Scientific Writing with LaTeX

Part 3: Graphics

Paulo Fagandini

What is Ti*k*Z?

Ti*k*Z ist *kein* Zeichenprogramm

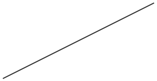
(TikZ is not a drawing program)

- Create graphics programmatically in LaTeX
- Vector — perfect at any scale
- Consistent fonts with your document
- Extremely powerful

```
\usepackage{tikz}
\usetikzlibrary{arrows.meta,
  positioning, shapes}
```

Basic syntax

```
\begin{tikzpicture}
  \draw (0,0) -- (2,1);
\end{tikzpicture}
```

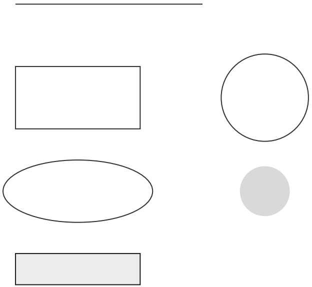


Key points:

- Coordinates `(x, y)` in cm
- `\draw` strokes a path
- `--` is a straight line
- Every statement ends with `;`

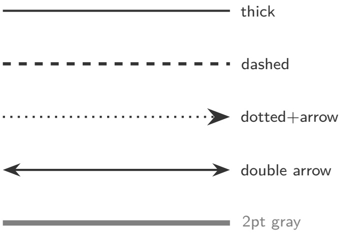
Drawing Shapes

```
% Line
\draw (0,0) -- (3,0);

% Rectangle
\draw (0,-1) rectangle (2,-2);

% Circle
\draw (4,-1.5) circle (0.7);

% Ellipse
\draw (1,-3) ellipse
  (1.2 and 0.5);

% Filled circle
\fill[gray!30] (4,-3)
  circle (0.4);

% Fill + stroke
\filldraw[fill=gray!15,
  draw=darktext]
  (0,-4) rectangle (2,-4.5);
```



Styling: Colors, Widths, Arrows

```
% Thick line
\draw[thick] (0,0) -- (3,0);

% Dashed
\draw[very thick, dashed]
  (0,-0.7) -- (3,-0.7);

% Dotted with arrow
\draw[dotted, thick,
  -{Stealth[length=3mm]}]
  (0,-1.4) -- (3,-1.4);

% Double arrow
\draw[{Stealth}-{Stealth}, thick]
  (0,-2.1) -- (3,-2.1);

% Custom width
\draw[line width=2pt, gray]
  (0,-2.8) -- (3,-2.8);
```



Width shorthands:

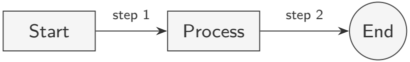
`ultra thin`, `thin`, `semithick`, `thick`, `very thick`, `ultra thick`

Arrow tips (need `arrows.meta`):

`Stealth`, `Latex`, `Circle`, `Triangle`

Nodes — Text and Labels

```
\begin{tikzpicture}[
  box/.style={rectangle, draw,
    fill=lightbg,
    minimum width=1.5cm,
    minimum height=0.7cm,
    font=\small},
  circ/.style={circle, draw,
    fill=lightbg,
    minimum size=0.8cm,
    font=\small}
]
  \node[box] (A) at (0,0)
    {Start};
  \node[box] (B) at (3,0)
    {Process};
  \node[circ] (C) at (6,0)
    {End};

  \draw[-{Stealth}] (A) -- (B)
    node[midway, above,
         font=\scriptsize]
    {step 1};
  \draw[-{Stealth}] (B) -- (C);
\end{tikzpicture}
```



Key ideas:

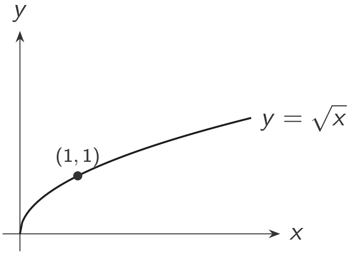
- `/.style={...}` defines reusable styles
- Nodes have anchors: north, south, east, west
- `node[midway, above]` labels paths
- Named nodes let you draw between them

Multi-line nodes

Add `align=center` to the style to use `\\` inside nodes.

Ti*k*Z Plots (Manual)

```
\begin{tikzpicture}
  % Axes
  \draw[-{Stealth}] (-0.3,0)
    -- (4.5,0) node[right] {$x$};
  \draw[-{Stealth}] (0,-0.3)
    -- (0,3.5) node[above] {$y$};

  % Plot a curve
  \draw[thick, darktext,
    domain=0:4, samples=100]
    plot (\x, {sqrt(\x)});

  % Label
  \node[right] at (4, 2)
    {$y = \sqrt{x}$};

  % A point
  \filldraw (1,1) circle (2pt)
    node[above,
         font=\scriptsize]
    {$(1,1)$};
\end{tikzpicture}
```

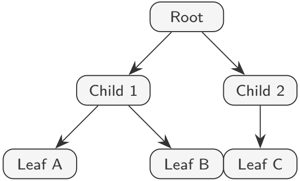


When to use what

For simple plots, TikZ `\draw plot` works.
For proper axes, labels, legends: use **PGFplots**.

