Graphical Elements in Mathematical Writing

and the Tools to Make Them

Ana Cannas

29. October 2025

Mathematical Writing, ETH Zürich

Slides prepared with assistance from ChatGPT 5.

Plan for today

1. Why and what graphical elements?

   as introduction

2. Matrices, arrays, and tables

   with mostly basic LaTeX

3. Graphs, diagrams, and plots

   with TikZ

4. Weaving graphics into a paper

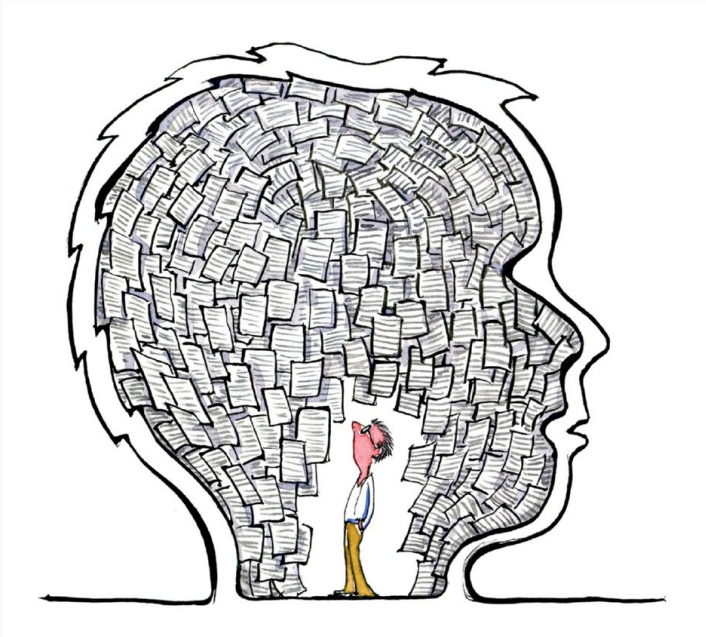
   with captions, references, and alt text

More information on Overleaf, for instance:

Overleaf on Matrices Overleaf on Tikz

Why and what graphical elements?

Why graphical elements?



Author HikingArtist via wikimedia

A picture is worth a thousand words.

Brain processes visual information significantly faster than text

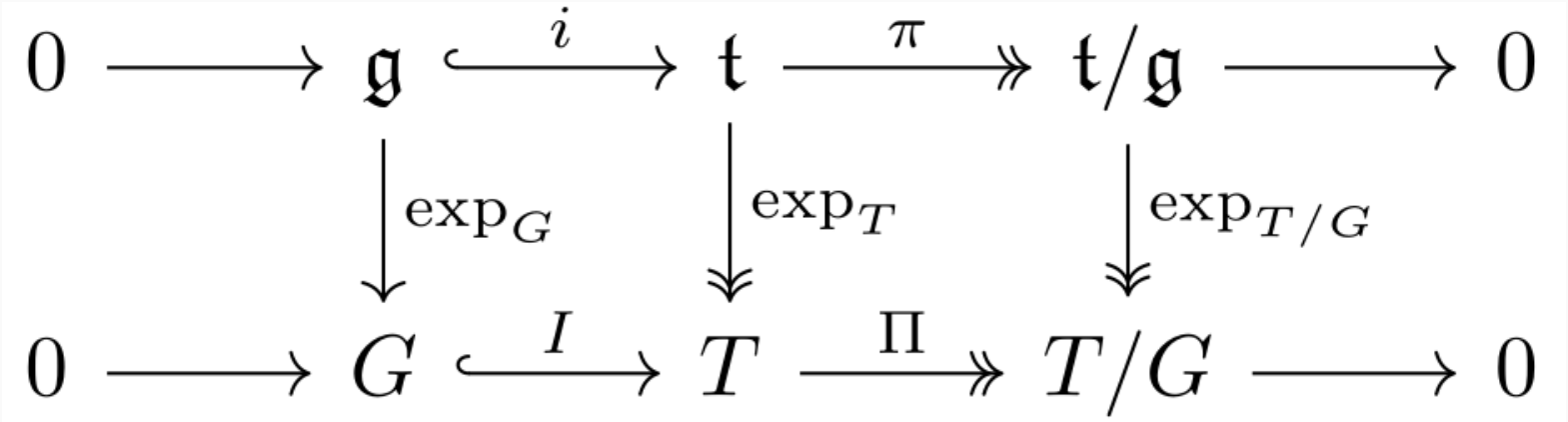
$\Longrightarrow$ Add visuals to text **and** add text to visuals

A picture may be worth a thousand words,
a formula is worth a thousand pictures.

Edsger Dijkstra, 1930-2002, computer scientist

Example: A commutative diagram

A commutative diagram is worth a thousand formulas.



LaTeX code to be found in template on the webpage.

Clearer and more effective communication

- **Offload**:

  Visual schemes convey patterns, structure, and dependence.
- **Guidance**:

  Well-chosen diagrams enlighten proofs and algorithms.
- **Compression**:

  Tables and charts condense information efficiently.
- **Redundancy**:

  Effective images reinforce the message and prevent confusion.
- **Distillation**:

  Graphs and diagrams chunk and signal complex connections.

Common graphical elements

- matrices
- arrays
- tables

} mostly basic LaTeX

- graphs
- diagrams
- plots

} mostly `tikz` and `pgfplots`

- photos (JPG, PNG, GIF)
- other (EPS, PDF, SVG)
- etc

} mostly with `graphicx`

Some program names

TEX, pronounced “tech”

typesetting program created by Donald Knuth in the 1970s

LATEX, pronounced “lah-tech” or “lay-tech”

created by Leslie **La**mport in the early 1980s, building on TEX

TikZ

created by **Ti**ll Tantau in 2005; **k***ein* **Z***eichenprogramm*

GeoGebra

created by Markus Hohenwarter in 2001; **geo**metry & al**gebra**

Matrices, arrays, and tables

Generic set-up

```
\begin{ENVIRONMENT_NAME}
  DATA
\end{ENVIRONMENT_NAME}
```

where `ENVIRONMENT_NAME` can be

`matrix` or `array` or `tabular` or ...

