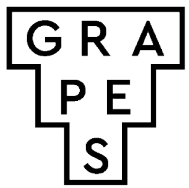
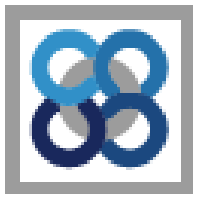
Advanced LaTeX

Albert Ruiz Cirera

Universitat Autònoma de Barcelona





BGSMath
BARCELONA GRADUATE
SCHOOL OF MATHEMATICS

Contents

Universitat Autònoma de Barcelona

About LaTeX

About LaTeX from https://www.latex-project.org/about/

LaTeX, which is pronounced "Lah-tech" or "Lay-tech" (to rhyme with "blech" or "Bertolt Brecht"), is a document preparation system for high-quality typesetting. It is most often used for medium-to-large technical or scientific documents but it can be used for almost any form of publishing.

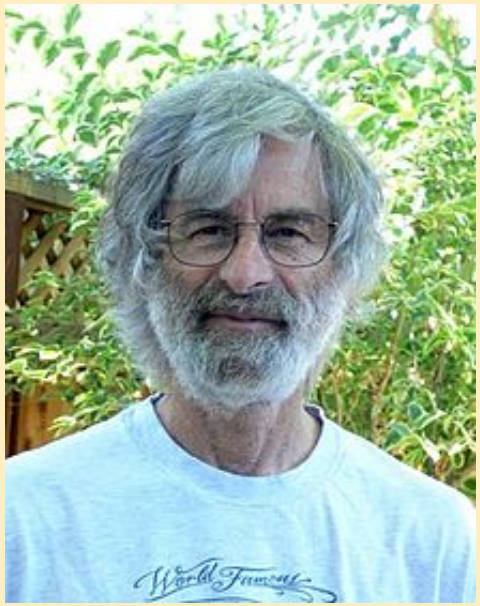
LaTeX is not a word processor! Instead, LaTeX encourages authors not to worry too much about the appearance of their documents but to concentrate on getting the right content.

History

1977 Donald E. Knuth creates TEX [Wik21b]: a typesetting system designed with two main goals: to allow anybody to produce high-quality books with minimal effort, and to provide a system that would give exactly the same result in all computers.
New versions are TeX82 (version 2) and TeX90 (version 3), which include improvements like allowing different encoding sources. Since version 3, each new version adds an extra digit so that the number asymptotically approaches π. Current version is 3.141592653.

History

1984 Leslie Lamport creates LaTeX [Wik21a]: is a software system for document preparation which uses TeX as engine: it comprises a collection of TeX macros and a program to process documents. The current version is LaTeX 2_ε: a major release from 1994 with minors updates till now.

