The Ti*k*Z graphics package

Sebastiano Tronto

2021-03-26

UNIVERSITÉ DU LUXEMBOURG

TikZ

"TikZ ist *kein* Zeichenprogramm"

- "Writing" graphics as you write text and formulas in LaTeX

$$\text{TikZ} : \text{graphics} = \text{LaTeX} : \text{text}$$

- Draw shapes, paths, diagrams...
- Countless extension packages

UNIVERSITÉ DU LUXEMBOURG

References

- Wikibooks, short introduction:
  `https://en.wikibooks.org/wiki/LaTeX/PGF/TikZ`
- Official manual:
  `http://ctan.cs.uu.nl/graphics/pgf/base/doc/pgfmanual.pdf`
  (Too long, but nice examples in part I)
- Extension packages and their documentation:
  `https://www.ctan.org/topic/pgf-tikz`

UNIVERSITÉ DU LUXEMBOURG

Using TikZ

- In preamble:

```
\usepackage{tikz}
\usetikzlibrary{something} % if needed
```

- In document body:

```
\begin{tikzpicture} (...)  \end{tikzpicture}
```

- Use `[scale=`*n*`,rotate=`*angle*`]` to scale or rotate the whole picture.

UNIVERSITÉ DU LUXEMBOURG

Coordinates

- 3 ways to express coordinates:
  - Cartesian, no unit = cm
    Example: `(2cm,11pt)`
  - Polar
    Example: `(180:7cm)`
  - Intersection of vertical line through p_1 and horizontal line through p_2, points expressed as above (no parenthesis)
    Example: `(0,1 |- 30:2)`
- More intersections: `\usetikzlibrary{intersections}`
- Give names to points: `\coordinate (X) at (1,-4);`

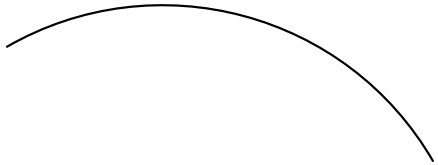
Drawing straight lines

```
\draw (P1) -- (P2) -- ...  -- (Pn);
```

- Points expressed in coordinates as before.
- Add `-- cycle` to close the path.

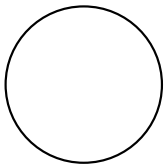
UNIVERSITÉ DU LUXEMBOURG

Curved lines and other shapes

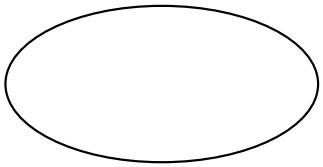


```
\draw (0,0) arc [start angle=30,
  end angle=120, radius=2cm];
```

```
\draw (0,0) rectangle (2,1);
```



```
\draw (0,0) circle [radius=0.5];
```



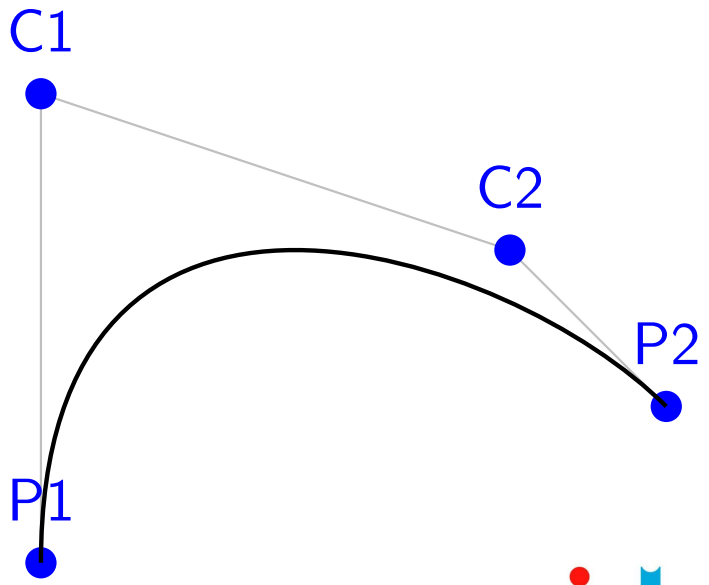
```
...[x radius=1,y radius=0.5];
```

Bezier curves

```
\draw (P1) ..controls (C1) and (C2)..  (P2);
```

A curve from P1 to P2, starting in direction of C1 and arriving from the direction of C2 (usually not touching the control points).