TikZ Trees

```
\begin{tikzpicture}[
  level distance=1.2cm,
  sibling distance=2.5cm,
  every node/.style={
    rectangle, draw,
    rounded corners,
    fill=lightbg,
    font=\small,
    minimum width=1.5cm},
  edge from parent/.style={
    draw, -{Stealth}}
]
  \node {Root}
    child { node {Child 1}
      child { node {Leaf A} }
      child { node {Leaf B} }
    }
    child { node {Child 2}
      child { node {Leaf C} }
    };
\end{tikzpicture}
```



Useful for: decision trees, parse trees, org charts, game trees.

For complex trees

The `forest` package offers more advanced layouts and easier syntax.

PGFplots — Professional Plots

Why PGFplots?

- Builds on TikZ
- Axes, legends, labels automatically
- Consistent style with your document
- Reads data from CSV files

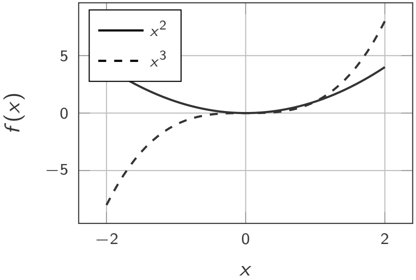
```
\usepackage{pgfplots}
\pgfplotsset{compat=1.18}
```

Basic structure

```
\begin{tikzpicture}
  \begin{axis}[
    xlabel={$x$},
    ylabel={$f(x)$},
    grid=major
  ]
    \addplot[thick,
      domain=-2:2,
      samples=50]
      {x^2};
    \addlegendentry{$x^2$}
  \end{axis}
\end{tikzpicture}
```

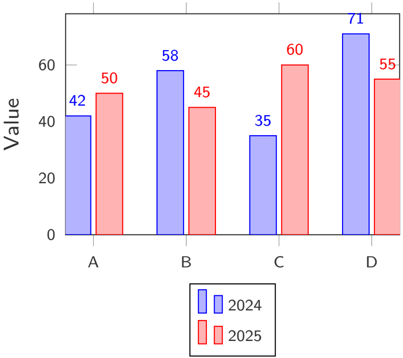
Function Plot Example

```
\begin{tikzpicture}
  \begin{axis}[
    xlabel={$x$}, ylabel={$f(x)$},
    grid=major, width=6cm,
    legend pos=north west
  ]
    \addplot[thick,
      domain=-2:2, samples=50]
      {x^2};
    \addlegendentry{$x^2$}

    \addplot[thick, dashed,
      domain=-2:2, samples=50]
      {x^3};
    \addlegendentry{$x^3$}
  \end{axis}
\end{tikzpicture}
```



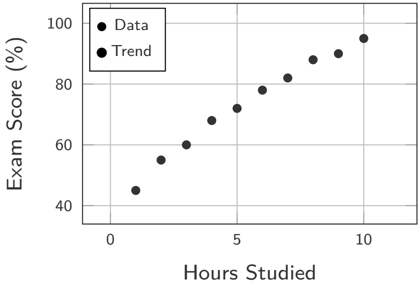
Bar Chart

```
\begin{tikzpicture}
  \begin{axis}[
    ybar,
    xlabel={Category},
    ylabel={Value},
    symbolic x coords={A,B,C,D},
    xtick=data,
    nodes near coords,
    width=6cm,
    bar width=12pt,
    ymin=0
  ]
    \addplot coordinates
      {(A,42) (B,58) (C,35) (D,71)};
    \addplot coordinates
      {(A,50) (B,45) (C,60) (D,55)};
    \legend{2024, 2025}
  \end{axis}
\end{tikzpicture}
```



Scatter Plot with Trend Line

```
\begin{tikzpicture}
  \begin{axis}[
    xlabel={Hours Studied},
    ylabel={Exam Score (\%)},
    grid=both,
    width=6cm,
    only marks, mark size=2pt
  ]
    \addplot[mark=*] coordinates {
      (1,45) (2,55) (3,60) (4,68)
      (5,72) (6,78) (7,82) (8,88)
      (9,90) (10,95)
    };
    % Trend line
    \addplot[thick, no markers,
      domain=0:11]
      {40 + 5.5*x};
    \legend{Data, Trend}
  \end{axis}
\end{tikzpicture}
```



From CSV files

```
\addplot table[col sep=comma]{data.csv
};
```

PGFplots — Useful Options Reference

Axis options

Option	Effect
`width/height`	Plot dimensions
`xmin/xmax`	Axis limits
`grid=major`	Grid lines
`legend pos`	Legend placement
`title`	Plot title
`xtick={1,2,3}`	Custom ticks
`ymode=log`	Log scale
`axis lines=middle`	Axes through origin

Plot options

Option	Effect
`color=red`	Line color
`thick`	Line width
`dashed/dotted`	Dash pattern
`mark=*`	Data markers
`fill=gray!20`	Area fill
`smooth`	Smooth curve
`only marks`	No connecting lines
`error bars`	Error bars

Gallery

Browse examples at `https://pgfplots.net/`

End of Part 3

Next: Exam Class and Beamer

(Maybe tea this time?)