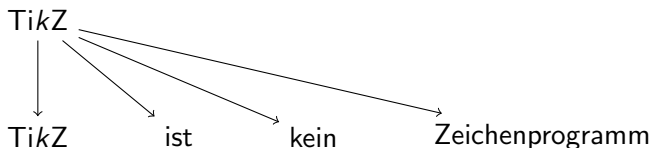


# TikZ an introduction

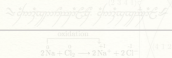
Dominic Branford

January 25, 2017

# TikZ



- ▶ TikZ is not a drawing program  $\Rightarrow$  forget about WYSIWYG.
- ▶ TikZ generates vector graphics directly into PDFs straight from code.
- ▶ “Programmed images”  $\Rightarrow$  easier to automate, simpler to make significant changes



# TeX

[Questions](#) [Tags](#) [Users](#) [Badges](#) [Unanswered](#)[Ask Question](#)

## Tags

[popular](#) [name](#) [new](#)

A tag is a keyword or label that categorizes your question with other, similar questions. Using the right tags makes it easier for others to find and answer your question.

Type to find tags: **[tikz-pgf]** × 15023

a higher-level drawing language built on top of the PGF graphics framework. For questions specifically about the PGF layer use [pgf-core](#)

9 asked today, 58 this week

**[tables]** × 7745

about the tabular environment and related packages such as [array](#), [booktabs](#), [tblrn](#), [tblort](#) and [longtable](#). For

33 asked this week, 159 this month

**[beamer]** × 5746

a document class for creating presentations and slides. For general questions about presentations or slides, independent of the

18 asked this week, 105 this month

**[math-mode]** × 4952

about typesetting mathematical content, e.g. tweaking the appearance of spacing and symbols in a formula, or producing specific

15 asked this week, 79 this month

**[pgfplots]** × 4785

a package for creating 2D and 3D plots of mathematical functions and numerical data, using the PGF graphics framework. For

17 asked this week, 81 this month

**[fonts]** × 4738

about fonts and how to use them. It deals with questions about changing family, shape and weight of fonts, but also with font packages. If

13 asked this week, 55 this month

**[spacing]** × 4350

about both horizontal and vertical white space, e.g. space between letters or words, or space between paragraphs. For adjusting the space

18 asked this week, 54 this month

**[macros]** × 4207

for questions specifically concerning TeX's macro processor or Note that, because TeX uses macros ubiquitously, most questions

13 asked this week, 45 this month

**[biblatex]** × 4082

a complete reimplemention of the bibliographic facilities provided by LaTeX. It has its own backend ([biber](#)) which can be

6 asked today, 16 this week

**[graphics]** × 4002

about inclusion of external graphic files in your document. For questions about vector graphics created programmatically inside a

9 asked this week, 63 this month

**[table-of-contents]** × 3638

about the creation or modification of the Table of Contents or similar listings like List of Figures and List of Tables as well as custom

10 asked this week, 42 this month

**[bibtex]** × 3419

specifically about Oren Patashnik's BibTeX program, its [.bst](#) style file and [.bib](#) database file formats, and use of these in creating

9 asked this week, 45 this month

**[floats]** × 3408

about the floating environments figure, table and other, custom-defined floats, and concerns issues such as positioning,

25 asked this week, 67 this month

**[bibliographies]** × 3346

about creating lists of publications and citing these in a document. If possible, replace this tag with the specific bibliography package you

16 asked this week, 60 this month

**[sectioning]** × 3150

about commands like `\chapter` or `\section` forming the logical structure of documents. For questions specifically about `(parts)`.

8 asked this week, 38 this month

**[horizontal-alignment]** × 3043

about aligning document elements horizontally, e.g. typesetting paragraphs using `\centering` or `\raggedright` or adjusting the

8 asked this week, 39 this month

**[errors]** × 2891

for questions regarding or including compilation or similar errors.