Matrices and arrays are for math mode.

Controlling alignment within the `DATA`:

`r`	right alignment
`c`	center alignment
`l`	left alignment
`&`	start new column
`\\`	start new row (or line)

Matrix environments with package amsmath

Matrix (parentheses):

Input:

```
\[
A=\begin{pmatrix}
1 & 2 & 3\\
4 & 5 & 6
\end{pmatrix}
\]
```

Output:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

Matrix (square brackets):

Input:

```
\[
A=\begin{bmatrix}
1 & 2 & 3\\
4 & 5 & 6
\end{bmatrix}
\]
```

Output:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Matrices: plain vs. customized (with package amsmath)

Input:

```
\[
A=\begin{matrix}
1 & 2 & 3\\
4 & 5 & 6
\end{matrix}
\]
```

Output:

$$A = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{matrix}$$

Input:

```
\[
A=\left\langle
\begin{matrix}
1 & 2 & 3\\
4 & 5 & 6
\end{matrix}
\right\}
\]
```

Output:

$$A = \left\langle \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{matrix} \right\}$$

Aligned arrays with package `amsmath`

Unnumbered `align*`

Input:

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

Output:

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

Numbered `align`

Input:

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

Output:

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

Aligned arrays with plain LaTeX

Aligned formulas

Input:

```
\[
\begin{array}{rcl}
f(x) &=& x^2-1\\
f'(x)&=& 2x
\end{array}
\]
```

Output:

$$\begin{array}{rcl} f(x) &=& x^2-1\\ f'(x)&=& 2x \end{array}$$

More aligned formulas

Input:

```
\[
\begin{array}{rclcrcl}
A &=& B &\iff& C&=&D\\
AA&=& BB&\iff&CC&=&DD
\end{array}
\]
```

Output:

$$\begin{array}{rclcrcl} A &=& B &\iff& C&=&D\\ AA&=& BB&\iff&CC&=&DD \end{array}$$

Example: row elimination with plain LaTeX

Input:

```
\[
\left(\begin{array}{cc|c}
  1 & 2 & 3\\
  4 & 5 & 6\\
  3 & 2 & 1
\end{array}\right)
\longrightarrow
\left(\begin{array}{cr|r}
1 & 2 & 3\\
0 & -3 & -6\\
0 & -4 & -8
\end{array}\right)
\]
```

Output:

$$\left(\begin{array}{cc|c} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 3 & 2 & 1 \end{array}\right) \longrightarrow \left(\begin{array}{cr|r} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & -4 & -8 \end{array}\right)$$

Example with `array`, `pmatrix`, `align`, `color`, and `tikz`

LaTeX code to be found in template on the webpage.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{i1} & a_{i2} & \dots & \textcircled{a_{ij}} & \dots & a_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mj} & \dots & a_{mn} \end{pmatrix} \leftarrow i\text{th row}$$

$$\uparrow$$

jth column

Text tables with `tabular` environment

Input:

```
\begin{tabular}{|c|c|l|}
\hline
Function  & Domain        & Name \\ \hline \hline
$\cos$    & $\mathbb{R}$ & Cosine      \\ \hline
$\arccos$ & $[-1,1]$     & Arccosine   \\ \hline
\end{tabular}
```

Output:

Function	Domain	Name
cos	$\mathbb{R}$	Cosine
arccos	$[-1, 1]$	Arccosine

`r`	right
`c`	center
`l`	left
`|`	vert. line
`\hline`	horiz. line

Combine rows or columns using package `multirow`

```
\begin{tabular}{|p{2cm}|p{2cm}|p{2.5cm}||p{2.5cm}|}
\hline
\multicolumn{3}{|c|}{\textbf{Multiple column}}
 & Simple column \\ \hline
First column wraps down & Column 2 & Column 3
 & Column 4 \\ \hline $B_1$ & $B_2$ &
\multirow{2}{6em}{\textbf{Multiple row}} & $B_4$ \\
\cline{1-2} \cline{4-4} $C_1$ & $C_2$ & & $C_4$ \\ \hline
\end{tabular}
```

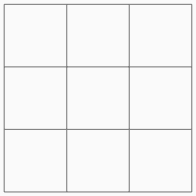
Multiple column			Simple column
First column wraps down	Column 2	Column 3	Column 4
B_1	B_2	**Multiple row**	B_4
C_1	C_2		C_4

Graphs, diagrams, and plots

Getting started with the package `tikz`

The **default coordinates** are centimeters, with the usual sense:

$(0,3)$ $(3,3)$



$(0,0)$ $(3,0)$

The examples on this and the next three pages are adapted from J. Lees-Miller Overleaf slides "An Interactive Introduction to LaTeX, Part 3".

Input:

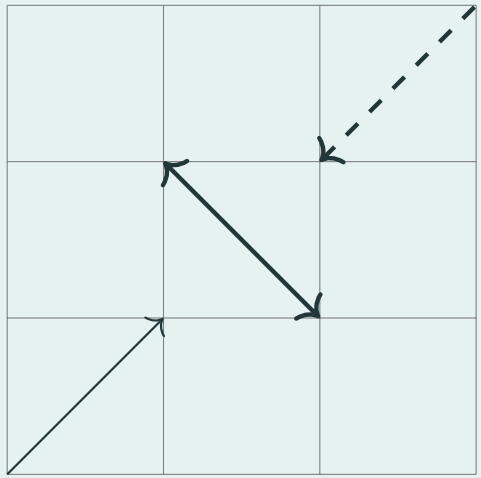
```
\begin{tikzpicture}

% Draw a grid
\draw[help lines] (0,0) grid (3,3);

% Draw three arrows with different styles
\draw[->] (0,0) -- (1,1);
\draw[<->, thick] (2,1) -- (1,2);
\draw[<-, thick, dashed] (2,2)--(3,3);

\end{tikzpicture}
```

Output:



Drawing with TikZ

The **options in the first line** modify the whole picture.

