

Grafiken in \LaTeX mit TikZ und PGFPLOTS

Eine Einführung

Patrick Schulz

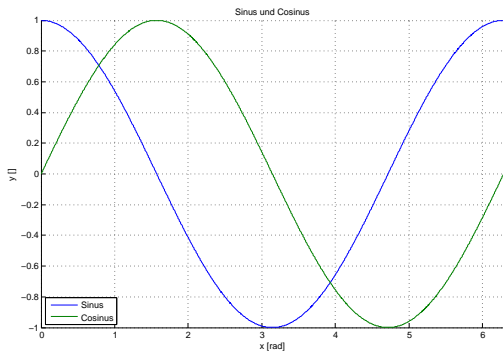
12. Februar 2016

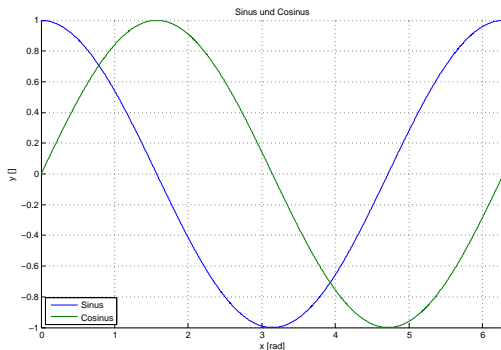
Thema

- Grafiken mit TikZ
- Plots (Kurven, Graphen) mit PGFPLOTS

Grafikprogramme

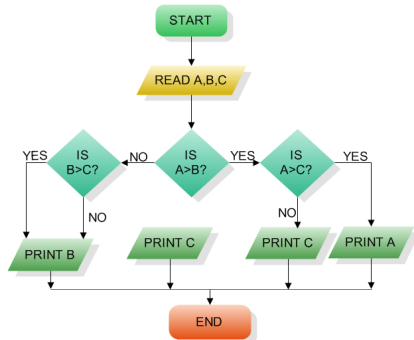
- Matlab, Octave, Gnuplot
- ...
- Paint?

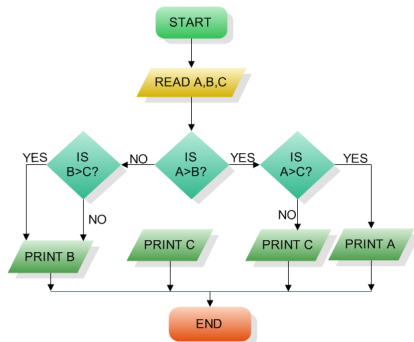




Probleme

- Schriftart
- Kleine Schrift
- Liniendicke





Probleme

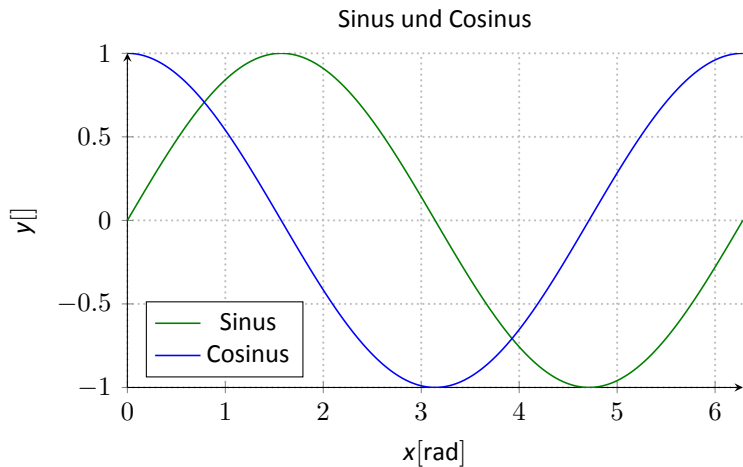
- Schriftart
- Bunte Farben
- Skalierbarkeit
- Schattierungen

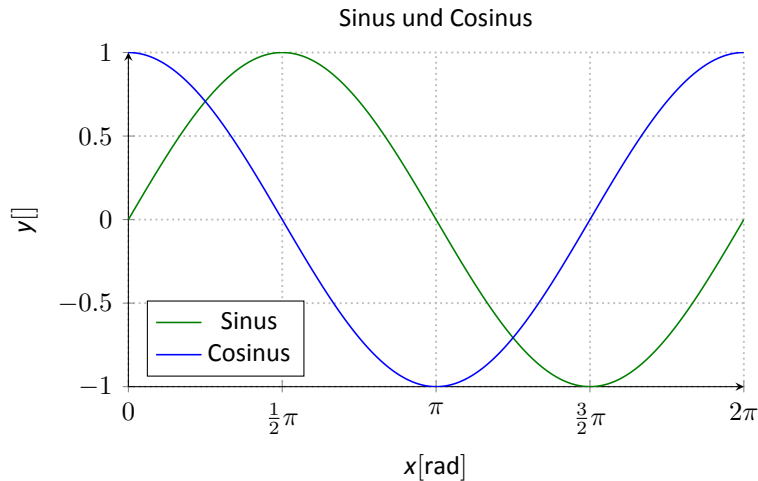
Grafiken in \LaTeX

- + identische Schriftarten
- + alle \LaTeX -Befehle verfügbar (z.B. Mathematikmodus)
- + einfaches Anpassen aller Grafiken
- umständlich
- eventuell langsam

