We are organising an informal **workshop** on the **intermediate and advanced topics** in LaTeX. Although there are various LaTeXtutorials and templates floating around, but they often omit some tools and packages that are useful in meteorology and geophysics. The workshop is primary focused on PhD students who are starting to write their thesis, but it is open to other LaTeXusers as well.

1. Monday 19. June — from 12:45 in CIP room
2. Wednesday 21. June — from 15:00 in CIP room

You can work from CIP workstation or bring your own device.

target audience:	people with previous experience with LaTeX
aims:	discuss LaTeX topics practise skills
duration:	one and half hour from start time or until you start getting tired
topics:	see following chapters 1, 2, and 3
registration:	comment in this Slack thread

Advanced LaTeX Workshop

Your Name Here

19 June 2023

Chapter 1

Intro

Shall we begin?

1.1 Combining Document from Pieces

Combining documents from multiple files speeds up the editing process, makes collaboration easier, and also lowers the risk of accidentally rewriting something. You can see an example how most of the header of this document is in a separate file. To try this, write some dummy text in a separate file and insert is here using the `input` command:

The `input` statement can also work on multiple levels

1. Make a copy of `style1headerfooter.tex` and modify it.
2. Open `in1header.tex` and replace `style1headerfooter.tex` with the name of your new file.
3. Recompile the main document.

1.1.1 Automatically Generated Lists

There is an easy way how to create list of figures, tables, as well as index of phrases.

Overview of Pieces

List of Tables

List of Figures

Listings

List of Symbols

notation	unit	meaning
$\sim$	$\cdot$	similar - assignment of probability distribution
$\propto$	$\cdot$	proportional equivalent to; i.e. equivalent up to a constant
$\overline{(\,\cdot\,)}$	$\cdot$	horizontal averaging
$\overline{\varphi}$	$\cdot$	mean value of a quantity φ

List of Abbreviations

notation	meaning
AWS	automatic weather station
ABL	atmospheric boundary layer
CAO	cold–air outbreak
CBL	convective boundary layer
SBL	stable boundary layer
Sc	stratocumulus
SH	sensible heat
TKE	turbulent kinetic energy

Chapter 2

Monday

2.1 Modifying Plots and Schematics

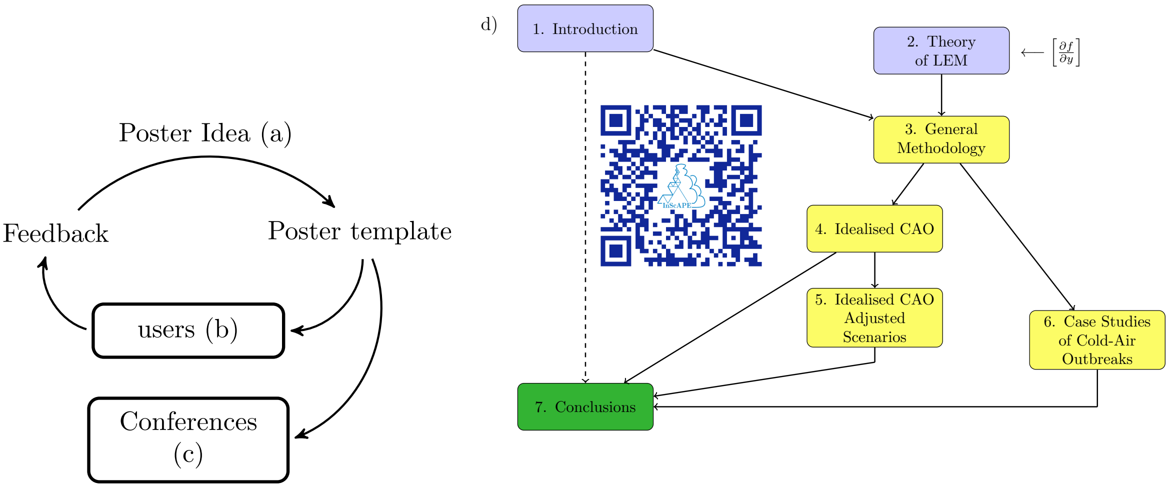

Figure 2.1: Here we combine a **Tikz** diagrams and images. We can also compile the figure as separate **standalone** pdf and then include it as a graphical element.

2.2 Counters

How did we suddenly jump from section 2.1 to 2.2 ?

2.3 Customizing Links, References, and Citations

Such as modifying the style of links to other parts of the same document (Automatically Generated Lists) and links to external websites.

2.4 Version Control and Comparison

We also look at the external tools such as `latexdiff` that compares two LaTeXfiles, and `pdfdiff` that compares . . . you know what.

Chapter 3

Wednesday

We will start with: If you see this text in black, it means that the difference between files is not highlighted.