**[xetex]** × 2879

a Unicode-capable variant of the TeX engine. Use this tag if your question is about Xe(La)TeX specific problems and is not

**[formatting]** × 2714

for general questions about formatting document elements. Use this tag in addition to other tags specifying what should be

**[equations]** × 2707

about the various environments to typeset displaystyle mathematics, e.g. `equation` and `align`. For general questions about

## Resources and Examples

- ▶ PGF Manual (CTAN)
- ▶ A very minimal introduction to TikZ - Jacques Crémer
- ▶ A TikZ tutorial: Generating graphics in the spirit of T<sub>E</sub>X - Andrew Mertz and William Slough (TUGboat **30** 2)
- ▶ L<sup>A</sup>T<sub>E</sub>X and Friends - Marc van Dongen
- ▶ L<sup>A</sup>T<sub>E</sub>X Cookbook - Stefan Kottwitz
- ▶ The tikz-pgf tag on <https://tex.stackexchange.com>
- ▶ <https://texample.net/tikz>
- ▶ <https://PGFPlots.net>

# Using TikZ

- ▶ Necessary setup

```
\usepackage{tikz}
```

- ▶ Introduces a new command and environment

- ▶ The tikz macro

```
\tikz
```

- ▶ The tikzpicture environment

```
\begin{tikzpicture}
```

```
<TikZ code>
```

```
\end{tikzpicture}
```

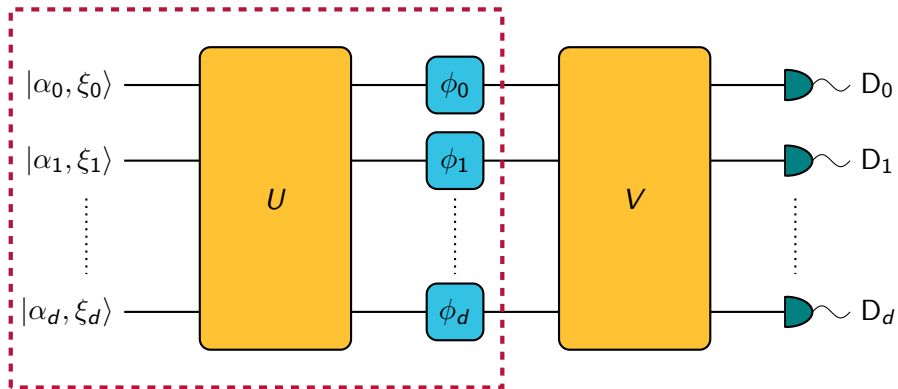
- ▶ Loads pgf

# pgf

## Under the hood

- ▶ `pgf` is the lower-level system upon which `TikZ` is based.
- ▶ `pgfplots` is also based on `pgf` but without requiring `TikZ`.
- ▶ The `pgf` bundle includes other  $\LaTeX$  features such as a key-val interface (`pgfkeys`), loops (`pgffor`), and mathematical libraries (`pgfmath`).

