

# Grafiken mit TikZ

## LaTeX-Kurs der Unix-AG

Klemens Schmitt

17.06.2015



**TikZ**

**Libraries**

**Pakete aufbauend auf TikZ**

**Zusammenfassung und Ausblick**

## TikZ

Einbindung

Linien

Formen und Füllungen

Farben

Hilfslinien und Zuschneiden

Stile und Punkte

Flussdiagramme und Beschriftungen

Auf andere Grafiken zugreifen

Schleifen und Verzweigungen

## Libraries

## Pakete aufbauend auf TikZ

## Zusammenfassung und Ausblick

## TikZ

- ▶ Bisher: externe Grafiken einbinden
  - ▶ `\includegraphics`-Befehl
- ▶ TikZ - TikZ ist kein Zeichenprogramm
  - ▶ Grafiken in  $\LaTeX$  programmieren

## Alternativen und Hilfen

- ▶ PSTricks (TikZ für Postscript)
- ▶ externe Grafikprogramme mit TikZ-Export (Inkscape, Matlab, ...)
- ▶ WYSIWYG-Editoren für TikZ (TikZEdt, QTikZ, ...)

## Vorteile

- ▶ Vektorgrafik
- ▶ genaue Kontrolle über Positionierung
- ▶ programmierbar, spart Arbeit
- ▶ fertige Befehle/Bausteine für „technische“ Zeichnungen
- ▶ globale Stildefinitionen
- ▶ gleiche Schriftart wie Dokument

## Nachteile

- ▶ steile Lernkurve
- ▶ schlecht für Fotos, Bildmanipulation
- ▶ lange Kompilierzeit

### Prämbel

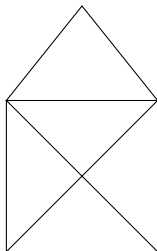
- ▶ `\usepackage{tikz}`
- ▶ `\usetikzlibrary{...}`

### Im Dokument

- ▶ `\tikz <Befehle>` oder
- ▶ `\begin{tikzpicture}<Befehle>\end{tikzpicture}`
- ▶ Befehle enden immer auf ;

- ▶ `\draw` erzeugt Linien:

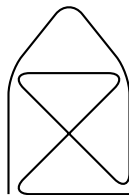
```
1 \begin{tikzpicture}
2   \draw (0,0) -- (0,2) -- (1,3.25) -- (2,2) -- (2,0) --
      (0,2) -- (2,2) -- (0,0) -- (2,0);
3 \end{tikzpicture}
```



### Optionen

- ▶ entweder `\begin{tikzpicture}[<Optionen>]`
- ▶ oder `\draw[<Optionen>]`

```
1 \begin{tikzpicture}[thick,rounded corners=8pt,scale=.8]
2   \draw (0,0) -- (0,2) -- (1,3.25) -- (2,2) -- (2,0) --
   (0,2) -- (2,2) -- (0,0) -- (2,0);
3 \end{tikzpicture}
```





### Formen

- ▶ normale Linie `--`
- ▶ nur rechtwinklige Wege `-|` bzw `|-`
- ▶ Kreisbögen `arc (<start>:<end>:<radius>)` oder  
`arc [start angle=<start>, end angle=<end>, radius=<radius>]`
- ▶ `cos`, `sin`
- ▶ Parabeln `(<Startpunkt>)parabola bend (<Scheitel>)(<Endpunkt>)`
- ▶ Bézier `(<Punkt>).. controls (<Punkt>)and (<Punkt>).. (<Punkt>)`

### Stil

- ▶ `solid`, `dotted`, `dashed`

# TikZ

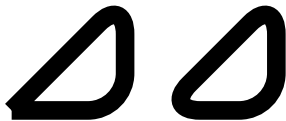
## Linien

```
1 \begin{tikzpicture}[ultra thick]
2   \draw (0,0) .. controls (1,1) and (2,1) .. (2,0);
3   \draw [dashed] (0,0) -- (1,1) (2,1) -- (2,0);
4   \draw (3,0) -| (4,1);
5   \draw [dotted] (3,0) -- (4,1);
6 \end{tikzpicture}
```