https://en.wikipedia.org/wiki/Bézier_curve



UNIVERSITÉ DU LUXEMBOURG

Colors

- Color names already defined: `red`, `green`, `blue`, `yellow`, `black`, `white`, `gray`, `darkgray`, `lightgray`, `brown`, `pink`...
- Specify intensity: `color!n` with $0 \leq n \leq 100$.
- Mix colors: `color1!n1!color2!n2!...`
- Example:

  `blue!50!red!50!green`

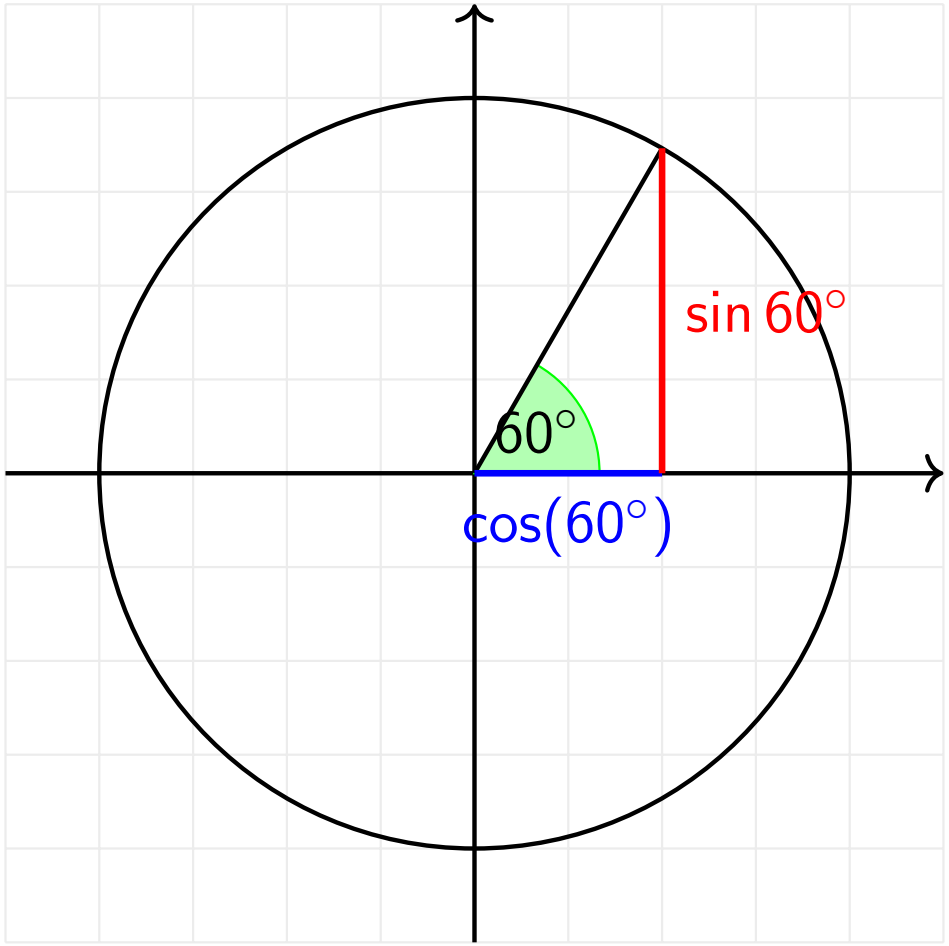
  is 50% blue, 25% red and 25% green.

Filldraw, change color and line style

- `\draw[`*`colorname`*`]` to specify color.
- `\filldraw[fill=`*`fillcolor`*`, draw=`*`bordercolor`*`]` to fill path or `\fill` for no border.
- Line width: `\draw[`*`thickness`*`]`, where *`thickness`* can be `very thin, thin, thick, very thick...` or `\draw[line width=`*`length`*`]` where *`length`* can be `3pt, 0.1mm...`
- Line style: `[dashed]` for dashed, `[->]` or `[<-]` for arrow.