Example: `[scale=0.7,rotate=45]`

Each drawing command should end with a **semicolon** ; .

The command `\draw` can take **options**, including colors.

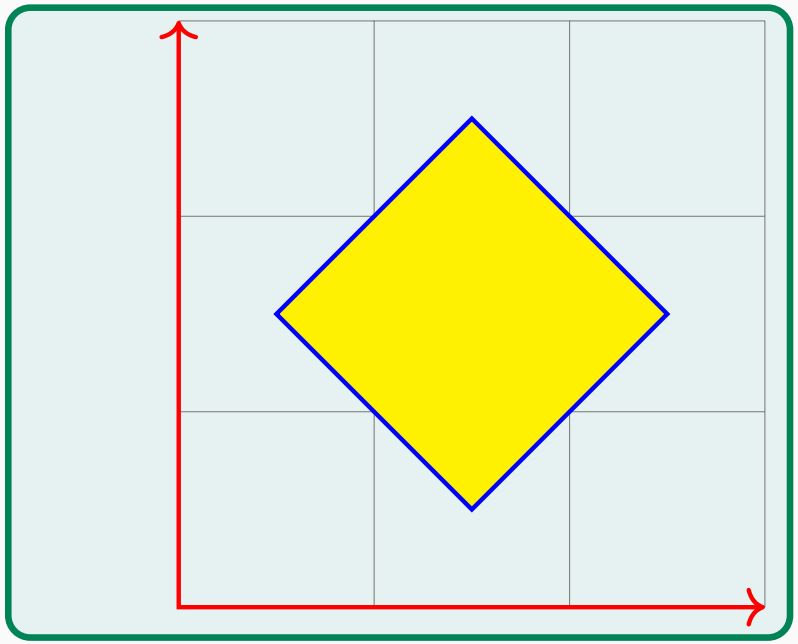
Input:

```
\begin{tikzpicture}[scale=1.25]
% grid
\draw[help lines] (0,0) grid (3,3);

% axes
\draw[<->, thick, red]
  (0,3)--(0,0)--(3,0);

% diamond
\draw[thick, blue, fill=yellow]
  (1.5,0.5) -- (2.5,1.5) --
  (1.5,2.5) -- (0.5,1.5) --
  cycle;
\end{tikzpicture}
```

Output:



Nodes and labels in TikZ

Use **nodes** to place text (and math) in TikZ drawings.

You can also use nodes as coordinates — useful for diagrams.

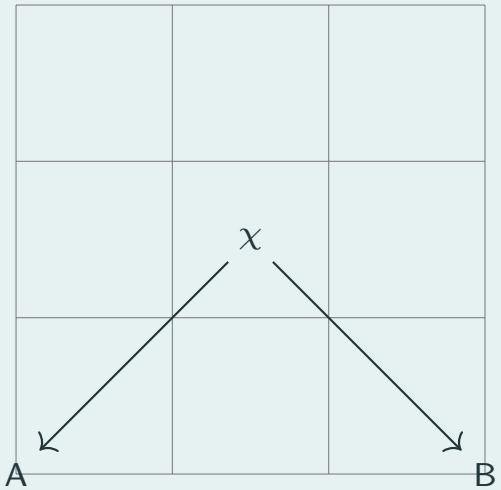
Input:

```
\begin{tikzpicture}
\draw[help lines] (0,0) grid (3,3);

% nodes
\node (a) at (0,0) {A};
\node (x) at (1.5,1.5) {$\chi$};
\node (b) at (3,0) {B};

% arrows
\draw[->] (x) -- (a);
\draw[->] (x) -- (b);
\end{tikzpicture}
```

Output:

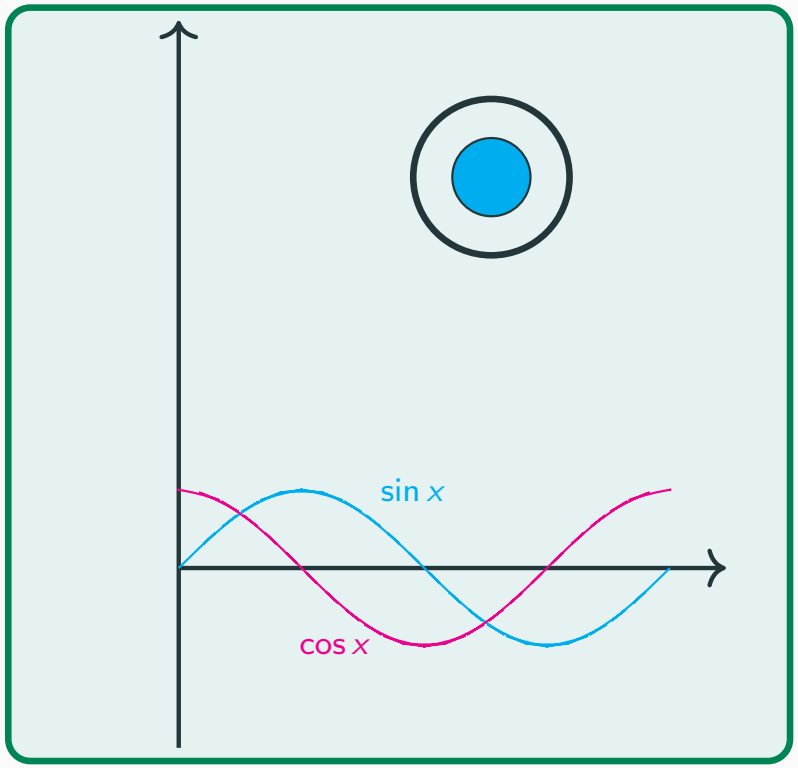


Simple shapes and simple functions with TikZ

Input:

```
\begin{tikzpicture}[scale=0.5]
% axes x and y
\draw[ ->, thick] (0,0) -- (7, 0);
\draw[->, thick] (0,-2) -- (0,7);
% curves
\draw[cyan,domain=0:2*pi]
  plot (\x, {sin(\x r)});
\draw[magenta,domain=0:2*pi]
  plot (\x, {cos(\x r)});
% circle of radius 1 centered at (4,5)
\draw[very thick] (4,5) circle (1);
% disk of radius 0.5 centered at (4,5)
\draw[fill=cyan] (4,5) circle (0.5);
% text labels
\node[cyan] at (3,1) {$\sin x$};
\node[magenta] at (2,-1) {$\cos x$};
\end{tikzpicture}
```