- ▶ geschlossene Kurven mit `cycle`

```
1 \begin{tikzpicture}[line width=7pt,rounded corners=14pt]
2   \draw (0,0) -- (1.5,0) -- (1.5,1.5) -- (0,0);
3   \draw (2,0) -- (3.5,0) -- (3.5,1.5) -- cycle;
4 \end{tikzpicture}
```



# TikZ

## Formen und Füllungen

```
1 \begin{tikzpicture}[line width=3pt]
2   \fill (0,0) rectangle (1,1);
3   \filldraw [fill=red,draw=blue] (2,0) rectangle (3,1);
4   \shade (4.5,.5) ellipse (1cm and .5cm);
5   \shadedraw [left color=green,right color=blue] (6.5,.5)
   circle (.5cm);
6   \shade [ball color=red] (8.5,.5) circle (.5cm);
7 \end{tikzpicture}
```



### Farben

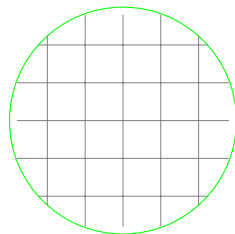
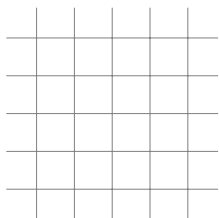
- ▶ Paket `xcolor` (auch ohne TikZ, `\color...`)
- ▶ vordefinierte Farben: `red`, `blue`, ...
- ▶ Farbe definieren `\definecolor{meinefarbe}{<model>}{<Werte>}`
  - ▶ `rgb/RGB`
  - ▶ `cymk`
  - ▶ `HTML`
- ▶ Mischen mit `<Farbe1>!<Prozentwert>!<Farbe2>`

```
\color{gray}{...}  
  
\color[wave]{532}{...}  
  
\definecolor{meinblau}{RGB}{0,0,127}  
  
\color{yellow!50!red}{...}
```

# TikZ

## Hilfslinien und Zuschneiden

```
1 \begin{tikzpicture}
2   \draw[step=.5cm,gray,very thin] (-1.4,-1.4) grid
   (1.4,1.4);
3   \begin{scope}[green]
4     \clip [draw] (3,0) circle (1.5cm);
5     \draw[step=.5cm,gray,very thin] (4.4,-1.4) grid
   (1.6,1.4);
6   \end{scope}
7 \end{tikzpicture}
```



### Stile definieren

- ▶ mystyle als Zusammenfassung mehrerer Eigenschaften:
- ▶ `\tikzstyle{mystyle}=[<Stil>]` (ausserhalb der Umgebung)
- ▶ oder `\begin{tikzpicture}[mystyle/.style={<Stil>}]`

### Punkte definieren

- ▶ Punkte mit Namen versehen:
- ▶ `\coordinate (<Name>)at (<Koordinaten>)`
- ▶ oder `\draw (<Punkt>)coordinate (<Name>)...;`

# TikZ

## Stile und Punkte

```
1 \tikzstyle{mystyle}=[blue,line width=10pt]
2 \begin{tikzpicture}[mystyle2/.style={line join=bevel}]
3   \coordinate (Ecke) at (0,1);
4   \draw [mystyle,mystyle2] (0,0) -- (Ecke) -- (1,1);
5 \end{tikzpicture}
```





### Nodes

- ▶ Beschriftungen mit `\node at (<Punkt>)[<Optionen>] {<Text>};`
- ▶ Position `above,below,left,right`
- ▶ Position auf Linien `at end/start, (very)near end/start, midway`

```
1 \begin{tikzpicture}
2   \fill [gray] (0,0) circle (.1cm);
3   \node at (0,0) [rotate=30] {Text};
4   \fill [gray] (2,0) circle (.1cm) node [left] {\LaTeX};
5   \draw (3,0) -- (4,0) node [midway,above] {Linie};
6 \end{tikzpicture}
```

Text

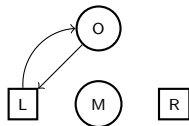
L<sup>A</sup>T<sub>E</sub>X ●

Linie

# TikZ

## Flussdiagramme und Beschriftungen

```
1 \tikzstyle{mycircle}=[circle,draw,thick]
2 \tikzstyle{myrect}=[rectangle,draw,thick]
3 \begin{tikzpicture}
4   \node[mycircle] (oben) {\tiny O};
5   \node[mycircle] (mitte) [below of=oben] {\tiny M};
6   \node[myrect] (rechts) [right of=mitte] {\tiny R};
7   \node[myrect] (links) [left of=mitte] {\tiny L}
8   edge [->,bend left=45] (oben)
9   edge [->] (mitte);
10 \end{tikzpicture}
```