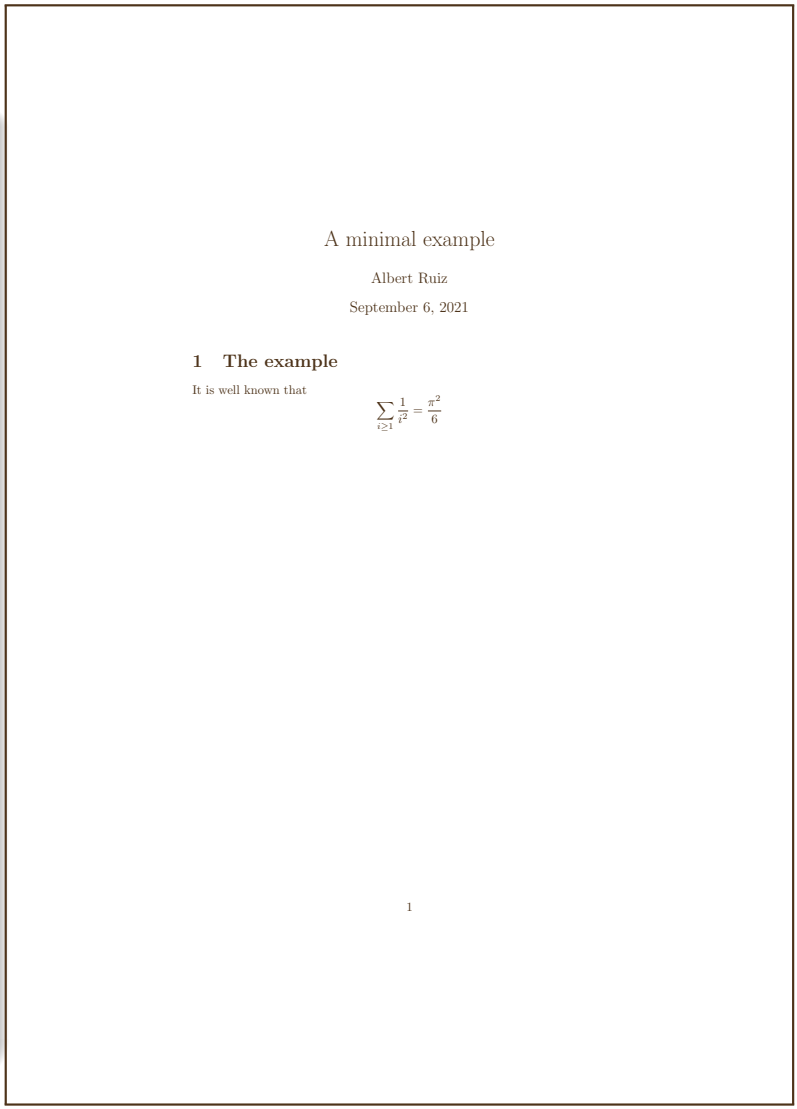


Minimal LaTeX document

Universitat Autònoma de Barcelona

minimal.tex

```
% Minimal document
\documentclass{article}
\title{A minimal example}
\author{Albert Ruiz}
\begin{document}
 \maketitle
 \section{The example}
 It is well known that
 \[
  \sum_{i\geq1} \frac{1}{i^2}=
  \frac{\pi^2}{6}
 \]
\end{document}
```


A minimal example

Albert Ruiz

September 6, 2021

1 The example

It is well known that

$$\sum_{i\geq 1} \frac{1}{i^2} = \frac{\pi^2}{6}$$

1

Important facts

- There is a main structure:
  - % is used for comments: the content of the line after % is not compiled.
  - Declare the type of document `\documentclass{}`
  - Add some headers.
  - Declare the beginning and end of the text `\begin{document}`, `\end{document}`.
- There is a structure in the document: `\section{}`.
- Special commands for producing math formulas: `\[` and `\]` are the limits and we use `\sum`, `\frac` commands to produce sums and fractions, with `{` and `}` as delimiters.
- Compilation generates some extra files: `.aux`, `.log`, `.toc`, ...
- LaTeX commands start with a backslash `\`.
- Producing the final document may need more than one compilation.

Special characters

We have seen that \ is a special character, but there are more:

- %: used for comments.
- $: used for switching between text and math mode.
- ^: used for superscripts.
- &: used for alignments.
- _: used for subscripts.
- { and }: used as delimiters for arguments.
- ~: used as space.
- #: used for arguments.

If we need to print one of those characters, we can add a \ before it: for example, \% produces %.

Preamble

Commands between `\documentclass{}` and `\begin{document}` are the preamble of the document. Usually it contains `\usepackage{}` and definition of particular commands to make it easier to produce the final result.

Body

The following commands structure the body of the document:

```
\part{title}
 \chapter{title}
  \section{title}
   \subsection{title}
    \subsubsection{title}
     \paragraph{title}
      \subparagraph{title}
```

This gives consistency to the fonts (size and type) and also produces the table of contents (if asked) and heads and foot pages (if configured).

Cross references

We can label a part of the structure of the document and reference it either before or after the position of the label.

```
\section{Structure}
\label{structure}
...
As we have seen in
Section~\ref{structure}
...
```

3. Structure

...

As we have seen in Section 3

...

Trick

The package `showkeys` shows the label in the PDF (in the corresponding place). It can be used when producing the document, specially if it has different contributors.

Lists

The environments `itemize` and `enumerate` produce lists with the same syntax.

```
\begin{itemize}
 \item The first item.
 \item The second item.
\end{itemize}
\begin{enumerate}
        \item The first item.
        \item The second item.
\end{enumerate}
```

- The first item.
- The second item.

1. The first item.
2. The second item.

Trick

The package `enumerate` allows to configure the type of enumeration. For example, `\begin{enumerate}[\bf 1.-]` produces an enumeration with labels **1.-**, **2.-**, ...