# Diagrams

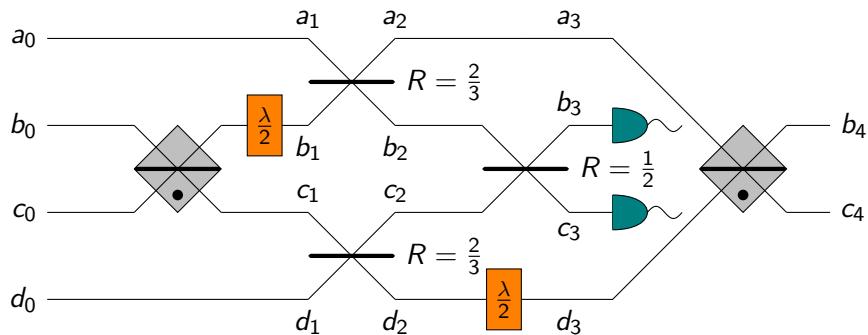


# The (not particularly intelligent) code

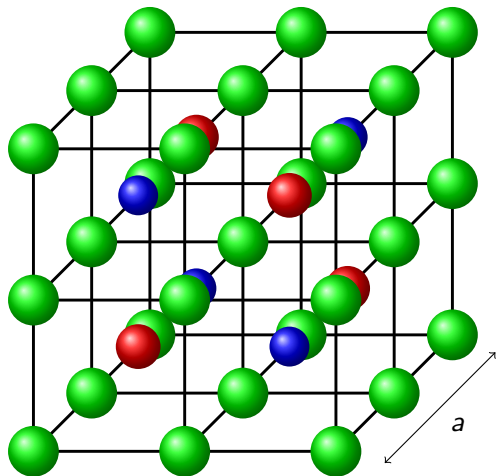
```
\foreach \i / \ilbl in { 0 / d, 2 / 1, 3 / 0 }
{
\node [left] at (0,\i) {\$\ket{\alpha_{\ilbl},\xi_{\ilbl}}$};
\draw [thick] (0,\i) -- (1,\i);
\draw [thick] (4.75,\i) -- (5.75,\i);
\draw [thick] (3,\i) -- (4,\i);
\draw [rounded corners, thick, fill=\phasecolour] (4,\i-0.375) rectangle (4.75,
\node at (4.375,\i) {\$\phi_{\ilbl}$};
\draw [thick] (7.75,\i) -- (8.75,\i);
\draw [thick] (8.74,\i-0.21) -- (8.74,\i+0.21);
\draw [thick, domain=0:1, fill=\detectorcolour] plot ({8.75+0.35*\sin(180*\x)},{
\draw [domain=0:1] plot ({9.1+0.5*\x},{\i+0.08*\sin(360*\x)});
\node [right] at (9.6,\i) {\$\text{D}_{\ilbl}$};
}
\draw [rounded corners, thick, fill=\ucolour] (1,-0.5) rectangle (3,3.5);
\draw [rounded corners, thick, fill=\vcolour] (5.75,-0.5) rectangle (7.75,3.5);
\draw [thick, dotted] (-0.5,1.5) -- (-0.5,0.5);
\draw [thick, dotted] (4.375,1.5) -- (4.375,0.5);
\draw [thick, dotted] (8.875,1.5) -- (8.875,0.5);
\node at (2,1.5) {\$U$};
\node at (6.75,1.5) {\$V$};
\draw [dashed, warwickrubyred, ultra thick] (-1.5,-1) rectangle (5,4);
```



# Optics



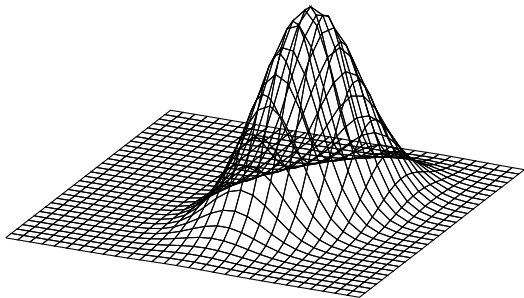
# Atomic Lattice



Source: Nelson Yeung

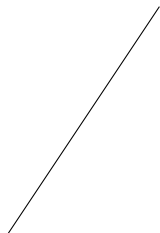
## pgfplots

```
\begin{tikzpicture}
  \begin{axis}[xmin=-4,xmax=4,ymin=-4,ymax=4,hide axis]
    \addplot3[mesh,draw=black,samples=30,domain=-4:4] %
      {0.15*exp(-0.9*x^2 + 1.8*x %
        + 0.5*x*y - 0.5*y - 0.4*y*y)};
  \end{axis}
\end{tikzpicture}
```