Output:



Commutative diagrams with `tikz-cd`

Input:

```
\[
\begin{tikzcd}
A \arrow[r,"f"] \arrow[d,"g"']
& B \arrow[d,"h"] \\
C \arrow[r,"k"']
& D
\end{tikzcd}
\]
```

Output:

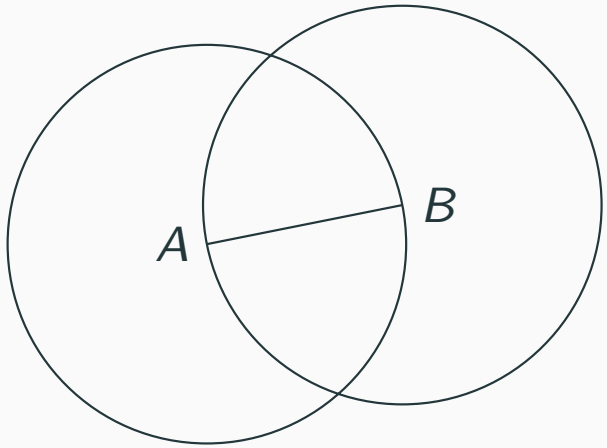
$$\begin{tikzcd} A \arrow[r,"f"] \arrow[d,"g"'] & B \arrow[d,"h"] \\ C \arrow[r,"k"'] & D \end{tikzcd}$$

Using TikZ library `calc` for computing within TikZ

Input:

```
\begin{tikzpicture}
\coordinate [label=left:$A$]  (A) at (0,0);
\coordinate [label=right:$B$] (B) at (1.25,0.25);
\draw (A) -- (B);
\draw let \p1 = ($ (B) - (A) $),
          \n1 = {veclen(\x1,\y1)}
       in
          (A) circle (\n1)
          (B) circle (\n1);
\end{tikzpicture}
```

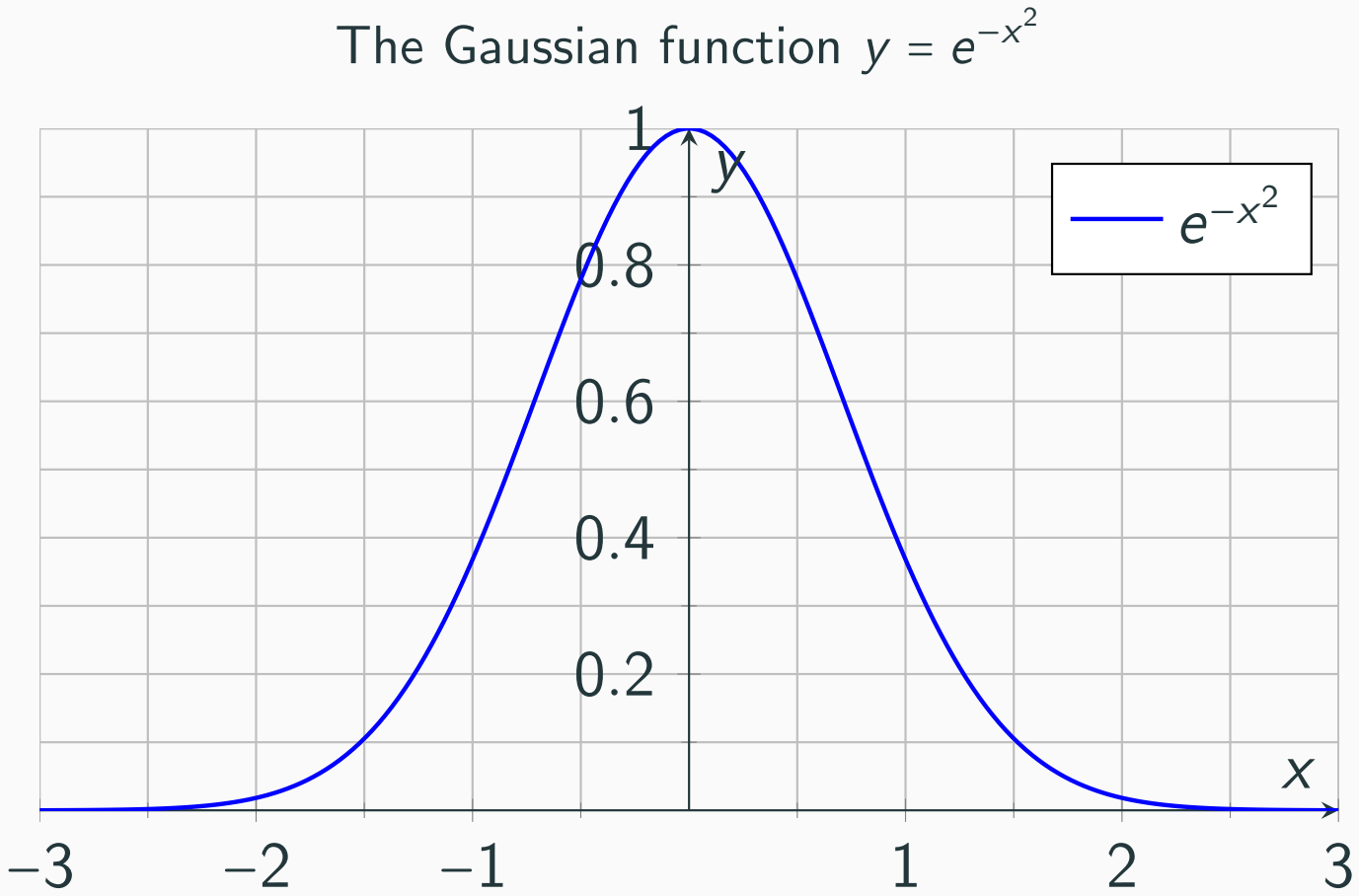
Output:



From a tutorial in the TikZ Manual online

`\x*` and `\y*` are the coordinates of the point `\p*` defined with "`let`".