# TikZ

## Auf andere Grafiken zugreifen

- ▶ Bildposition ausserhalb des Bildes merken:
  - ▶ `remember picture`
- ▶ Bilder ohne eigenen Platzbedarf:
  - ▶ `overlay`

```
1 Ein roter Punkt \tikz[remember picture] \node [circle,fill
   =red] (node1) {}; im Text\\
2 und ein blaues Rechteck \tikz[remember picture] \node[fill
   =blue] (node2) {};\n
3 und ein Pfeil \tikz[remember picture,overlay] \draw[->,
   very thick,gray] (node1) to [bend right =30] (node2);
```




Ein roter Punkt  im Text  
und ein blaues Rechteck 

# TikZ

## Auf andere Grafiken zugreifen

- ▶ Bildposition ausserhalb des Bildes merken:
  - ▶ `remember picture`
- ▶ Bilder ohne eigenen Platzbedarf:
  - ▶ `overlay`

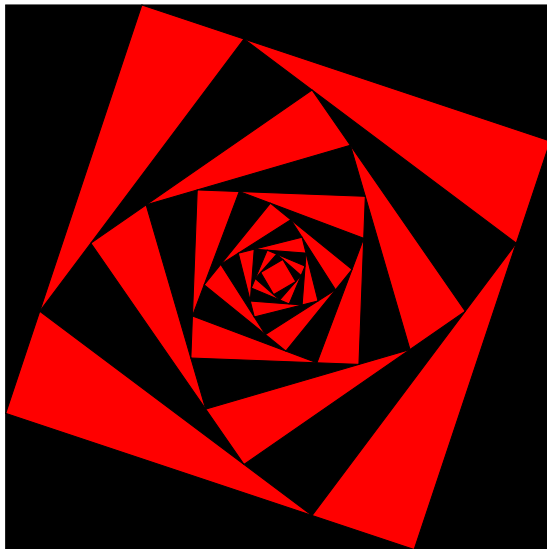
```
1 Ein roter Punkt \tikz[remember picture] \node [circle,fill
   =red] (node1) {}; im Text\\
2 und ein blaues Rechteck \tikz[remember picture] \node[fill
   =blue] (node2) {};\n
3 und ein Pfeil \tikz[remember picture,overlay] \draw[->,
   very thick,gray] (node1) to [bend right =30] (node2);
```

Ein roter Punkt  im Text  
und ein blaues Rechteck   
und ein Pfeil 

### Hintergrundbilder

- ▶ `page.<Ort>` als Eckpunkte der Seite

```
1 \begin{tikzpicture}[remember picture,overlay]
2   \draw [line width=1mm,opacity=.25] (current page.north
      east) -- (current page.south west);
3   \draw [line width=1mm,opacity=.25] (current page.center)
      circle (3cm);
4 \end{tikzpicture}
```



▶ `\usepackage{ifthen}`

```
1 \begin{tikzpicture}[scale=0.6]
2   \path (0,0) coordinate (A) (12,0) coordinate (B)
3     (12,12) coordinate (C) (0,12) coordinate (D);
4   \foreach \i in {1,...,14}{
5     \ifthenelse{\isodd{\i}}{\def\farbe{black}}{\def\farbe{
6       red}}
7     \draw[fill=\farbe] (A)--(B)--(C)--(D)--cycle;
8     \path (A) coordinate (TMP);
9     \path (A)--(B) coordinate[near end] (A)
10      --(C) coordinate[near end] (B)
11      --(D) coordinate[near end] (C)
12      --(TMP) coordinate[near end] (D);
13   }
14 \end{tikzpicture}
```

## TikZ

### Libraries

arrows, decorations

calc

intersections

shapes

spy

external

weitere

## Pakete aufbauend auf TikZ

## Zusammenfassung und Ausblick



## Bibliotheken

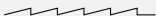
- ▶ erweitern Funktionsumfang
- ▶ einbinden mit `\usetikzlibrary{...}`
- ▶ falls Beispiele nicht kompilieren, eventuell Libraries einbinden