UNIVERSITÉ DU LUXEMBOURG

Example



UNIVERSITÉ DU LUXEMBOURG

Adding text: nodes

```
\draw (P1) -- node[position] {text} (P2) ...

\draw (P1) node[position] {text} -- (P2) ...
```

- A node can refer to a line or to a point
- *position* can be above, below, left or right
- *text* can also be $math$

UNIVERSITÉ DU LUXEMBOURG

Macros

\pgfmathsetmacro{\x}{*value*}

Examples:

```
\pgfmathsetmacro{\r}{4}
\pgfmathsetmacro{\a}{30}
```

UNIVERSITÉ DU LUXEMBOURG

Example - TikZ code

```
\begin{tikzpicture}
    \colorlet{coscolor}{blue}
    \colorlet{sincolor}{red}
    \tikzset{anglefill/.style={draw=green,fill=green!30}}
    \pgfmathsetmacro{\r}{4}
    \pgfmathsetmacro{\a}{60}

    \draw[lightgray!30] (-5,-5) grid[step=1] (5,5);
    \draw[thick,->] (0,-5) -- (0,5);
    \draw[thick,->] (-5,0) -- (5,0);

    \filldraw[anglefill] (0,0) -- node[above]{$\a^\circ$}
        (\r/3,0) arc [start angle=0,end angle=\a,radius=\r/3] -- cycle;
    \draw[thick] (0,0) circle[radius=\r] -- (\a:\r);
    \draw[very thick,coscolor] (0,0) --
        node[below]{$\cos(\a^\circ)$} (\r*cos{\a},0);
    \draw[very thick,sincolor] (\r*cos{\a},0) --
        node[right]{$\sin\a^\circ$}(\a:\r);
\end{tikzpicture}
```

UNIVERSITÉ DU LUXEMBOURG

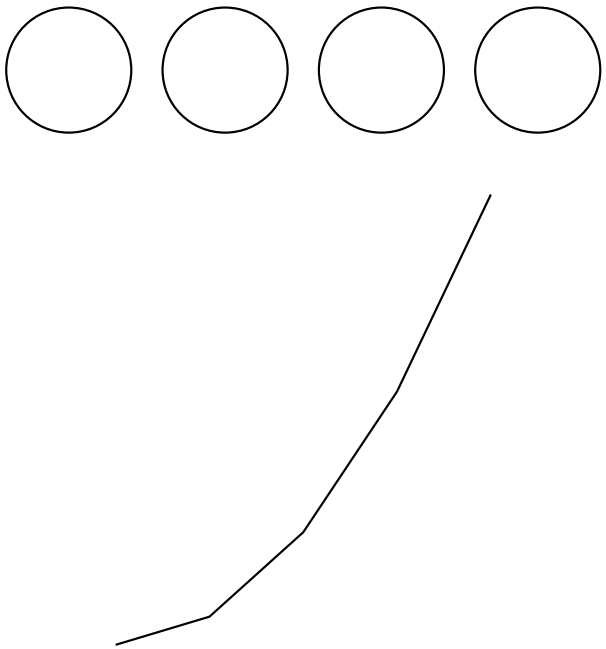
The \foreach command

\foreach \i in {*list*} { *commands* };

- *list* can be fully explicit (like {1,7.2,-42}) or partially implicit (like {1.5,1.6,...,5.0})
- *commands* will be repeated with \i varying in *list*
- One can use foreach inside a \draw

UNIVERSITÉ DU LUXEMBOURG

\foreach examples



```
\foreach \i in {1,2,3,4}
{\draw (\i,0) circle [radius=0.4];}
```

```
\draw (0,0) \foreach \i in
 {0.0,0.3,...,1.5} {-- (\i,\i^2)};
```