Technische Grafiken

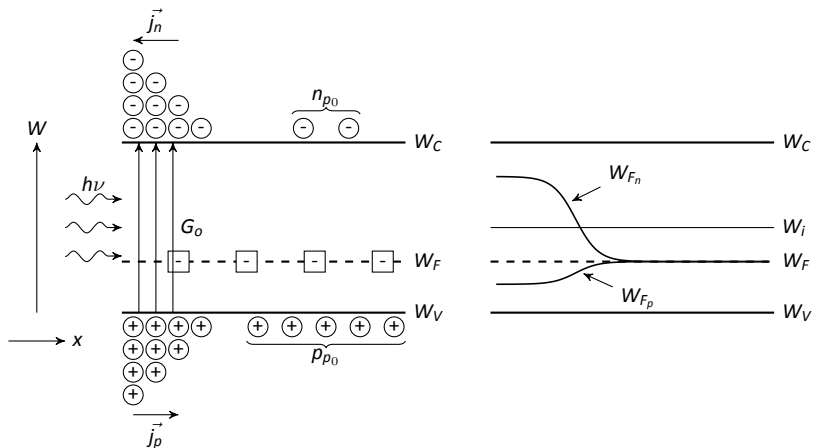
- einfache Strukturen
 - keine Schnörkel und Dekorationen
- Gut Programmierbar