## draw

```
\begin{tikzpicture}  
  \draw (0,0) -- (2,3);  
\end{tikzpicture}
```

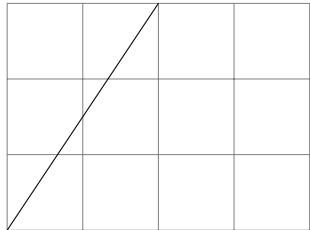


```
\begin{tikzpicture}  
  \path (0,0) -- (2,3);  
  \path [draw] (1,2) -- (2,1);  
\end{tikzpicture}
```



## help lines

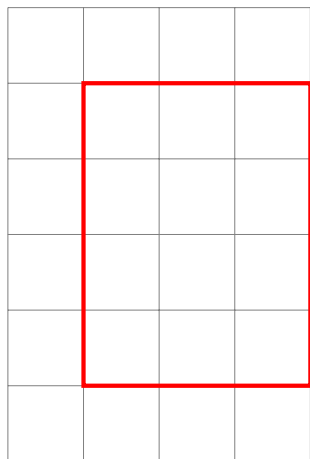
```
\begin{tikzpicture}
  \draw [help lines] (0,0) grid (4,3);
  \draw (0,0) -- (2,3);
\end{tikzpicture}
```



- ▶ `--` produces an edge
- ▶ `grid` produces a grid
- ▶ `rectangle` produces a rectangle
- ▶ `[help lines]` option to the `\draw` command reduces opacity of the grid

## Drawing a rectangle

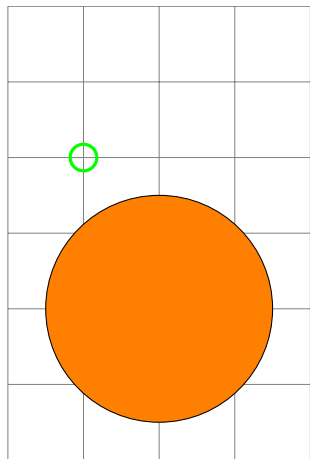
```
\draw [red,ultra thick] (1,1) rectangle (4,5);
```



- ▶ `\draw [foo] options foo` are keyval options used to draw the path
- ▶ Thickness
  - ▶ Natural language `thick`, `very thick`, `ultra thick`, `thin`, `very thin`
  - ▶ Precise dimensions  
`line width=<dimension>`
- ▶ `red` colours can also be used directly

## Drawing a circle

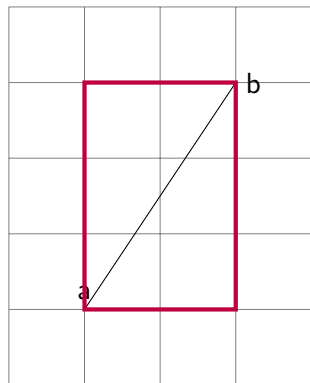
```
\draw [fill=orange] (2,2) circle (1.5cm);  
\draw [green,very thick] (1,4) circle (5pt);
```



- ▶ circle takes an origin and a radius as argument
- ▶ fill option specifies a colour to be used for the interior of a path

## Using a coordinate

```
\coordinate [label=a] (one) at (1,1);  
\coordinate [label=right:b] (two) at (3,4);  
\draw (one) -- (two);  
\draw [purple,ultra thick] (one) rectangle (two);
```



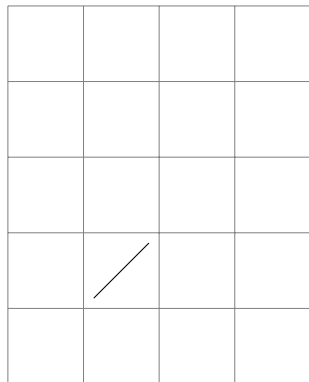
- ▶ `\coordinate (<name>) at (<coord>)`
  - ▶ Creates coordinate named `<name>` at `<coord>` which can be re-used
  - ▶ Enables relative positioning  $\Rightarrow$  a complicated image can be defined by only a few points which receive an explicit coordinate
- ▶ `label=a` produces label above  
`label=right:a`