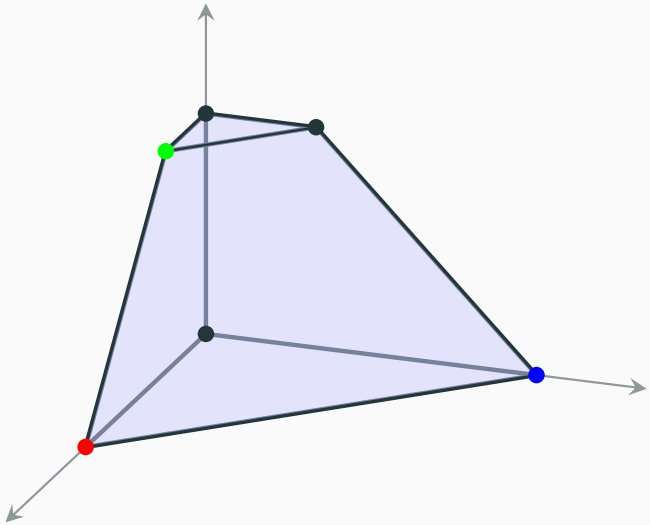
Using package `pgfplots` for a function graph with TikZ

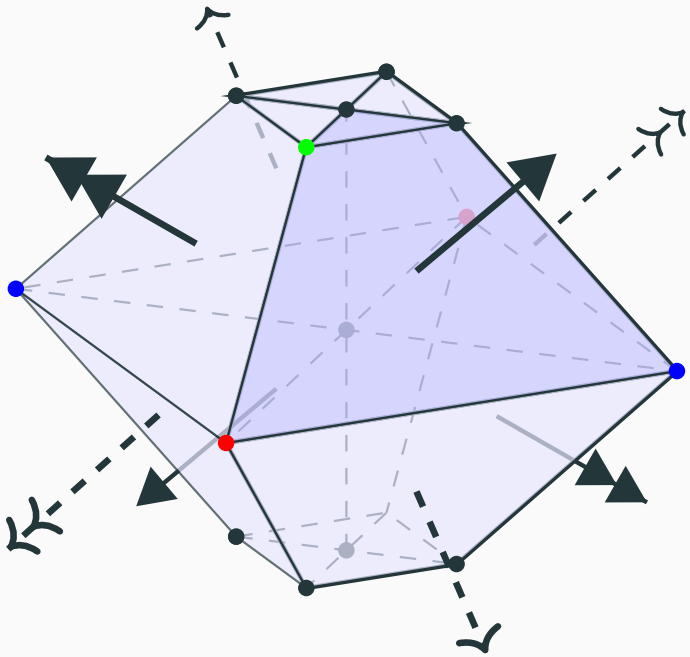


LaTeX code to be found in template on the webpage.

A figure with package `tikz-3dplot`

The viewers angle can be changed with `\tdplotsetmaincoords`.

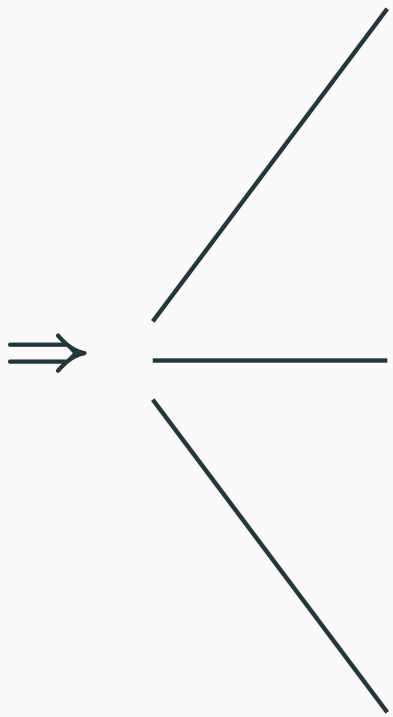




LaTeX code to be found in template on the webpage.

Shortcuts to obtain TikZ code

Figures take a long time to create.



Use free **TikZ editors** such as

- TikZiT – to install
- Commutative diagrams – online
- TikzMaker – online

or adapt from free **examples on line**,

for instance at TeXample.net

or prompt **AI**.

- Works best when similar figures are available online.
- Break down complex figures into simpler components to tackle separately.

Example 1: Prompting ChatGPT for 2D picture

Prompt:

Please give me latex tikz code for a light shaded triangle with vertices at (1,0), (2,0), (1,1) drawn with thick edges and including the axes labelled x and y.

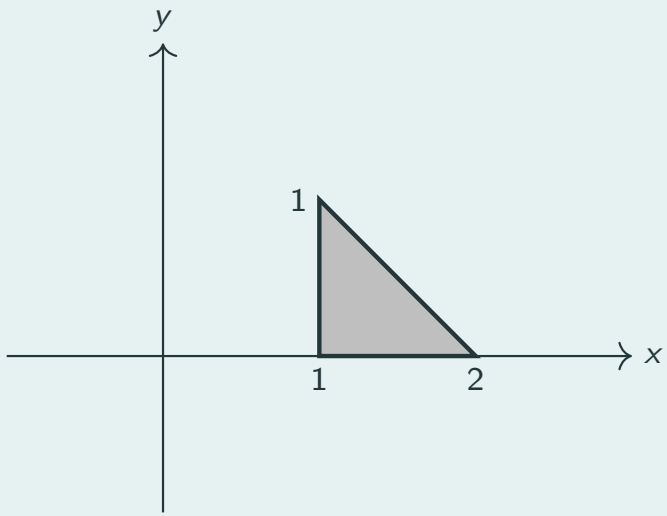
Code:

```
\begin{tikzpicture}
% Axes
\draw[->] (-1,0)--(3,0) node[right] {$x$};
\draw[->] (0,-1)--(0,2) node[above] {$y$};

% Triangle
\draw[thick, fill=lightgray]
(1,0) -- (2,0) -- (1,1) -- cycle;

% Vertex labels
\node[below] at (1,0) {$1$};
\node[below] at (2,0) {$2$};
\node[left] at (1,1) {$1$};
\end{tikzpicture}
```