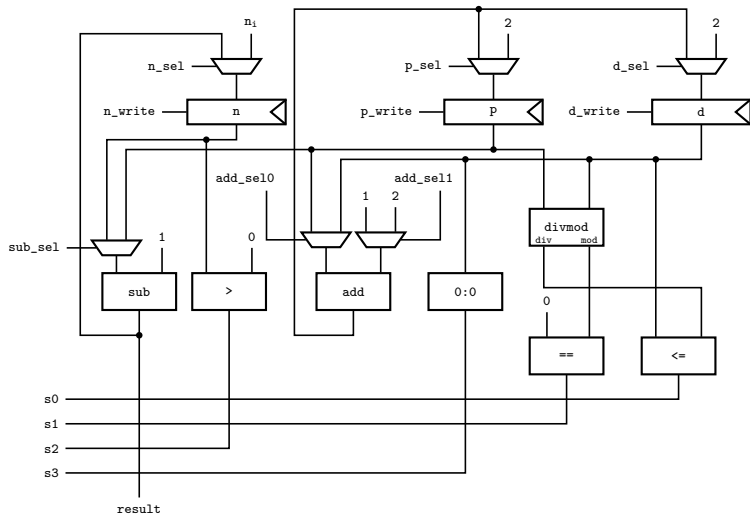


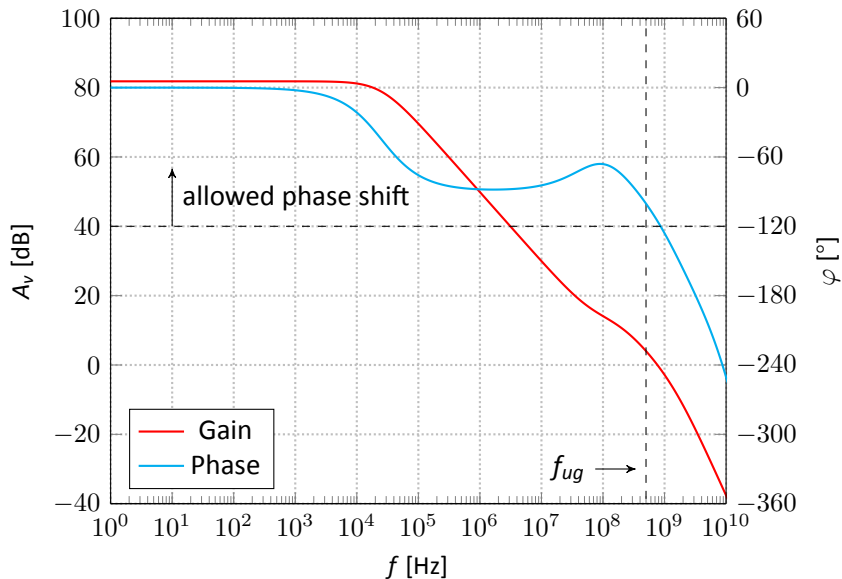


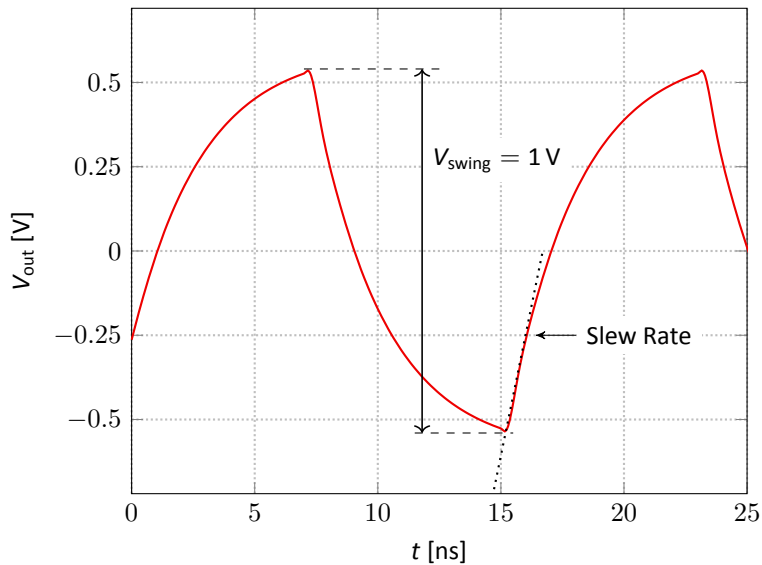
TikZ-Grafiken

Beispiel Bändermodell









TikZ

- Einführung
- Leitfaden mit Beispielen
- Hinweise

PGFPLOTS

- Einführung
- Leitfaden mit Beispielen
- Hinweise

Weitere Informationen

- Allgemeine Hinweise
- Hilfsmittel und Quellen

TikZ

Hintergrund

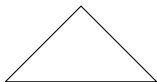
- TikZ ist kein Zeichenprogramm
→ Sprache zum Erstellen von Grafiken
- Autor: Till Tantau
- TikZ als Frontend für *PGF*
- Reguläres \LaTeX -Paket

Verwendung

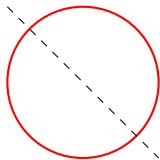
- `\usepackage{tikz}`
- Zeichenbefehle innerhalb von
`\begin{tikzpicture} ... \end{tikzpicture}`
- Zeichnen mit `\draw`, `\fill`, `\node` usw.
- Koordinaten mit `(x, y)`, `(phi:r)` und weitere
- Pfadoperationen: `--`, `rectangle`, `circle`
- Grundlegende Struktur: `\draw (x1, y1) pfadoperation (x2, y2);`
- Zeichenbefehle mit Semikolon abschließen

Grundlegende Pfadoperationen

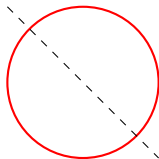
```
\begin{tikzpicture}  
  \draw (0, 0) -- (1, 1) -- (2, 0) -- (0, 0); % Dreieck  
  \draw (3, 0) rectangle (5, 1); % Rechteck  
  \draw (7, 0.5) circle (0.5); % Kreis  
\end{tikzpicture}
```



```
\begin{tikzpicture}  
  \draw[red, thick] (0, 0) circle (1);  
  \draw[dashed] (-1, 1) -- (1, -1);  
\end{tikzpicture}
```



```
\begin{tikzpicture}  
  \draw[red, thick] (0, 0) circle (1);  
  \draw[dashed] (-1, 1) -- (1, -1);  
\end{tikzpicture}
```



Optionen

- Farben
- Linienstile: Dicken und Pfeilspitzen
- Transparenz
- viele, viele weitere

Text in Grafiken

- Platzierung mit `\node`
- Position auf dem aktuellen Pfad, mit `at (x, y)` oder relativ zu anderen Koordinaten
- Nodes haben eine Form
- Nodes können benannt werden
- Text in `{...}` (Nicht optional, kann aber leer sein)
- Syntax: `\node[options] at (x, y) (name) { Text };`

```
\begin{tikzpicture}
  \node { Harder };
  \node[draw] at (2, 0) (n1) { Better };
  \node[right=of n1] { Faster };
  \path (6, 0) node[draw, circle] { Stronger };
  \draw (0, -2) -- node[above] { Go Crescendolls! } (6, -2);
\end{tikzpicture}
```

Harder

Better

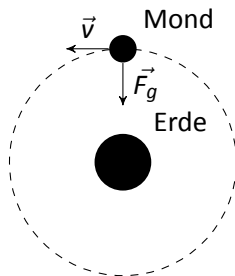
Faster

Stronger

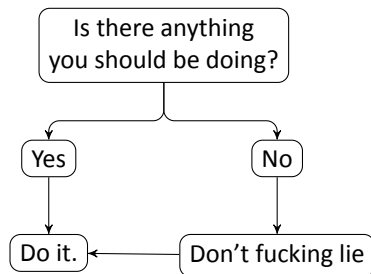
Go Crescendolls!