3.1 Guidelines

To speed up today the progress of the workshop today, the new commands are not only listed in the text but also included as comments. To make the commands actives, just remove or add comment signs.

3.2 External tools

First we will have a quick look at the *latexdiff* tool from the last time. But this time, we will show its shortomings. First, look at the beginning of today chapter and swap which lines are commented between `sillytext1.txt` and `sillytext2.txt`
Secondly, open your shell, change to the folder with this file and type (example a):

```
latexdiff day2latex.tex day2latex_v2.tex > different.tex
```

and then compile the output file. Does it highlight any differences?

Fortunatelly there is a way to show the differences in the included files. Use the option `--flatten` which merges all the `input` files before comparing (example c):

```
latexdiff --flatten  day2latex.tex day2latex_v2.tex > differentshow.tex
```

We can also play with the style of the latexdiff file.

And with the option `--type`, we can also define how the changes appear. The two most useful choices are:

- UNDERLINE — the added text is not only blue, but also underlined
- PDFCOMMENT — changes are not marked in the text, but as PDF comments!

Go ahead and give it a try (example d):

```
latexdiff --flatten --type=PDFCOMMENT day2latex.tex day2latex_v2.tex > differentcom.tex
```

3.2.1 Other tools

There are a couple of other tools that can speed up your work with latex. Unfortunately they are not installed here. But you can try them later:

- *pdfgrep* — similar to grep, searches inside a PDF document that was already compiled; See: tutorial, man page
- *pdftk* — tool for splitting, combing and mixing pdf documents; See: pdft examples

3.3 Fillers

3.3.1 Filler text

Sometimes we need to test whether the structure of the document and the formatting works. Instead of spending time copying and pasting random stuff, we use the *Lore Ipsum* generator. In the header we already inserted `lipsum` package, and here we insert the following command (example e):

```
\lipsum[0-2]     %generates dummy text
```

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

3.4 Code Listing with Highligths

You have already seen example of code listing in the previous paragprahs. Now we will look at this topic in more detail. With teh package **listings**, you can show the structured code and highlight parts of it. The location of the code could be:

- Directly copy and paste to the latex document 3.1.
- You point to the source code file.
    - ○ You can event point to the source code of this very file (see 3.2).

```
lname = "LaTeX"     # strings brown, comments in dark green
print("Do you sometimes use {} to generate {} files?".format("Python",lname))
```

Listing 3.1: Copy and paste code (example a)

```

\setcounter{lstlisting}{\value{samplecode}}
\refstepcounter{samplecode}
\begin{lstlisting}[language={python},caption={Copy and paste code (example \alph{samplecode})},label={code:python},
    frame=single, captionpos=b, basicstyle=\footnotesize\color{cyan}, showstringspaces=false,
    keywordstyle=\color{magenta}, stringstyle=\color{brown}, commentstyle=\color{ForestGreen}
```

```
     ]
 lname = "LaTeX"      # strings brown, comments in dark green
 print("Do you sometimes use {} to generate {} files?".format("Python",lname))
\end{lstlisting}

\refstepcounter{samplecode}
\lstinputlisting[ language={[latex]tex},       % tex syntax, latex option
  caption={Self-showing code (example \Alph{samplecode})}, label={code:thislatex},
  firstline=368, firstnumber=368, lastline=387, numbers=left , captionpos=b,
  breaklines=true , basicstyle=\scriptsize\color{darkgray}, frame=single ,
  keywordstyle=\bf\color{magenta}, commentstyle=\color{ForestGreen}  % comments in different colour
]{./day2latex_backup.tex}    % code location

\noindent You can try it yourself:
```

Listing 3.2: Self-showing code (example B)

You can try it yourself:

1. Change colour which highlights keywords.
2. Change the comments fonts to italics (hint: `\it`)
3. Adjust the path to the code so it is listing part of this file that your are working on.

3.4.1 Callback to Monday

1. Make the symbol after the word “example” in 3.2 is consitent with the rest of examples.
2. Instead of the *listing* counter is starts with the value of the *example* counter, make it that the *example* counter being (re)set to the value of the *listing* counter.
3. Recompile and look how it appears on the page vii

Further Reading

- Overleaf Guidelines on code listings.
- NASA code listing examples
- Documentation of Listing Package

3.5 Tables

Sometimes we directly write a table in the latex document. But sometimes we need values from and external file. Shall we just copy and paste them?

Sure, if you want to risk messing up the whole document and losing the newest values.

A better solution is to export the table in a separate file and just input here, such as:

```
 \input{/somefolder/exported_table.tex}
```

But there are even some more comfortable solutions.