Output:

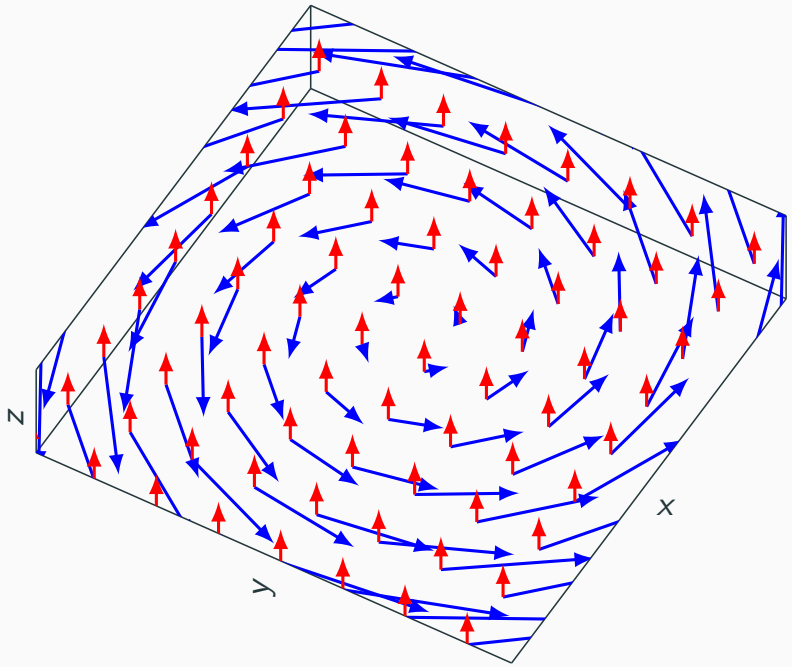


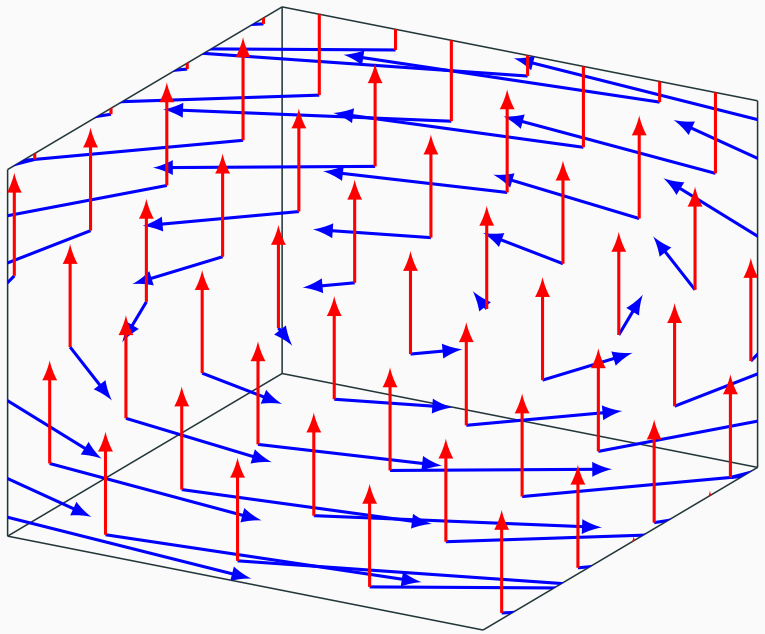
Example 2: Prompting ChatGPT for 3D plot (using `pgfplots`)

Prompt:

produce latex code for an image of vector field (-y,x,0) in blue and for vector field (0,0,2) in red with axes but without showing any numbers

Output variations after manual tweaks:

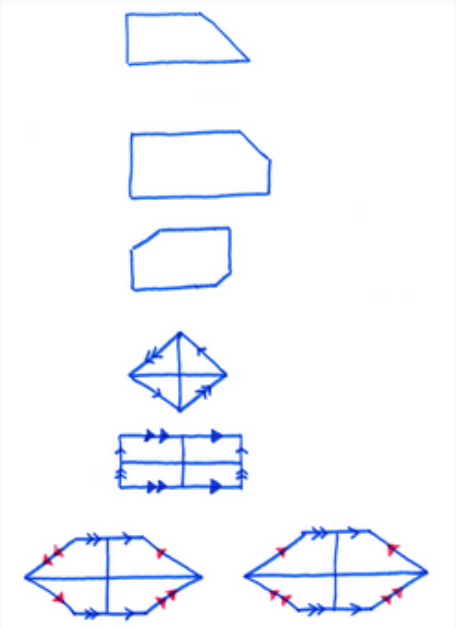




Example 3: Prompting ChatGPT with hand drawing

Prompt:

Could you please give me LaTeX Tikz code for each of the seven polygonal figures in this file?



Result:

Polygon 1: Triangle

A simple triangle can be drawn using TikZ with the following code:

latex Copy code

```
\documentclass[tikz,border=10pt]{standalone}
\begin{document}
\begin{tikzpicture}
    \draw (0,0) -- (1,0) -- (0.5,0.86) -- cycle;
\end{tikzpicture}
\end{document}
```

This code creates an equilateral triangle. Adjust the coordinates to match the specific triangle in the image if needed.