Beispiel

Erde und Mond, Flowchart



Drawing to scale.

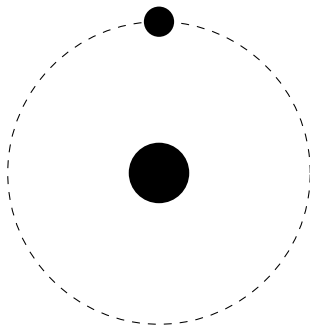


```
\begin{tikzpicture}  
  \fill (0, 0) circle (0.4); % Erde im Ursprung (0, 0)  
\end{tikzpicture}
```

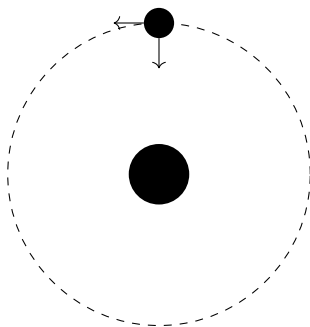


→ Wichtig: die internen Koordinaten spielen keine Rolle!

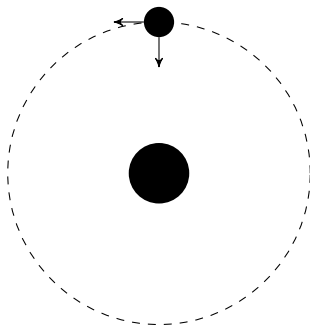

```
\begin{tikzpicture}  
  \fill (0, 0) circle (0.4);  
  \draw[dashed] (0, 0) circle(2); % Umlaufbahn  
  \fill (0, 2) circle (0.2); % Mond auf der Umlaufbahn  
\end{tikzpicture}
```



```
\begin{tikzpicture}
  \fill (0, 0) circle (0.4);
  \draw[dashed] (0, 0) circle(2);
  \fill (0, 2) circle (0.2);
  \draw[->] (-0.2, 2) -- (-0.6, 2); % Bewegungsrichtung
  \draw[->] (0, 1.8) -- (0, 1.4); % Kraftrichtung
\end{tikzpicture}
```



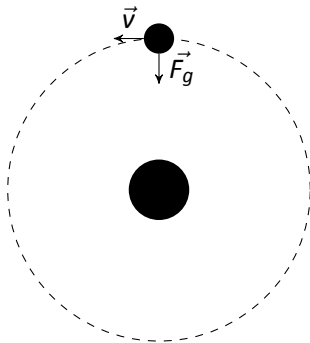
```
\begin{tikzpicture}[>=stealth'] % mit TikZ-Bibliothek 'arrows'  
  \fill (0, 0) circle (0.4);  
  \draw[dashed] (0, 0) circle(2);  
  \fill (0, 2) circle (0.2);  
  \draw[->] (-0.2, 2) -- (-0.6, 2);  
  \draw[->] (0, 1.8) -- (0, 1.4);  
\end{tikzpicture}
```



Erde und Mond

Beschriftung der Pfeile

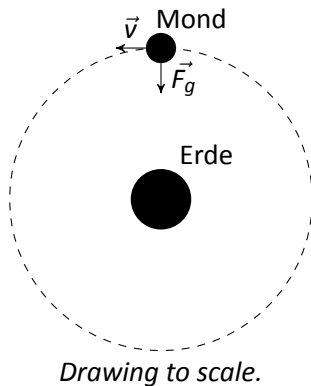
```
% Code wie zuvor  
\node[above] at (-0.4, 2) {  $\vec{v}$  };  
\node[right] at (0, 1.6) {  $\vec{F}_g$  };  
\end{tikzpicture}
```



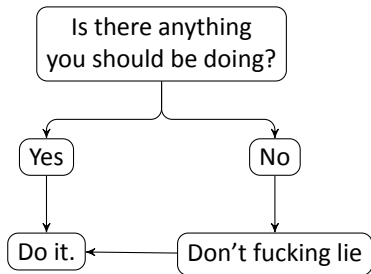
Erde und Mond

Text und Hinweis

```
% code wie zuvor
\node[above] at (-0.4, 2) {  $\vec{v}$  };
\node[right] at (0, 1.6) {  $\vec{F}_g$  };
\node at (0.6, 0.6) { Erde };
\node at (0.4, 2.4) { Mond };
\node[below] at (0, -2) { \emph{Drawing to scale.} };
\end{tikzpicture}
```