Math mode

The packages `amsmath`, `amssymb` and `amsthm` add a lot of new commands to the usual LaTeX which allow to write math symbols and formulas. So, we consider that this packages are loaded in our document.
There are two type of math modes:

- Inline equations: the formula is processed as a box which will be inserted as a character in the text. The limits are $ (open and close) or `\(` for opening and `\)` for closing.
- Display equations: the formula is processed as a box will be inserted in a new line and centered horizontally. The limits are $$ (open and close) or `\[` for opening and `\]` for closing (or `\begin{equation*}` and `\end{equation*}`).

In both cases, most spaces and line breaks do not have any significance.

Example


```
A sum inline is like
$
\sum_{i=1}^\infty\frac{1}{i^2}
$
while a sum in display style is
\[
\sum_{i=1}^\infty\frac{1}{i^2}
\]
We can force display style inline
by doing
$\displaystyle{
\sum_{i=1}^\infty\frac{1}{i^2}}
$.
```

A sum inline is like $\sum_{i=1}^\infty \frac{1}{i^2}$ while a sum in display style is

$$\sum_{i=1}^\infty \frac{1}{i^2}$$

We can force display style inline by doing $\displaystyle{\sum_{i=1}^\infty \frac{1}{i^2}}$.

The height of the math expression affects to the line separation. If we want to avoid this fact we can use `\smash{$\sum_{i=1}^\infty\frac{1}{i^2}$}`

Spaces in math mode

Math mode ignores most of the spaces we type in equations. If we want to add some extra space, or remove some space we have the following options:

- `\,` tiny space,
- `\;` small space,
- `\ ` (backslash with a space) a medium space,
- `\quad` a large space,
- `\qquad` a larger space and
- `\!` a tiny negative space.

```
\[ \int \int f(x,y)
  \mathrm{d}x \mathrm{d} y \]
vs
\[ \int\!\!\!\int f(x,y)
  \,\mathrm{d}x\,\mathrm{d} y \]
```

$$\int \int f(x,y)\mathrm{d}x\mathrm{d}y$$

vs

$$\iint f(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

Math operators

Most of the math functions are already defined: for example `$\sin(\pi)=0$` produces $\sin(\pi) = 0$ (see the difference in the font with `$sin(\pi)=0$` which produces $sin(\pi) = 0$).

We can also add operators which are not previously defined. For example, if we need to use $\mathrm{Im}(z)$ for the imaginary part of $z \in \mathbb{C}$, we can do declaring the operator in the header of the document with `\DeclareMathOperator{\ima}{Im}` or type it directly in a equation as `\operatorname{Im}`.

Delimiters (I)

A delimiter adapts the size to the vertical box which is bounding. We use commands `\left` and `\right` to define them.

```
\[
A_m=\left\{ (x_1,\dots,x_n) \mid
 \sum_{i=1}^n x_i=m \right\}
\]
```

$$A_m = \left\{ (x_1, \dots, x_n) \mid \sum_{i=1}^{n} x_i = m \right\}$$

Not every character can be a delimiter, and to avoid a delimiter (in one side) we can use a . (dot).

Delimiters (II)

Possible delimiters are (some of them need amssymb package):

```
\begin{array}{|c|c|c|c|c|c|c|c|c|c|}
 \hline [ & ] & \{ & \} & | &\| & \langle & \rangle & ( & )\\
 \hline \downarrow & \Downarrow & \uparrow & \Uparrow &
        \updownarrow & \Updownarrow & \lceil &\rceil &\lfloor &\rfloor\\
 \hline   / & \backslash &  \lmoustache & \rmoustache &
        \lgroup & \rgroup & \arrowvert & \Arrowvert & \lvert & \rvert \\
 \hline \lVert & \rVert &  \ulcorner & \urcorner & \llcorner &
        \lrcorner &  \bracevert \\ \cline{1-7}
\end{array}
```

[	]	{	}	|	‖	⟨	⟩	(	)
↓	⇓	↑	⇑	↕	⇕	⌈	⌉	⌊	⌋
/	\	⎰	⎱	⎛	⎞	|	‖	|	|
‖	‖	⌜	⌝	⌞	⌟	⎪			

Delimiters (III)