UNIVERSITÉ DU LUXEMBOURG

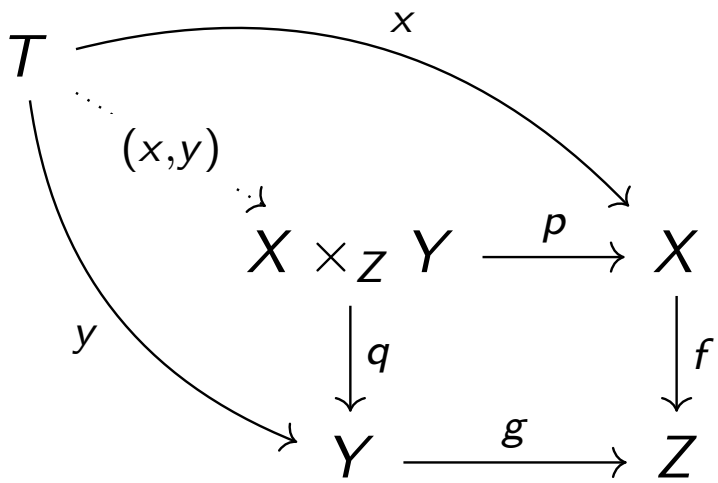
External packages

Many external packages, include with \usepackage:
`https://www.ctan.org/topic/pgf-tikz`

- Graphs and similar: `tikz-cd`, `adigraph`, `binarytree`...
- Diagrams: `pgf-pie`, `bchart`, `venndiagram`...
- Other sciences: `chemfig`, `CircuiTikZ`...
- Fun: `battleship`, `TikZducks`, `tikz-among-us`...

UNIVERSITÉ DU LUXEMBOURG

Commutative diagrams



UNIVERSITÉ DU LUXEMBOURG

tikz-cd

Reference:

`http://ctan.cs.uu.nl/graphics/pgf/contrib/tikz-cd/tikz-cd-doc.pdf`

```
\usepackage{tikz-cd}
```

```
\begin{tikzcd}...\end{tikzcd}
```

- Works as a `tabular` or `matrix` (with `&` and `\\`)
- Everything is in math mode by default

UNIVERSITÉ DU LUXEMBOURG

Arrows

`\arrow[`*direction*`,"label",other options]`

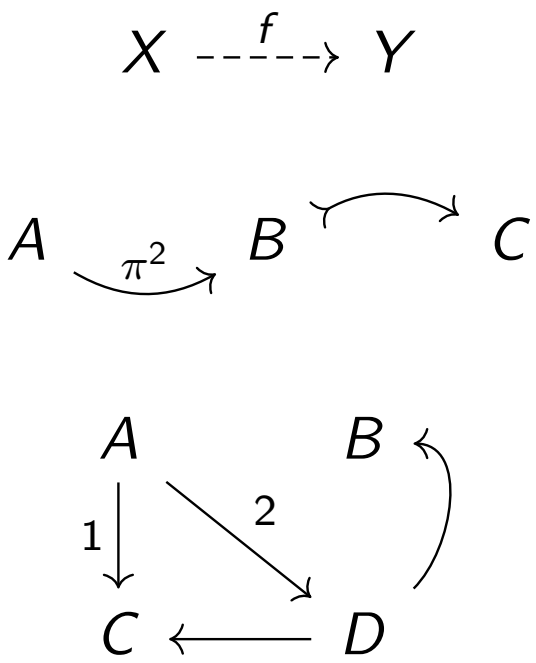
- *direction* can be any combination of the letters `r` (right), `l` (left), `d` (down) and `u` (up)
- The target must exist:

```
X \arrow[r] & Y    % Ok
X \arrow[r]        % Error
X \arrow[r] & {}   % Ok
```

- Other options describe the shape and style of the arrow

UNIVERSITÉ DU LUXEMBOURG

Examples




```
X \arrow[r,dashed,"f"] & Y
```

```
A \arrow[r,bend right,"\pi^2"] &
B \arrow[r,bend left,tail] & C
```

```
A \arrow[d,"1"'] \arrow[dr,"2"] & B \\
C & D \arrow[l] \arrow[u,out=45,in=0]
```

UNIVERSITÉ DU LUXEMBOURG