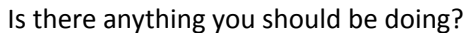
```
\begin{tikzpicture}[>=stealth']
  \fill (0, 0) circle (0.4); % Erde
  \draw[dashed] (0, 0) circle(2); % Umlaufbahn
  \fill (0, 2) circle (0.2); % Mond
  \draw[->] (-0.2, 2) -- (-0.6, 2); % Geschwindigkeit
  \draw[->] (0, 1.8) -- (0, 1.4); % Kraft
  \node[above] at (-0.4, 2) {  $\vec{v}$  }; % Vektor
  \node[right] at (0, 1.6) {  $\vec{F}_g$  }; % Vektor
  \node at (0.6, 0.6) { Erde }; % Text Erde
  \node at (0.4, 2.4) { Mond }; % Text Mond
  % Hinweis:
  \node[below] at (0, -2) { \emph{Drawing to scale.} };
\end{tikzpicture}
```



```
\node[draw, rectangle] { Is there anything you should be doing? };
```

Is there anything you should be doing?

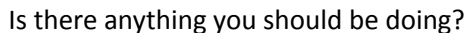

```
\node[draw, rectangle] { Is there anything you should be doing? };
```



Is there anything you should be doing?

```
\node[below left=of ...] { Yes };
```

```
\node[draw, rectangle] { Is there anything you should be doing? };
```



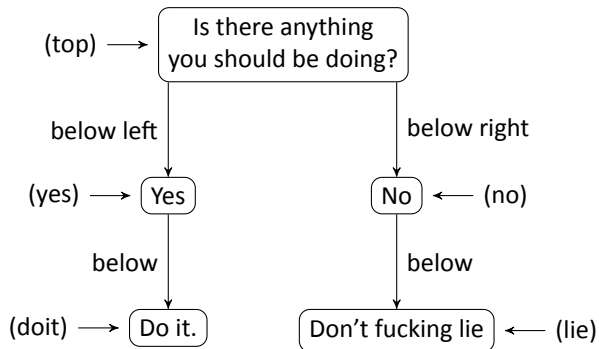
Is there anything you should be doing?

```
\node[below left=of ...] { Yes };
```

→ Die Nodes müssen benannt werden.

Flowchart

Positionierung und Benennung



Flowchart

Restliche Knoten – relative Positionierung

```
\node[draw, rectangle] (top) { Is there anything you should be  
doing? };  
\node[draw, rectangle, below left=of top] (yes) { Yes };  
\node[draw, rectangle, below right=of top] (no) { No };  
\node[draw, rectangle, below =of yes] (doit) { Do it. };  
\node[draw, rectangle, below =of no] (lie) { Don't fucking lie };
```

Is there anything you should be doing?

Yes

No

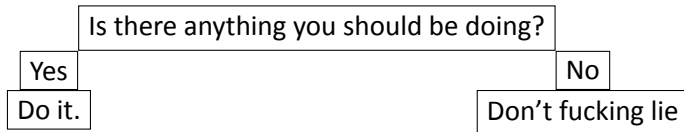
Do it.

Don't fucking lie

Flowchart

Abstand zwischen den Knoten

```
\begin{tikzpicture}[node distance=0cm and 0cm]
  \node[draw, rectangle] (top) { Is there anything you should
    be doing? };
  \node[draw, rectangle, below left=of top] (yes) { Yes };
  \node[draw, rectangle, below right=of top] (no) { No };
  \node[draw, rectangle, below =of yes] (doit) { Do it. };
  \node[draw, rectangle, below =of no] (lie) { Don't fucking
    lie };
\end{tikzpicture}
```



Flowchart

Abstand zwischen den Knoten

```
\begin{tikzpicture}[node distance=0.75cm and -0.5cm]
  \node[draw, rectangle] (top) { Is there anything you should
    be doing? };
  \node[draw, rectangle, below left=of top] (yes) { Yes };
  \node[draw, rectangle, below right=of top] (no) { No };
  \node[draw, rectangle, below =of yes] (doit) { Do it. };
  \node[draw, rectangle, below =of no] (lie) { Don't fucking
    lie };
\end{tikzpicture}
```

Is there anything you should be doing?