Sometimes we don't like the size of a delimiter (usually too large), or we need a delimiter to a expression which splits in severar lines. In this case, instead of using `\left` and `\right` commands, we can just change the size manually with commands like `\big`.

```
$\Big((x+1)(x-1)\Big)^2$\\
$\big(\Big(\bigg(\Bigg(\quad
\big\}\Big\}\bigg\}\Bigg\}$
```

$$\Big((x+1)(x-1)\Big)^2$$

$$\big(\Big(\bigg(\Bigg(\quad\big\}\Big\}\bigg\}\Bigg\}$$

Matrices

The environment `array` using, whenever wanted, delimiters, allows us to write matrices.

```
\[\left(\begin{array}{cc}
a & b \\ c & d
\end{array}\right)\]
```

$$\left(\begin{array}{cc} a & b \\ c & d \end{array}\right)$$

The `amsmath` package provides direct construction of matrices without specifying the horizontal alignment of the columns (which will be centered). The environments are `matrix`, `pmatrix(`, `bmatrix[`, `Bmatrix{`, `vmatrix|` and `Vmatrix||`.:

```
\[\begin{pmatrix}
a & b \\ c & d
\end{pmatrix}
\]
```

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Special cases of matrices (I)

Small matrices

If we need to inset a matrix in a inline equation, may be is better to use the environment `smallmatrix` (from `amsmath` package).

```
It is better to use $\left(\begin{smallmatrix} a & b \\
 c & d \end{smallmatrix}\right)$
than $\begin{matrix} a & b \\ c & d \end{matrix}$.
```

It is better to use $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ than $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

Special cases of matrices (II)

UAB
Universitat Autònoma de Barcelona

Cases

We can define a function by cases using `array` environment or `cases` environment (also from `amsmath` package).

```
\[|x|=\left\{\begin{array}{rl}
-x & \text{if } x<0, \\
x  & \text{if } x \geq 0
\end{array}\right.\]
```

$$|x| = \left\{ \begin{array}{rl} -x & \text{if } x < 0, \\ x & \text{if } x \geq 0 \end{array} \right.$$

```
\[|x|=\begin{cases}
-x & \text{if } x<0, \\
 x  & \text{if } x \geq 0
\end{cases}\]
```

$$|x| = \begin{cases} -x & \text{if } x < 0, \\ x & \text{if } x \geq 0 \end{cases}$$

Alignments (I)

We could also use `array` environment for grouping multiple equations, but it is better to use the `align` environment:

```
\begin{align}
  x &= (y+1)^2 \\
    &= y^2+2y+1
\end{align}
```

$$x = (y+1)^2 \quad (1)$$
$$= y^2+2y+1 \quad (2)$$

We can avoid the numbers (or labels):

```
\begin{align*}
  x &= (y+1)^2 \\
        &= y^2+2y+1
\end{align*}
```

$$x = (y+1)^2$$
$$= y^2+2y+1$$

Alignments (II)


```
\begin{align}
x  &  = (y+1)^2  \nonumber\\
   &  = y^2+2y+1
\end{align}
```

$$x = (y+1)^2 \\ = y^2 + 2y + 1 \quad (3)$$

`align` is itself a full equation, but it can be inserted in a bigger formula with `split`:

```
\[\boxed{\begin{split}
x  &  = (y+1)^2\\
   &  = y^2+2y+1
\end{split}}\]
```

$$\boxed{\begin{aligned} x &= (y+1)^2 \\ &= y^2 + 2y + 1 \end{aligned}}$$

Alignments (III)

For more than one point of alignment we can use the `alignat` environment (equivalent to `align`) and the `alignedat` environment (equivalent to `split`).

```
\begin{alignat*}{2}
x & = a + b &&-2c \\
y & = 2a &&+3c
\end{alignat*}
```

$$x = a + b - 2c$$
$$y = 2a \quad + 3c$$

Finally, in some cases we can also use the `multline` environment:

```
\begin{multline}
x = a+b+c+d+e+f+g+h+i\\
 +j+k
\end{multline}
```

$$x = a+b+c+d+e+f+g+h+i + j + k \quad (4)$$

Phantom spaces

The command \phantom produces a blank space with the same size of the argument.

Subscripts and superscripts on the left

```
\[
{}^{12}_{6}C \text{ is different than }
{}^{12}_{\phantom{1}6}C
\]
```

$$ {}^{12}_{6}C \text{ is different than } {}^{12}_{6}C $$

Tables

The main environment to build tables is the `tabular` environment:

Example of table

```
\begin{tabular}{|l|c|r|}
\hline First col. & Second col. &  Third col. \\
\hline Left & Center & Right \\
\hline
\end{tabular}
```

We get:

First col.	Second col.	Third col.
Left	Center	Right

Tables: joining columns al rows

The `\multicolumn{}{}{}` and `multirow{}{}{}` allows to join cells in a table (needs the package `multirow`).