3.5.1 Importing Tables

Instead of copy & paste values into tables, we just read them from an external csv file and format them.

```

\begin{table}[h!] \label{tab:import}
  \begin{center}
  \csvreader[
      tabular = |r|r|r|,
      table head  = \hline
          \( \quad  z  \; \lbrack \mathrm{m} \rbrack  \)
          &  \(  \; \bar{u} \;  \lbrack \mathrm{m\, s}^{-1} \rbrack \)
          & \(  \quad \quad \Big. \bar{ \theta }  \; \lbrack \mathrm{K} \rbrack \Big.
               \)
          \\
          \hline ,
      table foot = \hline
          & \multicolumn{2}{ c|}{\( \ldots \) }
          \\ \hline
  ]{values.csv}  % table location
  {  z=\zvec, u=\uvec, theta=\thvec    % mapping of columns
  }{%
     \zvec  & \uvec & \thvec          % which columns to show
  } %
  \caption[short table description]{This is a table imported from the csv file. Apart
       from that this caption is way to long to be shown in the list of tables. And
      we will also further style-up the table. }
```

Listing 3.3: Importing from csv file (example d)

z [m]	$\bar{u}$ [m s^{-1}]	$\bar{\theta}$ [K]
2001.0	2001.0	2001.0
16.0	16.0	16.0
15.0	15.0	15.0
14.0	14.0	14.0
13.0	13.0	13.0
12.0	12.0	12.0
13.0	13.0	13.0
12.0	12.0	12.0
11.0	11.0	11.0
7.0	7.0	7.0
2.0	2.0	2.0
	...	

Table 3.1: This is a table imported from the csv file. Apart from that this caption is way to long to be shown in the list of tables. And we will also further style-up the table.

3.5.2 Modifying Number Formats

3.5.3 Table Styling

We can easily add additional cells to the table.

	measured	
z [m]	$\bar{u}$ [m s^{-1}]	$\bar{\theta}$ [K]
180	2001.0	313.0
140	16.0	312.0
100	15.0	311.0
80	14.0	294.0
60	13.0	287.0
50	12.0	285.0
40	13.0	284.5
30	12.0	284.0
20	11.0	283.5
10	7.0	283.0
5	2.0	282.0
...	...	

Table 3.2: This is a table imported from the csv file. Apart from that this caption is way to long to be shown in the list of tables. And we will also further style-up the table.

What else would you like to modify in the table?

3.6 Macros

We can very much speedup work in LaTeXwith macros — simple scripts and aliases that speed-up the work.

If we repeatedly write a complicate math expression or symbol, such as

$$\theta^{(\mathrm{ML})} =$$

we can define it as a macro (in the preamble of the documents)
(example e):

```
    \newcommand{\thml}{                               % macro name
    \theta^{\scriptscriptstyle \mathrm{(ML)}}         % inner code
}                                                     % close brackets
```

and then just output $\theta^{(\mathrm{ML})}$ by calling the macro:

```
    \thml
```

Try to define a macro for your favourite math symbol and use it!

3.6.1 Macro Arguments

Macros can also take a number of arguments, here for example 3, with the first two being optional and predefined. You can see in the preable of this documents:

```
\newcommand{\sayabout}[3][mean]{
    The #2 known under name #3 is very #1
}
```

Listing 3.4: Macro with arguments (example f)

And then you call it:

```
    \sayabout[rude]{person}{Jan}
```

which outputs: The person known under name Jan is very rude .

Let us play with this macro:

1. Call it without the optional argument from the call.
2. Call it with changed arguments.

3.6.2 Redefing Macros

Not only we can define new macros, but we can *carefully* redefine them using the `\renewcommand`.

1. Look at the title at the page iv
2. Search in the preamble for `\renewcommand` and change it to something meaningful.
3. Try to redefine some other macro.

3.6.3 More on Macros

We can also write conditional macros which allows us to say IF something, THEN do something, ELSE do something else. For that we would use package `ifthen` and call it:

```
    \ifthenelse{\equal{\thing}{\string whatever}}
  {% True case
   \newcommand{\result}{thing is ‘whatever’.}%
  }
  {% false case
   \newcommand{\result}{thing is something else.}%
  }
```

We can even use `foreach` command from the package `tikz`. This is particularly useful for drawing repeated patterns.

3.7 Index Page

Index is a very popular part of books and longer documents. To generate it, we need to use the package `makeidx` in the preamble and the command

```
      \makeindex
```

also in the premable. You can see the result on the following page:

Index

3.7.1 Tasks for the Index

1. Looks up examples of index entries in this document.
2. Add some with subentries.
3. Modify the style of some of them.

3.8 Formatting

3.8.1 Dashes

Dash is not the same thing as minus. Lets see the difference between following symbols some symbols that should stay together:

-

–

—

3.9 Posters and Slideshows

If you have been struggling with Powerpoint, there is a way out . . .

The end