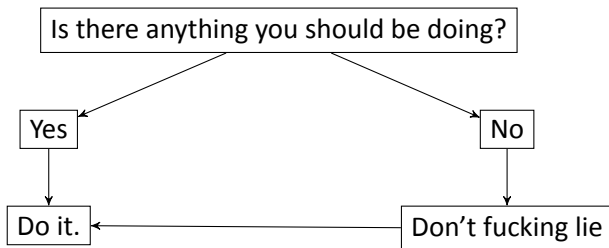
Yes

No

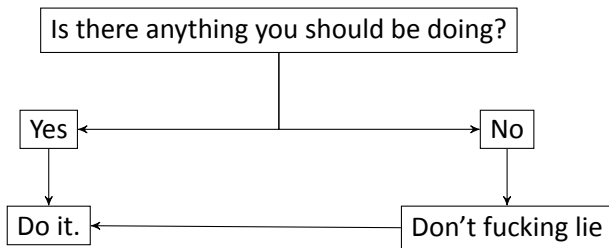
Do it.

Don't fucking lie

```
% nodes wie zuvor
\draw[->, >=stealth'] (top) -- (yes);
\draw[->, >=stealth'] (top) -- (no);
\draw[->, >=stealth'] (yes) -- (doit);
\draw[->, >=stealth'] (no) -- (lie);
\draw[->, >=stealth'] (lie) -- (doit);
```

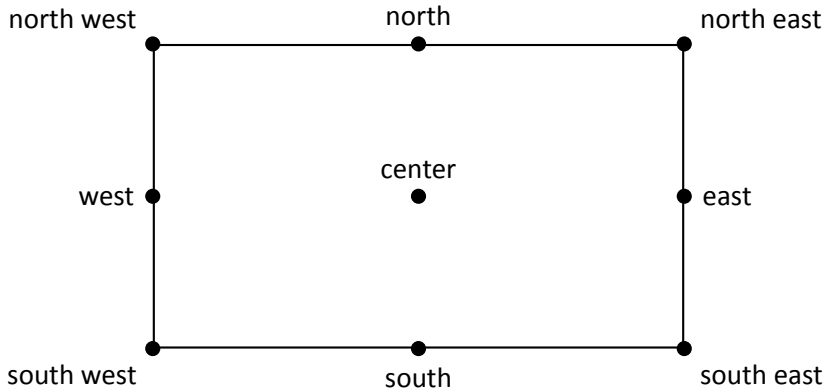


```
% nodes wie zuvor
\draw[->, >=stealth'] (top) |- (yes);
\draw[->, >=stealth'] (top) |- (no);
\draw[->, >=stealth'] (yes) -- (doit);
\draw[->, >=stealth'] (no) -- (lie);
\draw[->, >=stealth'] (lie) -- (doit);
```



Einschub

Nodes und Fixpunkte (anchors)



```
% nodes wie zuvor
```

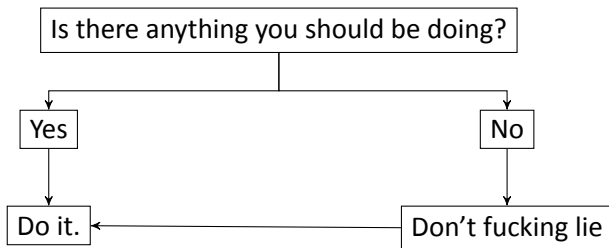
```
\draw[->, >=stealth'] (top.south) -- ++(0, -0.5) -| (yes);
```

```
\draw[->, >=stealth'] (top.south) -- ++(0, -0.5) -| (no);
```

```
\draw[->, >=stealth'] (yes) -- (doit);
```

```
\draw[->, >=stealth'] (no) -- (lie);
```

```
\draw[->, >=stealth'] (lie) -- (doit);
```



```
% nodes wie zuvor
```

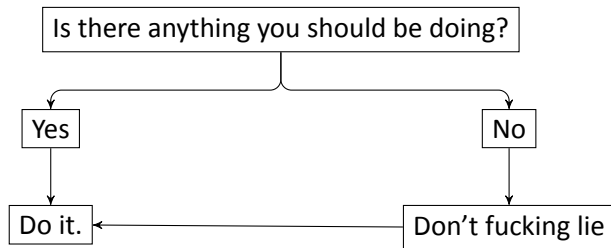
```
\draw[->, >=stealth', rounded corners] (top.south) -- ++(0, -0.5)  
  -| (yes);
```

```
\draw[->, >=stealth', rounded corners] (top.south) -- ++(0, -0.5)  
  -| (no);
```