# Libraries

## arrows, decorations

### Dekorationen


#### ▶ Verzierungen von Linien


▶ saw 

▶ zigzag 

▶ snake 

▶ coil 


▶ brace 

▶ expading waves 

### Pfeilspitzen

▶ als Option von Linien [ $\leftarrow$ - $\rightarrow$ ]

▶ Form globaler festlegen [ $\rightarrow$ =\*]


▶ \* 

▶ o 

▶ ) 

▶ angle 45 

▶ triangle 60 

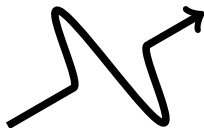
▶ diamond 

# Libraries

## arrows, decorations

```
1 \usetikzlibrary{decorations,decorations.text}
2 \begin{tikzpicture}[line width=.1cm]
3   \draw [decorate,decoration={text along path, text={
4     angepasster Text},text align={fit to path stretching
5     spaces}}] (0,0) arc (120:60:3);
6   \draw [->,rotate around={30:(5,0)},decorate,decoration={
7     snake,amplitude=1cm,segment length=1cm,post length=.5
8     cm,pre length=1cm}](5,0) -- (8,0);
9 \end{tikzpicture}
```

angepasster  
Text



### Koordinatensysteme

- ▶ Kartesisch ( $\langle x \rangle, \langle y \rangle, \langle z \rangle$ )
- ▶ Polar ( $\langle \text{Winkel} \rangle : \langle \text{Länge} \rangle$ )
- ▶ Koordinatenangaben
  - ▶ Absolut ( $\langle x \rangle, \langle y \rangle$ )
  - ▶ Relativ + ( $\langle x \rangle, \langle y \rangle$ )
  - ▶ Relativ (setzt Bezugspunkt neu) ++ ( $\langle x \rangle, \langle y \rangle$ )
  - ▶ Relativer Winkel ( $[\text{turn}] \langle \text{Winkel} \rangle : \langle \text{Länge} \rangle$ )

### Calc-Library

- ▶ Verschiedene Koordinatenangaben mischen: ( $\$ \dots \$$ )

# Libraries

## calc

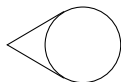
```
1 \usetikzlibrary{calc}
2 \begin{tikzpicture}[thick]
3   \draw[help lines] (0,0) grid (6,2);
4   \draw[red](0,1)-- +(1,1)-- +(1,-1)-- ++(2,1)-- +(1,-1);
5   \coordinate (A) at ($(5,0) +(120:2) $);
6   \coordinate (B) at (5,0);
7   \draw[blue] (A) -- (B) -- ([turn](90:1cm));
8   \fill[green] ($(A)!.6!(B) $) circle (.1cm);
9 \end{tikzpicture}
```



### weitere Koordinatensysteme, spezielle Punkte

- ▶ Relative Koordinaten in nodes `node cs`
- ▶ Tangenten an Kurven `tangent cs`
- ▶ Schnittpunkte von Kurven
- ▶ spezielle Punkte von vordefinierten Strukturen

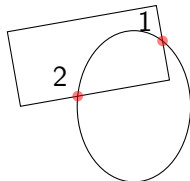
```
1 \begin{tikzpicture}[line cap=rect]
2   \coordinate (a) at (0,0);
3   \node[circle,draw] (c) at (1,0) [minimum size=1cm]{};
4   \draw(tangent cs:node=c,point={(a)},solution=1) -- (a);
5   \draw(tangent cs:node=c,point={(a)},solution=2) -- (a);
6 \end{tikzpicture}
```



# Libraries

## intersections

```
1 \usetikzlibrary{intersections}
2 \begin{tikzpicture}[every node/.style={opacity=1,black,
   above left}]
3   \draw [name path=elli] (1.5,0) ellipse (0.75cm and 1cm);
4   \draw [name path=rect,rotate=10] (0,0) rectangle +(2,1);
5   \fill [red, opacity=0.5,name intersections={of=elli and
   rect}]
6     (intersection-1) circle (2pt) node {1}
7     (intersection-2) circle (2pt) node {2};
8 \end{tikzpicture}
```



