then you'll get Theorem 1, Theorem 2, Lemma 3, Theorem 4, etc. You can also make the numbering of theorem like environments subordinate to sections (or subsections, etc) as follows:

```
\theoremstyle{theorem}
\newtheorem{thm}{Theorem}[section]
\newtheorem{lem}[thm]{Lemma}
```

This will make the `thm` environment (and all environments sharing its nubmering, like `lem`) numbered within the section, so for example, in section 1 we have Theorem 1.1, Lemma 1.2, Theorem 1.3, etc. and then in section 2 we have Theorem 2.1, Lemma 2.2, etc.

Problem 4. Alter your document to make the numbered environments share a common numbering system which is subordinate to the section numbering.

In case you want to make your equation numbers subordinate to your sections also, this is done slightly differently, by putting

```
\numberwithin{equation}{section}
```

in your preamble.

4 Multi-line equations

Often one wants to break a single equation across multiple lines, or collect several equations together in one set out display. The `amsmath` package provides several useful ways to do this.

The simplest is the `gather` environment, which allows you to put line breaks in your equation via `\\` and then centers the results. For example,

```
\begin{gather}
a = b \\
\sin\big(\omega (t - s)\big) = \cos\big(\omega_0(x - y)\big)
\end{gather}
```

gives

$$a = b \tag{1}$$
$$\sin\big(\omega(t-s)\big) = \cos\big(\omega_0(x-y)\big) \tag{2}$$

Notice that each line has its own number, and we can assign labels to each:

```
\begin{gather}
\label{E:partone}
a = b  \\
\label{E:parttwo}
\sin\big(\omega (t - s)\big) = \cos\big(\omega_0(x - y)\big)
\end{gather}
```

and reference them with `\eqref{E:partone}`, `\eqref{E:parttwo}`, etc.

Next is the `multline` environment, which is similar, except the first equation is justified left and the last equation is justified right, for example:

```
\begin{multline}
f = \text{some very very very long expression} \\
\times \text{even more long stuff in the product} \\
+ \text{another very long term}
\end{multline}
```

gives

$$\begin{aligned} f = &\text{some very very very long expression} \\ &\times \text{even more long stuff in the product} \\ &+ \text{another very long term} \end{aligned} \tag{3}$$

This type of display, which is useful for breaking up a single very long equation, has just one equation number.

Finally, for much more control, there is the `align` environment, which aligns each line in its display so that the points associated to the `&` symbol (which is not printed) are lined up:

```
\begin{align}
f(x) &= \int_0^x 2g(y) g'(y)\,dy \\
 &= \int_0^x (g^2(y))'\,dy \\
 &= g^2(x) - g^2(0)
\end{align}
```

gives

$$\begin{aligned} f(x) &= \int_0^x 2g(y)g'(y)\,dy && (4) \\ &= \int_0^x (g^2(y))'\,dy && (5) \\ &= g^2(x) - g^2(0) && (6) \end{aligned}$$

Multiple alignment points can be specified per line with additional ampersands; see the documentation or Google for examples.

Each of these environments is available in an unnumbered version by using its starred form `gather*`, `multline*`, and `align*`.

Problem 5. Try typesetting the following displays.

$$f : \{x \in \mathbb{R} : 1 < x < 2\} \to \mathbb{R},$$
$$x \mapsto 4x^3 + 2x - 3$$

$$\frac{a - 2b + c + 1}{d} + \frac{x^3 + x^2 + -x + 1}{y} = \frac{ay - 2by + cy + y + dx^3 + dx^2 - dx + d}{dy}$$

$$\begin{aligned} u_{xx} + u_{yy} &= 0 && \text{for } x^2 + y^2 < a^2 \\ u &= h(\theta) && \text{for } x^2 + y^2 = a^2 \end{aligned}$$

5 Math operators

Often we make use of mathematical operators whose names are words of several letters. There are already many mathematical operators defined by LaTeX, such as `\sin`, `\cos`, `\tan`, `\lim` and others, but sometimes you need a new one. Say we need a new operator called "op", which will act on variables. If we just write `$op(x)$` we get $op(x)$, which is not waht we want. Using `\text` or `\mathrm` is better: `$\mathrm{op}(x)$` looks right, giving $\mathrm{op}(x)$. However, the preferred way is to use `\operatorname`, as in

```
\operatorname{op}(x)
```

which gives

$$\operatorname{op}(x)$$

and also looks right.

If the two look the same, why is `operatorname` preferred? It gives LaTeX more guidance on how to employ spacing; compare for example

$$a\mathrm{op}x + b\mathrm{op}y \quad \text{vs} \quad a \operatorname{op} x + b \operatorname{op} y$$

which is the result of typesetting

```
a \mathrm{op} x + b \mathrm{op} y
\quad \text{vs} \quad
a \operatorname{op} x + b \operatorname{op} y
```

By default, super and sub scripts for an operator are typeset appear like this: op_a^b. To make subscripts appear below the operator (as in $\lim_{x\to 0}$) use the starred version `\operatorname*`. Thus we can write

```
\[
\operatorname*{Max}_{x \in A} f(x)
\]
```

to get

$$\operatorname*{Max}_{x\in A} f(x)$$

Since the `\operatorname` macro is long and cumbersome to type, you would usually want to define macros for your operators, for example by putting

```
\newcommand\myop{\operatorname{op}}
```

in your preamble. Note that the `amsmath` package defines a special macro `\DeclareMathOperator` which does the same thing:

```
\DeclareMathOperator{\myop}{op}
```

is equivalent to

```
\newcommand\myop{\operatorname{op}}
```

Likewise,

```
\DeclareMathOperator*{\Max}{Max}
```

is equivalent to

```
\newcommand\Max{\operatorname*{Max}}
```

Problem 6.

1. Define operators for kernel and cokernel and typeset the following linear algebra identity: if $A : V \to W$ is a linear transformation, then

   $$\dim \operatorname{Ker} A - \dim \operatorname{Coker} A = \dim V - \dim W$$

   Note `\dim` is already defined.

2. Use an operator definition to typeset

$$\operatorname*{Mat}_{n\times n}(\mathbb{R})$$

for a set of $n \times n$ real matrices.