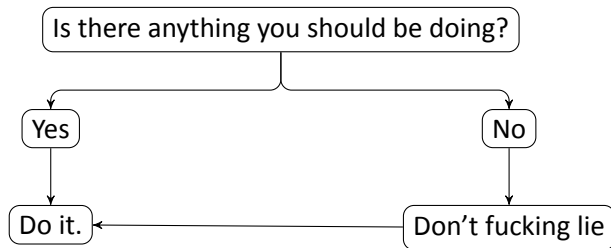
```
\draw[->, >=stealth'] (yes) -- (doit);
```

```
\draw[->, >=stealth'] (no) -- (lie);
```

```
\draw[->, >=stealth'] (lie) -- (doit);
```



```
\begin{tikzpicture}  
  [  
    node distance=0.75cm and -0.5cm,  
    rounded corners,  
    >=stealth'  
  ]  
  % code wie zuvor  
\end{tikzpicture}
```



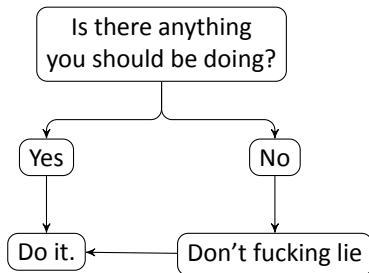
```
\node[draw] at (0, 0)
  { Ein langer Text, der (hoffentlich) umgebrochen wird. };
\node[draw, text width=4cm] at (0, -1.5)
  { Ein langer Text, der (hoffentlich) umgebrochen wird. };
\node[draw, align=center] at (0, -3)
  { Ein langer Text, der \\(manuell) umgebrochen wird. };
```

Ein langer Text, der (hoffentlich) umgebrochen wird.

Ein langer Text, der
(hoffentlich) umgebro-
chen wird.

Ein langer Text, der
(manuell) umgebrochen wird.

```
% Einstellungen wie zuvor
font=\small
]
\node[
draw,
rectangle,
align=center
] (top) { Is there anything \\ you should be doing? };
% Code wie zuvor
```



```
\begin{tikzpicture}
[
  % Einstellungen
  node distance=0.75cm and -0.5cm,
  rounded corners,
  font=\small,
  >=stealth'
]
% Knoten
\node[draw, rectangle, align=center]
  (top) { Is there anything \\ you should be doing? };
\node[draw, rectangle, below left=of top] (yes) { Yes };
\node[draw, rectangle, below right=of top] (no) { No };
\node[draw, rectangle, below =of yes] (doit) { Do it. };
\node[draw, rectangle, below =of no] (lie) { Don't fucking lie };
% Verbindungen
\draw[->] (top.south) -- ++(0, -0.5) -| (yes);
\draw[->] (top.south) -- ++(0, -0.5) -| (no);
\draw[->] (yes) -- (doit);
\draw[->] (no) -- (lie);
\draw[->] (lie) -- (doit);
\end{tikzpicture}
```

Global oder lokal?

- Lokale Einstellungen: `\draw[...]`, `\node[...]` und `\begin{tikzpicture}[...]`
- Globale Einstellungen: `\tikzset{...}`
- Einheitlicher Stil über gesamtes Dokument
- Wesentlich verkürzter Code
- Möglichkeit der Definition eines Stils:

```
\tikzset{
  state/.style={draw, rectangle, thick}
}
% Benutzung:
\node[state] { };
```


Erweiterung der TikZ-Befehle

- `positioning`, `arrows` bereits vorgestellt
- `calc`: Für erweiterte Koordinatenberechnungen (Projektionen, Schnittpunkte und mehr)
- `decorations`: Veränderung des Aussehens von Pfaden, zum Beispiel zufällige Sprünge, Sinuskurven oder Zick-Zack-Linien
- `matrix`: Definition von mehreren nodes als Einheit
- `pattern`: Füllen von Flächen mit Strukturen

→ Viele nützliche Bibliotheken, unbedingt nutzen!

PGFPLOTS

Übersicht

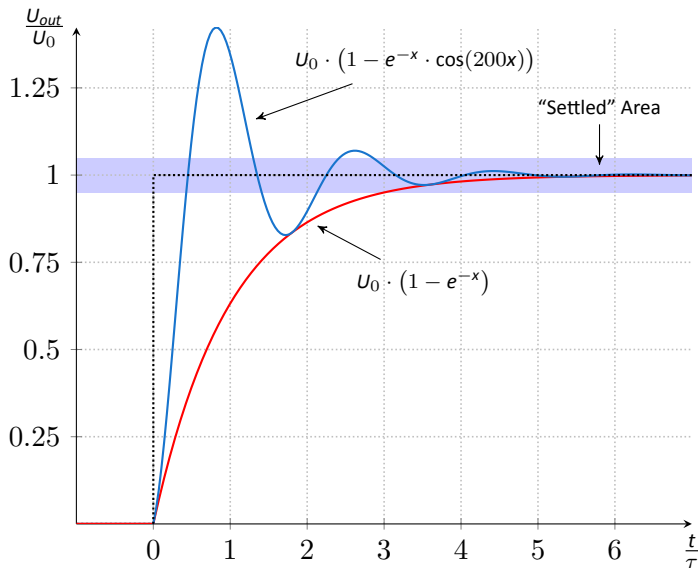
- Basiert auf TikZ/PGF
- Umgebungen für verschiedene Koordinatensysteme: Linear, Halb- / Volllogarithmisch, Polar, Smithcharts, ...
- Vielzahl von verfügbaren Dateneingaben
- Datenverarbeitung wie Regression und Filterung möglich
- Zeichnungen mit Hilfe von TikZ innerhalb Plots
- Sehr viele Optionen, aber gute Standardeinstellungen, hierdurch einfache Anwendung