### shapes

- ▶ vorgefertigte Formen
  - ▶ Polygone, Halbkreise, Sterne, ...

```
1 \begin{tikzpicture}
2 \foreach \a in {3,...,7}{
3 \draw[blue, dashed] (\a*2,0) circle(0.5cm);
4 \node[regular polygon, regular polygon sides=\a, minimum
   size=1cm, draw] at (\a*2,0) {};
5 }
6 \end{tikzpicture}
```





# Libraries

## shapes

```
1 \usetikzlibrary{shapes}
2 \tikzstyle{mybox}=[draw=red,fill=gray!20,very thick,
   rectangle,rounded corners,inner sep=15pt]
3 \tikzstyle{ftitle}=[fill=red,text=white,ellipse]
4 \begin{tikzpicture}
5   \node [mybox,text width=3cm] (box) {
6     Ein Text mit Formel  $e=mc^2$ .};
7   \node[ftitle] at (box.north) {Frame};
8 \end{tikzpicture}
```

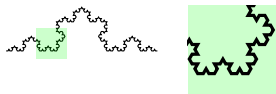
Frame

Ein Text mit Formel  $e = mc^2$ .

# Libraries

## spy

```
1 \usetikzlibrary{spy,decorations.fractals}
2 \begin{tikzpicture}[spy using overlays={size=12mm}]
3   \draw [decoration=Koch snowflake]
4     decorate{decorate{decorate{decorate{(0,0)--(2,0)}}}}};
5   \spy [green,magnification=3] on (0.6,0.1) in node at
6     (3,0);
7 \end{tikzpicture}
```



### Grafiken auslagern

- ▶ schnelleres Übersetzen (nur neue Bilder)
  - ▶ braucht Shellzugriff: `pdflatex -shell-escape <Jobname>`
- ▶ Grafiken auch als Datei für andere Zwecke verwenden
- ▶ einzelne Bilder deaktivieren mit `\tikzexternaldisable/enable`

```
1 \usetikzlibrary{external}  
2 \tikzexternalize [<Optionen>]
```

### Optionen

- prefix** Speicherordner (eventuell mit `\tikzsetnextfilename{<Name>}`)
- mode** shell-escape oder Makefile

## standalone-Dokumente

- ▶ Bisher: Tikz für Grafiken im  $\text{\LaTeX}$ -Dokument
- ▶ So bekommt man die Grafiken aus dem Dokument heraus (gut zum Testen):

```
1 \documentclass[tikz,border=0pt]{standalone}
2 \begin{document}
3 \begin{tikzpicture}
4   % Grafikbefehle
5 \end{tikzpicture}
6 \end{document}
```

## Grafiken einbinden

- ▶ externe (Pixel)grafiken einbinden
- ▶ einmal definieren, oft verwenden (kleinere PDFs)
  - ▶ `\pgfdeclareimage[<Optionen>]{<Name>}{<Datei>}`
  - ▶ `\pgfuseimage{<Name>}`

```
1 \pgfdeclareimage[width=2cm]{Logo}{UnixAG}
2 \begin{tikzpicture}
3   \node at (0,0) {\pgfuseimage{Logo}};
4 \end{tikzpicture}
```



# Libraries

weitere

## Weitere Libraries

**calendar** Kalender erstellen

**fadings** Transparenz, Verläufe

**mindmap** für Mindmaps

**folding** Bastelausschneidebögen

**pattern** Füllungen

**shadings** weitere Farbverläufe

**shadows** Schatten

**tree** Bäume

...

TikZ

Libraries

**Pakete aufbauend auf TikZ**

PGFPLOTS

chemfig

CircuitTikZ

Zusammenfassung und Ausblick

# Pakete aufbauend auf TikZ

## pgfplots

### Nur TikZ

- ▶ einfache Plots
  - ▶ `\tikz \draw plot[<Optionen>] file {<Daten.dat>;}`
- ▶ kompliziert
  - ▶ `\datavisualization` mit Library

### pgfplots

- ▶ `\usepackage{pgfplots}`
- ▶ relativ einfach zu bedienen
- ▶ mächtig
- ▶ Achtung: Speicherhungrig, eventuell mit Paket `etex`