Example

```
\begin{tabular}{|c||c|c|c|c|c||c|}
\hline
$W$ & $E_0$ & $E_1$ & $E_{-1}$ & $V_0$ & $V_1$ & $V_{-1}$ & Exotic? \\
\hline \hline
& & & & $\bigstar$ &  &  & {Yes}\\  \cline{5-8}
$\langle \omega \rangle$ &  &  &  &  & $\bigstar$ & $\bigstar$ &
{Yes}  \\ \cline{5-8}
& & & & $\bigstar$ & $\bigstar$ & $\bigstar$ &  No
\\ \hline $\langle \eta \rangle$ & $\bigstar$ &
\multicolumn{2}{c|}{ } &  & \multicolumn{2}{c||}{ } & No \\ \hline
\multirow{5}{*}{$\langle \eta, \omega \rangle$} & & \multicolumn{2}{c|}{ } &  &
\multicolumn{2}{c||}{$\bigstar$} & {Yes} \\ \cline{5-8}
&  & \multicolumn{2}{c|}{ } & \multirow{2}{*}{$\bigstar$} &
\multicolumn{2}{c||}{ } & {Yes} \\ \cline{6-8}
&  & \multicolumn{2}{c|}{ } &  &
\multicolumn{2}{c||}{$\bigstar$}  & No \\ \cline{2-8}
& \multirow{2}{*}{$\bigstar$} & \multicolumn{2}{c|}{ } &  &
\multicolumn{2}{c||}{ } & No \\ \cline{6-8}
&  & \multicolumn{2}{c|}{ } & & \multicolumn{2}{c||}{$\bigstar$} &
No \\ \hline
\end{tabular}
```

Result

W	E_0	E_1	E_{-1}	V_0	V_1	V_{-1}	Exotic?
$\langle\omega\rangle$				★			Yes
					★	★	Yes
				★	★	★	No
$\langle\eta\rangle$	★						No
$\langle\eta,\omega\rangle$					★		Yes
				★			Yes
					★		No
	★						No
					★		No

Long tables

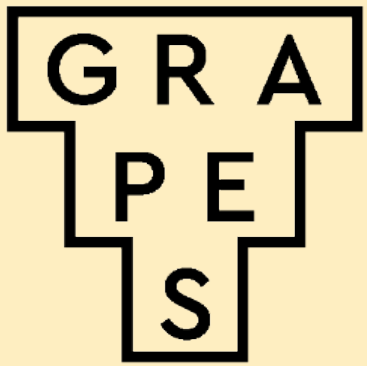
If you need a long table which may last for more than one page, we need that LaTeX 2$_\varepsilon$ decides where to break each row. In this case:

- The environment is called `longtable`.
- You have to define a main header and also a header which will show up after each page break.
- You have to define a footer which will show up before each page break and a main footer.

Including external files

The usual way to include graphics is using the package `graphicx` and use the command `\includegraphics[]{}`. In general, we don't have to worry about the format of the graphic file and LaTeX handle it correctly.

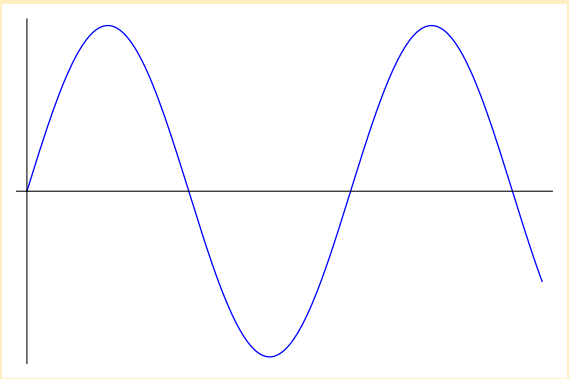
```
\begin{center}
\includegraphics[width=0.2\textwidth]{GRAPES.png}
\end{center}
```



Including LaTeX fonts

Assume that we have a picture where we are interested to add some text or formulas. For example:

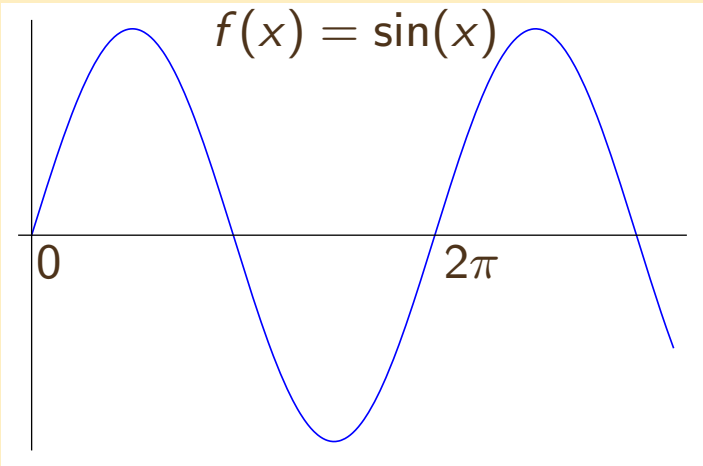
```
\begin{center}
\includegraphics[width=0.3\textwidth]{sin.pdf}
\end{center}
```



Including LaTeX fonts (II)

We can add LaTeX text or equations with the `overpic` package.

```
\begin{center}
\begin{overpic}[scale=0.4]{sin.pdf}
        \put(30,60){$f(x)=$\sin$(x)$}
    \put( 5,27){$0$}
        \put(63,27){$2$\pi$}
\end{overpic}
\end{center}
```

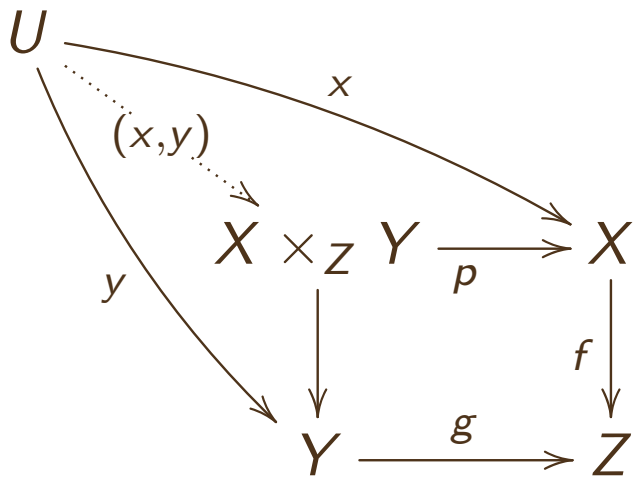


Creating graphics with LaTeX: xypic

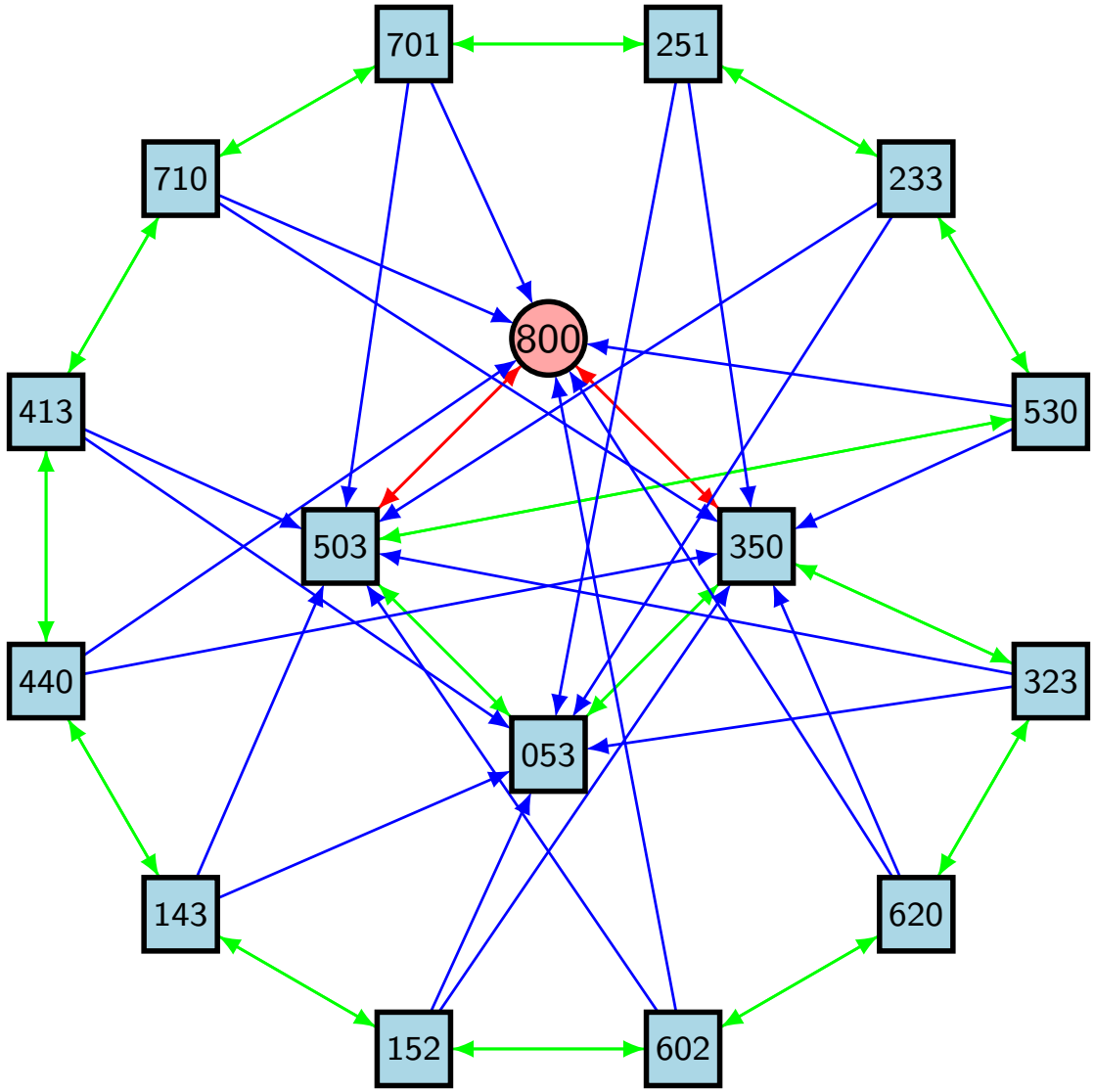
LaTeX also offers its own packages to produce graphics. Here there is a small list:

- `xypic` allows us to write diagrams in a matrix-like syntax:

```
\[\xymatrix{
U \ar@/_/[ddr]_y \ar@/^/[drr]^x
                  \ar@{.>}[dr]|-{(x,y)} \\
   & X \times_Z Y \ar[d] \ar[r]_p & X \ar[d]_f \\
   & Y \ar[r]^g  & Z }
\]
```



Creating graphics with LaTeX: tikz



Floating bodies

Figures and tables are floating bodies: you can define them in the document, but those can be placed in other pages. The final position can depend on the document (or journal) style.

```
\begin{table}
\centering
\begin{tabular}[c]{|l|c|r|}
\hline Emmy Noether & Bernhard Riemann & Gottfried Leibniz \\
\hline Carl Friedrich Gauss & Henri Poincaré
                                    & Sofya Kovalevskaya \\
\hline
\end{tabular}
\caption{Some important people}
\label{mathematicians}
\end{table}
```

Emmy Noether	Bernhard Riemann	Gottfried Leibniz
Carl Friedrich Gauss	Henri Poincaré	Sofya Kovalevskaya

Table 1: Some important people

Why floating bodies?

- The treatment of the figures and tables may depend on a journal style, so it is good to have them well defined.
- You can create a list of the figures and/or a list of the tables with commands `\listoffigures` and `\listoftables`.
- The floating position can avoid having a weird page: as it is a block, if it doesn't fit vertically at the end of a page, LaTeX will put it in a new page and will increase the separation of the lines in the previous page.

Trick

If you don't like the proposed position and you prefer to have it exactly in the place where it is typed, you can use the option `[H]` from the package `float`.

New commands (I)

Some times we would like to have more commands which are not provided by any package, and we would like to define a new one. For this purpose, we have the `\newcommand` command.
The most usual example is to define the different sets of numbers:

```
\newcommand{\N}{\mathbb{N}}
\newcommand{\Z}{\mathbb{Z}}
\newcommand{\Q}{\mathbb{Q}}
\newcommand{\R}{\mathbb{R}}
We use $\N$ for natural numbers
and $\R$ for real numbers.
```

We use $\mathbb{N}$ for natural numbers and $\mathbb{R}$ for real numbers.

New commands (II)

We can also add arguments to a command:

```
\newcommand{\bb}[1]{\mathbb{#1}}

We use $\bb{N}$ for natural
numbers and $\bb{R}$ for real
numbers.
```

We use $\mathbb{N}$ for natural numbers and $\mathbb{R}$ for real numbers.

```
\newcommand{\smat}[4]{
\tiny\left(
 \begin{array}{rr}
 #1&#2\\#3&#4
 \end{array}\right)}

We can type $\smat{-1}{2}{3}{-4}$
or $\smat abcd$.
```

We can type $\left(\begin{smallmatrix} -1 & 2 \\ 3 & -4 \end{smallmatrix}\right)$ or $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$.