Polygon 2: Quadrilateral

For a general quadrilateral:

latex Copy code

```
\documentclass[tikz,border=10pt]{standalone}
\begin{document}
\begin{tikzpicture}
    \draw (0,0) -- (1,0) -- (1.5,0.5) -- (0.5,0.5) -- cycle;
\end{tikzpicture}
\end{document}
```

This code is for a simple quadrilateral. Modify the points to fit the exact shape in the image.

Polygon 3: Pentagon

A simple pentagon can be represented as:

latex Copy code

```
\documentclass[tikz,border=10pt]{standalone}
\begin{document}
\begin{tikzpicture}
    \draw (0,0) -- (1,0) -- (1.2,0.8) -- (0.5,1.3) -- (-0.2,0.8) -- cycle;
\end{tikzpicture}
\end{document}
```

Adjust the vertices to match the specific pentagon.

Polygon 4: Hexagon

Weaving graphics into a paper

Add a caption

☞ Add a descriptive **caption** for the figure's content.

Input:

```
\begin{figure}
...
\caption{The $\Delta$-kaleidoscope...}
\label{fig:hirzebruch}  % The label should follow the caption.
\end{figure}
```

Output:

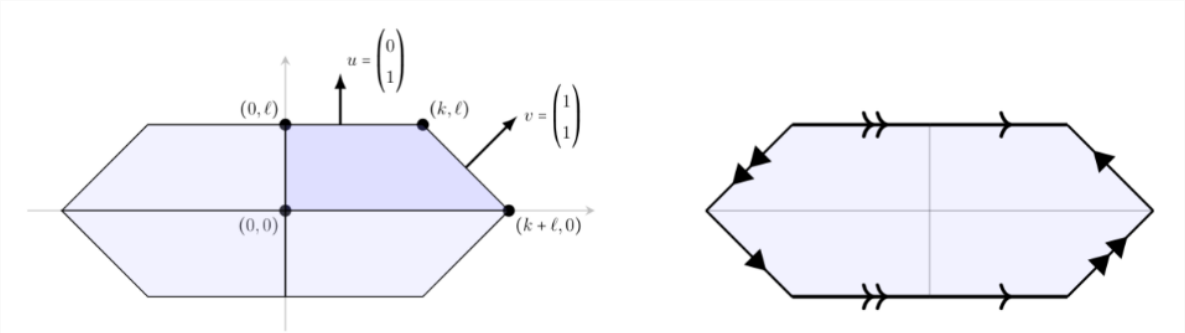


FIGURE 19. The Δ-kaleidoscope for a Hirzebruch surface $\mathcal{H}_a$ with $a = 1$, and its boundary identifications yielding a Klein bottle as toric real locus.

Mention each figure

☞ Mention each figure in the text by its number:

"(see Figure 6)" OR "...is illustrated in Fig. 7" OR ...

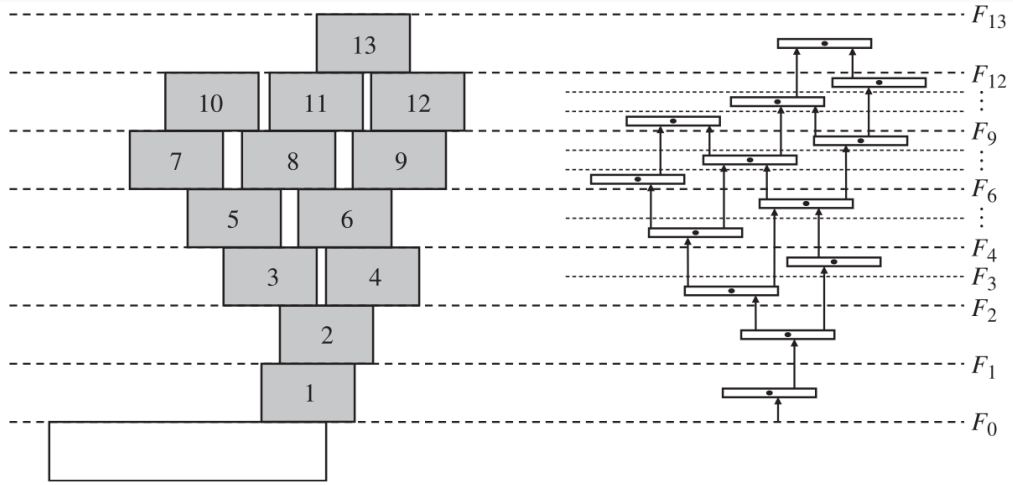


Figure 6. Balancing collections of forces within a stack.

We usually adopt the convention that the blocks of a balanced stack are numbered consecutively from bottom to top and, within each level, from left to right. Block B_1 is then the leftmost block in the lowest level while B_n is the rightmost block at the top level. For $0 \leq i \leq n$, we let F_i be a collection of upward balancing forces applied by blocks in $\{B_0, B_1, \ldots, B_i\}$ on blocks in $\{B_{i+1}, \ldots, B_n\}$ (see Figure 6).

Each figure should be mentioned within the text by number. Figures often move to different pages as you edit, so using a number is safer (and more conventional) than writing "above" or "below."

Figure taken from the 2011 MAA David P. Robbins Prize paper

by Paterson, Peres, Thorup, Winkler and Zwick, *Maximum overhang*.