# Pakete aufbauend auf TikZ

## pgfplots

### Datenformat

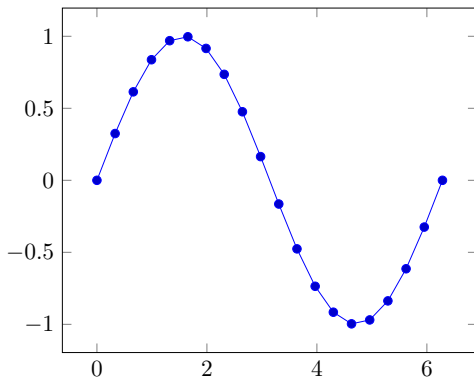
- ▶ geht inline, besser: externe Dateien
- ▶ einfache Textdateien
- ▶ exportiert aus Excel, Matlab, o.ä.

```
1 x          y
2 # sinus.dat
3 0          0
4 0.3306    0.3246
5 0.6613    0.6142
6 ...
7 6.2831    0
```

# Pakete aufbauend auf TikZ

## pgfplots

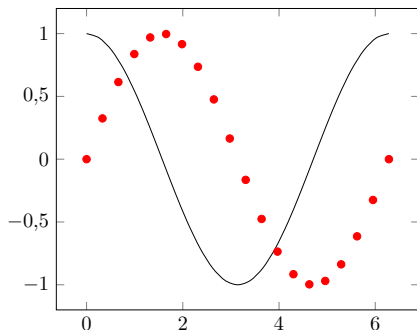
```
1 \begin{tikzpicture}[scale=.8]
2   \begin{axis}[only marks]
3     \addplot table {sinus.dat};
4   \end{axis}
5 \end{tikzpicture}
```



# Pakete aufbauend auf TikZ

## pgfplots

```
1 \pgfplotsset{/pgf/number format/use comma}
2 \begin{tikzpicture}[scale=.7]
3   \begin{axis}
4     \addplot [only marks,red] table {sinus.dat};
5     \addplot [domain=0:2*pi,smooth]{cos(deg(x))};
6   \end{axis}
7 \end{tikzpicture}
```



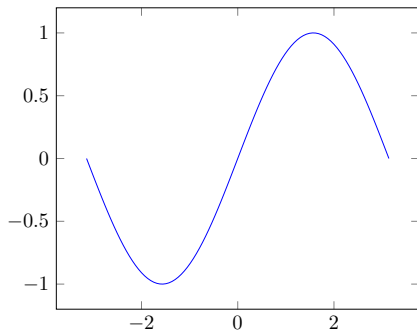
### Gnuplot

- ▶ Volle Gnuplot-Funktionalität
  - ▶ Fitten
  - ▶ Rechnen
- ▶ `pdflatex -shell-escape <Jobname>`

```
1 \begin{tikzpicture}[scale=.7]
2   \begin{axis}
3     \addplot+[raw gnuplot, mark=none] gnuplot {
4       f(x)=a*sin(b*x);
5       a=.5; b=.5;
6       fit f(x) 'sinus.dat' u 1:2 via a,b;
7       plot [x=-pi:pi] f(x);
8     };
9   \end{axis}
10 \end{tikzpicture}
```