New commands (II)

And also optional parameters which can have a fixed value when these are not provided:

```
\newcommand{\plusbin}[3][2]{(#2+#3)^#1}
3 is the numbers of parameters,
2 is the default value of the first parameter and
also implies that it is optional.
\[\plusbin{a}{b} \text{ or } \plusbin[3]{x}{y}\]
```

3 is the numbers of parameters, 2 is the default value of the first parameter and also implies that it is optional.

$$(a+b)^2 \text{ or } (x+y)^3$$

Finally, if we want to modify an existing command we must do it with `\renewcommand`.

Counters

Any document in LaTeX contain values in some variables to define counters (for example, the page number, the section number, ...). The name of the main counters are:

`part`	`paragraph`	`figure`	`enumi`
`chapter`	`subparagraph`	`table`	`enumii`
`section`	`page`	`footnote`	`enumiii`
`subsection`	`equation`	`mpfootnote`	`enumiv`
`subsubsection`			

Each counter has an associated command (which can be redefined) with the name `\the`+counter name. For example, `\thepage` prints the current page number in the same way as it is in the document.
To define a new counter we can use `\newcounter` and a value of a counter can be modified with `\setcounter` or `\addtocounter`.

Lengths

Any document in LaTeX contain values in some variables to define lengths (for example, to determine the layout of a page).
The value of a length can be modified with `\setlength` or `\addtolength`, and we always have to say the length unit. For example: `\addtolength{\topmargin}{1cm}` adds one centimeter to the top margin of the document.

- There are units which depend on the fonts: `em` is the width of "m", and `ex` is the height of "x".
- `\settoheight{l}{t}` (or `\settowidth{l}{t}`) is used to assign to the length `l` the height (or width) of the text `t`.
- If we are just interested in modifying the margins of a document, it is easier to use `geometry` package.

Bibliography (I)

There are several ways to add the bibliography in a document. As it may depend on the journal style, we can add it in a way that any entry is divided by fields which are processed by LaTeX.
First we need a `bib` file (call it here `file.bib`) which has the labeled entries of all references. For example:

```
@book {KnuthACP1,
        AUTHOR = {Knuth, Donald E.},
        TITLE = {The art of computer programming. {V}ol. 1},
        NOTE = {Fundamental algorithms,
                Third edition [of MR0286317]},
        PUBLISHER = {Addison-Wesley, Reading, MA},
        YEAR = {1997},
        PAGES = {xx+650},
        ISBN = {0-201-89683-4},
        MRCLASS = {68-02},
        MRNUMBER = {3077152},
}
```

Bibliography (II)

Then, all references in the `file.bib` file which are cite in the document will be added when required by the command `\bibliography{file.bib}`.

To cite a document, we need the command `\cite[op]{label}`, where `label` is the name in the bib file (`KnuthACP1` in our case) and the optional argument is used to add the precise result.

All entries in the `bib` file which are not cited will not appear in the bibliography. If we want to add a non cited reference we need to do it with the `\nocite{label}` command.

Current installations of LaTeX detect changes in the bibliography and updates the result whenever is needed. In old installations we need to run `BibTeX` to update the information.

Presentations with Beamer

`Beamer` is just a type of document which produces an output like this slides. All that is there is LaTeX, but contains a lot of particular parameters which can be configured.

One particularity is that with `beamer` one needs to define the page breaks (or the slides).

There is detailed information in [Til21].

Interaction with other software

The following software is prepared to produce LaTeX document directly from code:

1. `R` (or `RStudio`) with the package `knitr`. It allows to produce a LaTeX document including static text and also computed results and graphics.
2. Any `Jupyter` notebook (`Julia`, `Python`, `R`, `SageMath`,...).

References

- Till Tantau, Joseph Wright, Vedran Miletić, *The beamer class*, 2021, [Online; accessed 1-September-2021].
- Tobias Oetiker and Contributors, *The not so short introduction to latex*, 2021, [Online; accessed 1-September-2021].
- Wikipedia contributors, *Latex — Wikipedia, the free encyclopedia*, 2021, [Online; accessed 1-September-2021].
- _____, *Tex — Wikipedia, the free encyclopedia*, 2021, [Online; accessed 1-September-2021].

Thank you for your attention!