# Using a node

A coordinate with size

```
\node (a) at (1,1) {};  
\node (b) at (2,2) {};  
\draw (a) -- (b);
```

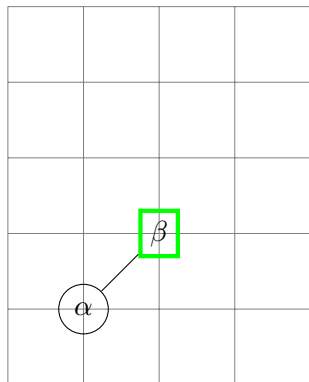


- ▶ A node can have things inside it, the contents of the `{}`
- ▶ By default it has some size to it

# Using a node

A coordinate with size

```
\node [circle,draw] (a) at (1,1) {$\alpha$};  
\node [rectangle,draw=green,ultra thick]%  
  (b) at (2,2) {$\beta$};  
\draw (a) -- (b);
```



- ▶ A node can have things inside it, the contents of the `{}`
- ▶ By default it has some size to it

# The positioning library

```
\documentclass[tikz]{standalone}
\usepackage{tikz}
\usetikzlibrary{positioning}
\begin{document}
\begin{tikzpicture}
  \node (tikzroot) {Ti\textit{k}Z};
  \node (tikz) [below={of tikzroot}] {Ti\textit{k}Z};
  \node (ist) [right={of tikz}] {ist};
  \node (kein) [right={of ist}] {kein};
  \node (zeichen) [right={of kein}] {Zeichenprogramm};
\end{tikzpicture}
\end{document}
```

TikZ

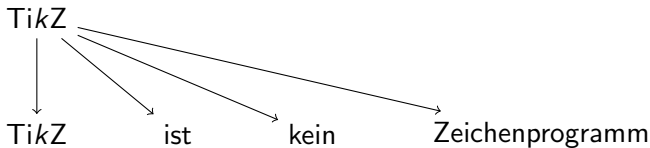
TikZ

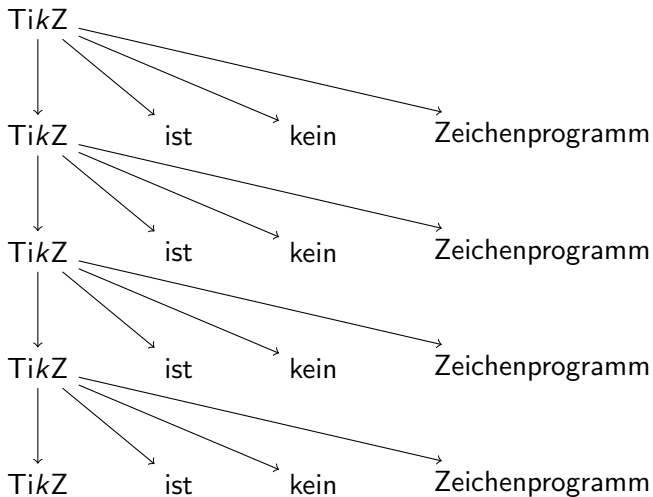
ist

kein

Zeichenprogramm

```
\begin{tikzpicture}
  \node (tikzroot) {\TikZ};
  \node (tikz) [below={of tikzroot}] {\TikZ};
  \node (ist) [right={of tikz}] {ist};
  \node (kein) [right={of ist}] {kein};
  \node (zeichen) [right={of kein}] {Zeichenprogramm};
  \draw [->] (tikzroot) -- (tikz);
  \draw [->] (tikzroot) -- (ist);
  \draw [->] (tikzroot) -- (kein);
  \draw [->] (tikzroot) -- (zeichen);
\end{tikzpicture}
```