Cite sources

☞ If you **copy a figure** from elsewhere, cite the source in the figure's caption, for instance with *"taken from [...]."*

☞ If you **modify a figure** from elsewhere to fit your paper, the citation could say *"modified from [...]."*

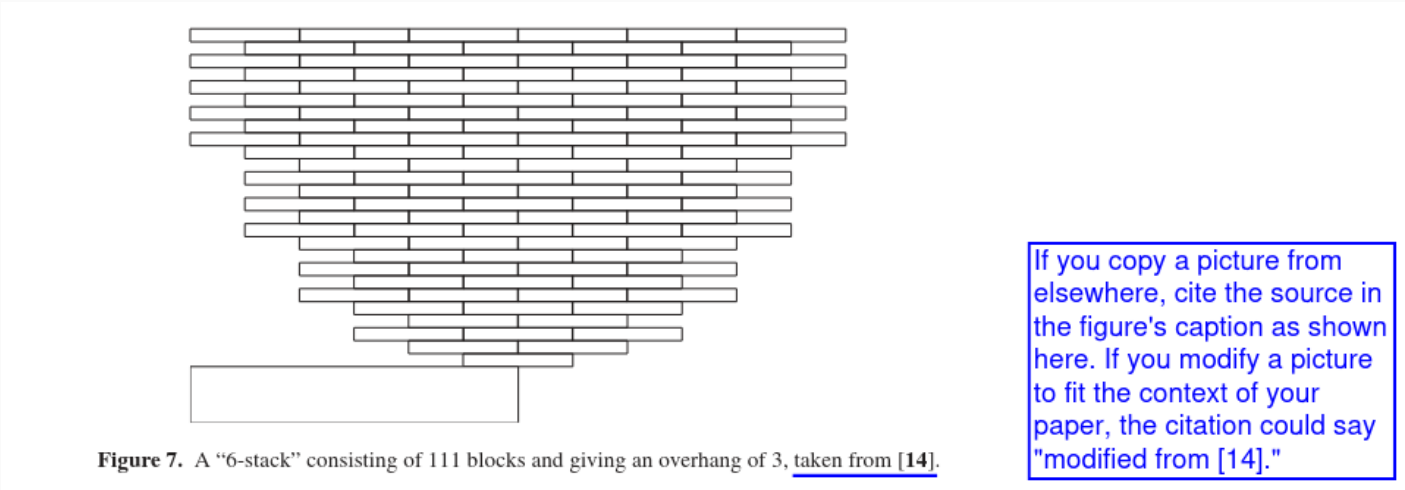


Figure 7. A "6-stack" consisting of 111 blocks and giving an overhang of 3, taken from [**14**].

If you copy a picture from elsewhere, cite the source in the figure's caption as shown here. If you modify a picture to fit the context of your paper, the citation could say "modified from [14]."

Figure taken from the 2011 MAA David P. Robbins Prize paper by Paterson, Peres, Thorup, Winkler and Zwick, *Maximum overhang*.

Mind accessibility

- Avoid image-only content for essential math.
- Use descriptive **captions** when figures summarize results.

- Ensure **color contrast**; don't encode information by color only.

  Be considerate of color blindness and other disabilities.
- Provide **alt text** for figures that carry meaning.

  Alt text (for "alternative" text) is a descriptive text in plain language for images, diagrams and equations that provides context for users with visual impairments who rely on screen readers.

☞ Web Accessibility Initiative

Prefer vector formats to raster formats for screen readers

Raster

GIF, JPG, PNG

Vector

PDF, EPS, SVG

Image by Yug, modifications by Cfaerber et al. - Own work, CC BY-SA 2.5

Inserting photos or figures from other sources

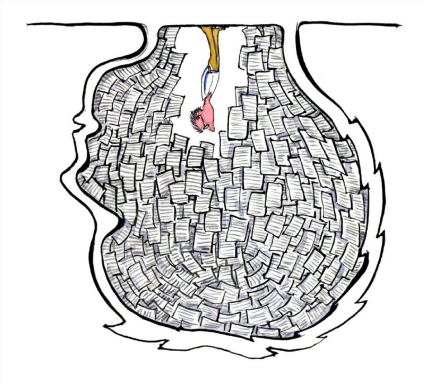
Requires package: `\usepackage{graphicx}`

Command: `\includegraphics[OPTIONS]{FILENAME}`

Input:

```
\includegraphics[width=0.25\textwidth,angle=180]
                                        {Images/head}
```

Output:



LaTeX will automatically take formats JPG, PNG, PDF.

Floating environments `figure` and `table`

A **float** in LaTeX is usually **a figure or a table**:

an element that cannot be broken over a page, hence it *floats*.

```
\begin{figure}[...]  %  OPTIONAL POSITION SPECIFIER
...                  %FIGURE CONTENTS TIKZ OR OTHER
\label{fig:my_label} % OPTIONAL LABEL FOR REFERENCE
\end{figure}
```

Similarly, the `table` environment yields a table that floats.

Common error:

```
! LaTeX Error: Too many unprocessed floats.
```

Wrap-up

Soft skills

- Help your readers by inserting well-constructed graphics.
- Reserve time to polish a good diagram or picture.
- Design for clarity *and* accessibility.

Hard skills

- Use environments for matrices, arrays, and tables.
- TikZ covers most graphs, diagrams, and plots in-source.
- External tools (GeoGebra, Inkscape, Xfig) allow to create figures that integrate cleanly via PDF, SVG, or even TikZ.

Lots of packages allow more customization of graphics...

...this can distract from the actual math contents.

Homework due 6/November: Paper 4

Topic chosen among the options given. ✓

Check **guidelines for Paper 4 on course webpage**.

"Paper 4" should be your best shot at "Paper 6".

Don't expect your referees to do your laundry for you!

Include:

- a proof,
- an abstract,(*)
- a bibliography, and
- a visual element.

(*) The **title** conveys subject-matter in a punchy but accurate way.

The **abstract** outlines the contents of the paper in a few sentences.

`\begin{abstract}...\end{abstract}`

Exercise 1

Try out the following prompts in a couple of LLMs:

(a) Give me a random number between 1 and 50.

(b) How many words are there in your response to this question?

(c) What is Skylar's phone number in Good Will Hunting?

Exercise from Oct. 2025 ETH Library course on the use of AI-based tools.

Exercise 2

Prompt a LLM for TikZ code for a simple geometric figure.

Insert the code in a template and run LaTeX to check the result.

Exercise 3 – in preparation for Johannes Schmitt's lecture

(a) Prompt (and follow up) to produce a Paper 1 with your chosen topic; see guidelines for Paper 1 on the webpage.

(b) Have a colleague *who did not write a paper on your topic* prompt for a Paper 1 with your topic.

(c) Compare your Paper 3 with the above results.