# Pakete aufbauend auf TikZ

## pgfplots



# Pakete aufbauend auf TikZ

## pgfplots

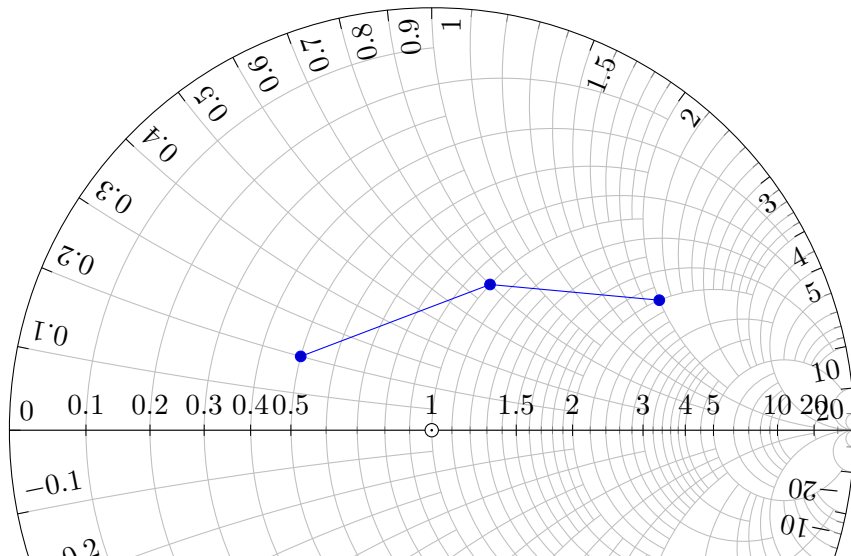
### andere Achsen

	<code>\begin{}</code>	<code>\usepgfplotslibrary{}</code>
Logarithmisch	<code>loglogaxis</code> <code>semilogyaxis</code>	
Ploar	<code>ploaraxis</code>	<code>polar</code>
Smith-Chart	<code>smithchart</code>	<code>smithchart</code>

```
1 \usepgfplotslibrary{smithchart}
2 ...
3 \begin{tikzpicture}
4   \begin{smithchart}[width=14cm]
5     \addplot coordinates {(0.5,0.2) (1,0.8) (2,2)};
6   \end{smithchart}
7 \end{tikzpicture}
```

# Pakete aufbauend auf TikZ

## pgfplots



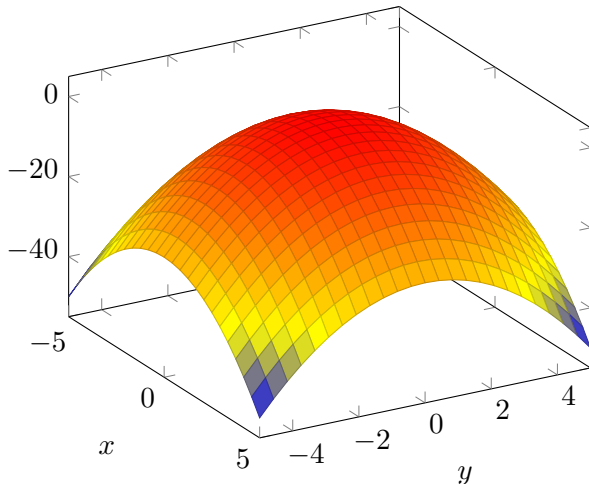
### 3D-Plots

▶ `\addplot3 {<Funktion>};`

```
1 \begin{tikzpicture}
2   \begin{axis}[view={60}{30}, xlabel=$x$, ylabel=$y$,
3     title=3D-Plot]
4     \addplot3 [surf] {-x^2-y^2};
5   \end{axis}
6 \end{tikzpicture}
```



3D-Plot



### Legenden

- ▶ mit `legend entries`-Option der `axis`-Umgebung
- ▶ mit `\addlegendentry` nach `\addplot`
- ▶ Formatierung
  - ▶ `legend style`
  - ▶ `legend pos`
  - ▶ `legend columns`

### Referenzen

- ▶ mit `\label` und `\ref`

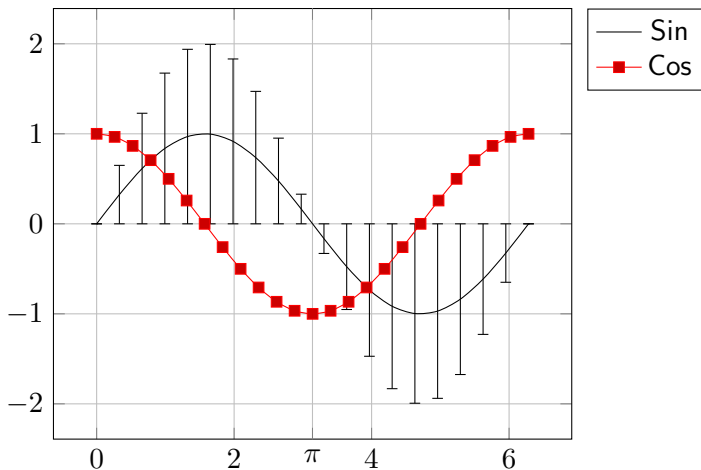
# Pakete aufbauend auf TikZ

## pgfplots

```
1 \begin{tikzpicture}
2   \begin{axis}[domain=0:2*pi,smooth,
3     legend pos=outer north east,legend entries={Sin,Cos},
4     grid=major,
5     extra x ticks={3.14},extra x tick labels={ $\pi$ }]
6     \addplot[error bars/y dir=both,error bars/y explicit]
7       table [x index=0,y index=1,y error index=1]
8         {sinus.dat};\label{legend:sin}
9     \addplot {cos(deg(x))};\label{legend:cos}
10  \end{axis}
11 \end{tikzpicture}
12 Das sind Sinus \ref{legend:sin} und Kosinus \ref{legend:
    cos}.
```

# Pakete aufbauend auf TikZ

pgfplots



Das sind Sinus — und Kosinus —■—.