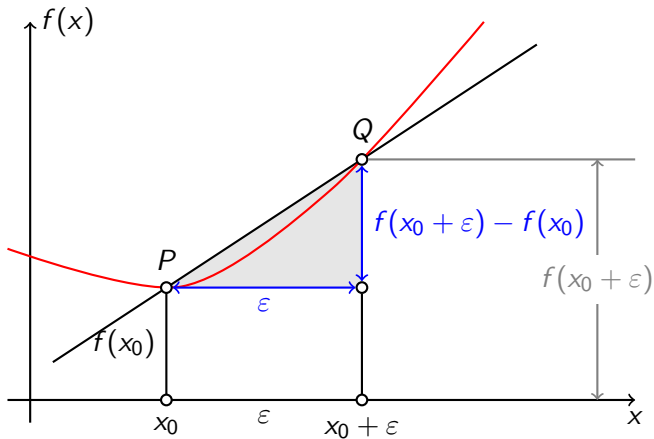




```

\begin{tikzpicture}
\node (tikz-0) at (0,0) {\TikZ};
\foreach [evaluate=\i as \j using {int(\i-1)}] \i in {%
  1,...,4}
{
  \node (tikz-\i) [below={of tikz-\j}] {\TikZ};
  \node (ist-\i) [right={of tikz-\i}] {ist};
  \node (kein-\i) [right={of ist-\i}] {kein};
  \node (zeichen-\i) [right={of kein-\i}] {Zeichenprogramm};
  \draw [->] (tikz-\j) -- (tikz-\i);
  \draw [->] (tikz-\j) -- (ist-\i);
  \draw [->] (tikz-\j) -- (kein-\i);
  \draw [->] (tikz-\j) -- (zeichen-\i);
}
\end{tikzpicture}

```



Source: Henri Menke

<http://tex.stackexchange.com/a/168307/>

Three coordinates on the axis and two paths are defined with absolute coordinates, everything else is constructed using relative coordinates.

```

\begin{tikzpicture}[thick,
  dot/.style = {draw,fill=white,circle,inner sep=0pt,minimum size=4pt}
]
\coordinate (0) at (0,0);
\draw[->] (-0.3,0) -- (8,0) coordinate[label={below:$x$}] (xmax);
\draw[->] (0,-0.3) -- (0,5) coordinate[label={right:$f(x)$}] (ymax);
\path[name path=x] (0.3,0.5) -- (6.7,4.7);
\path[name path=y] plot[smooth] coordinates {(-0.3,2) (2,1.5) (4,2.8) (6,5)};
\begin{scope}[name intersections={of=x and y,name=i}]
  \fill[gray!20] (i-1) -- (i-2 |- i-1) -- (i-2) -- cycle;
  \draw (0.3,0.5) -- (6.7,4.7);
  \draw[red] plot[smooth] coordinates {(-0.3,2) (2,1.5) (4,2.8) (6,5)};
  \draw (i-1) node[dot,label={above:$P$}] (i-1) {} -- node[left] {$f(x_0)$} (i-1);
  \path (i-2) node[dot,label={above:$Q$}] (i-2) {} -- (i-2 |- i-1) node[dot] (i-1);
  \draw (i-12) -- (i-12 |- 0) node[dot,label={below:$x_0 + \epsilon$}] {};
  \draw[blue,<->] (i-2) -- node[right] {$f(x_0 + \epsilon) - f(x_0)$} (i-12);
  \draw[blue,<->] (i-1) -- node[below] {$\epsilon$} (i-12);
  \path (i-1 |- 0) -- node[below] {$\epsilon$} (i-2 |- 0);
  \draw[gray] (i-2) -- (i-2 -| xmax);
  \draw[gray,<->] ([xshift=-0.5cm]i-2 -| xmax) -- node[fill=white] {$f(x_0 + \epsilon)$} (i-1);
\end{scope}
\end{tikzpicture}

```

Source: Henri Menke <http://tex.stackexchange.com/a/168307/>

Three coordinates on the axis and two paths are defined with absolute coordinates, everything else is constructed using relative coordinates.