### Weitere pgfplotslibraries

**clickable** Koordinaten anzeigen lassen

**colormaps** Weitere Farbschemata

**dateplot** Für Datumsangaben als Achsen

**groupplot** Matrizenanordnung von Plots

**statistics** Für Verteilungen, Fehlergrenzen

...

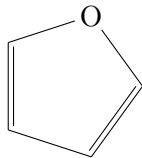
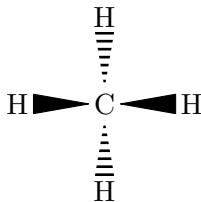
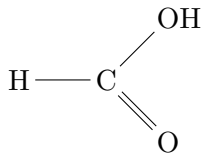
### ChemFig

- ▶ Setzt chemische Strukturformeln
- ▶ Bindungen - = ~ >
- ▶ Position [0-7] oder :<Winkel>
- ▶ Ringe mit <atom>\*<n>( <andere Atome>)
- ▶ Kann auch komplette Reaktionsmechanismen

# Pakete aufbauend auf TikZ

## chemfig

```
1 \chemfig{H-C~C-CH_3}
2 \chemfig{H-C(=[7]O)(-[1]OH)}
3 \chemfig{H>C(<:[:90]H)(<:[:-90]H)<H}
4 \chemfig{*5(-=O=)}
```



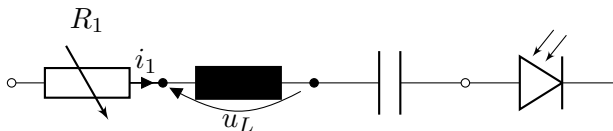
# Pakete aufbauend auf TikZ

## CircuitTikZ

### CircuitTikZ

- ▶ Für elektrische Schaltungen
- ▶ auch Logikschaltungen
- ▶ Eigene Umgebung `circuitikz` statt `tikzpicture`
- ▶ `\usepackage[european]{circuitikz}`

```
1 \draw (0,0) to [vR,l=$R_1$, i^>=$i_1$,o-*] (2,0) to [L,v=$u_L$] (4,0) to [C,*-o] (6,0) to [pD] (8,0);
```

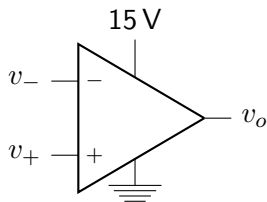




# Pakete aufbauend auf TikZ

## CircuitTikZ

```
1 \begin{circuitikz}
2   \draw (0,0) node [op amp] (opamp) {}
3     (opamp.+) node [left] {$v_+$}
4     (opamp.-) node [left] {$v_-$}
5     (opamp.out) node [right] {$v_o$}
6     (opamp.down) node [ground]{}
7     (opamp.up)-- ++(0,.5) node [above] {\SI{15}{\volt}};
8 \end{circuitikz}
```



TikZ

Libraries

Pakete aufbauend auf TikZ

**Zusammenfassung und Ausblick**

## Zusammenfassung

- ▶ Grafiken programmieren mit TikZ
- ▶ Funktionalität mit Bibliotheken erweitern
- ▶ Graphen mit pgfplots

## Nächste Vorträge

- ▶ Briefe und Lebensläufe
- ▶ Interaktive PDFs

Vielen Dank für eure Aufmerksamkeit!  
Danke an Klaus Denker für einige Beispiele.

**TikZ-Pakete Übersicht** <https://www.ctan.org/topic/pgf-tikz>

**pgf/TikZ** <https://www.ctan.org/pkg/pgf>

**xcolor** <https://www.ctan.org/pkg/xcolor>

**pgfplots** <https://www.ctan.org/pkg/pgfplots>

**circuitikz** <https://www.ctan.org/pkg/circuitikz>

**chemfig** <https://www.ctan.org/pkg/chemfig>