Verwendung

- Wahl des Koordinatensystems durch Umgebung: `\begin{axis}` ...
→ Umgebung innerhalb `tikzpicture`
- Erzeugen von Kurven durch `\addplot`
- Wahl der Datenquelle:
 - ▶ Funktionen (`expression`)
 - ▶ Koordinaten (`coordinates`)
 - ▶ externe Daten (`table`)
- Achsenbeschriftung durch `xlabel`, `ylabel`, `title`, ...
- Legenden durch `\legend{...}`

Beispiel: Sprungantwort

Das Ziel



Beispiel: Sprungantwort

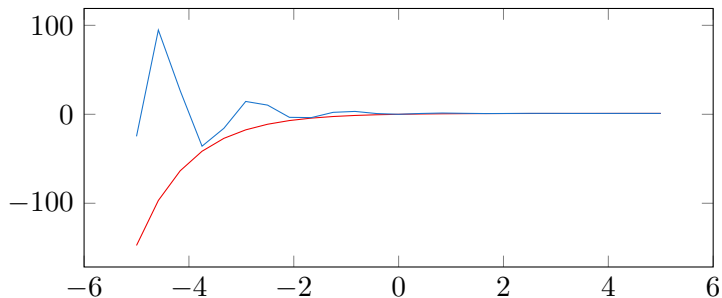
Erster Versuch

```
\begin{axis} % lineares Koordinatensystem
  % Sprung erster Ordnung
  \addplot expression { 1 - exp(-x) };
  % Sprung zweiter Ordnung
  \addplot expression { 1 - exp(-x) * cos(200*x) };
\end{axis}
```

Beispiel: Sprungantwort

Erster Versuch

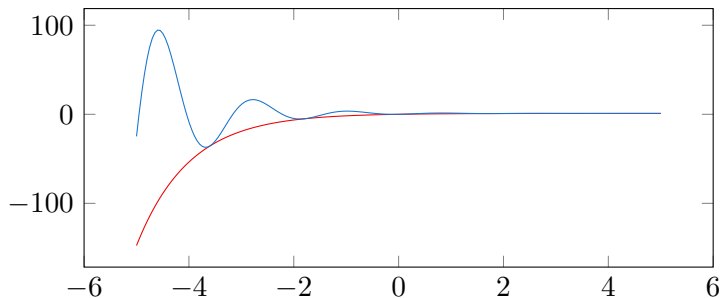
```
\begin{axis} % lineares Koordinatensystem
  % Sprung erster Ordnung
  \addplot expression { 1 - exp(-x) };
  % Sprung zweiter Ordnung
  \addplot expression { 1 - exp(-x) * cos(200*x) };
\end{axis}
```



Beispiel: Sprungantwort

Anzahl der Punkte

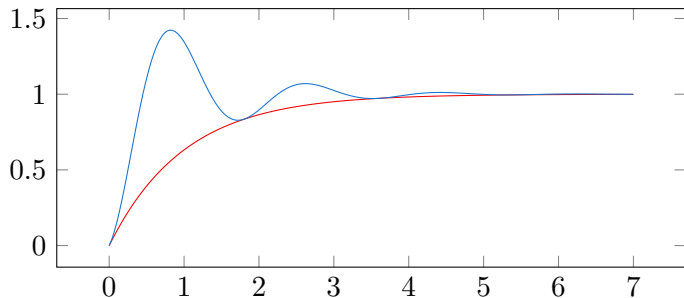
```
\begin{axis}[samples=200] % lineares Koordinatensystem
  % Sprung erster Ordnung
  \addplot expression { 1 - exp(-x) };
  % Sprung zweiter Ordnung
  \addplot expression { 1 - exp(-x) * cos(200*x) };
\end{axis}
```



Beispiel: Sprungantwort

Beschränkung der x-Koordinate

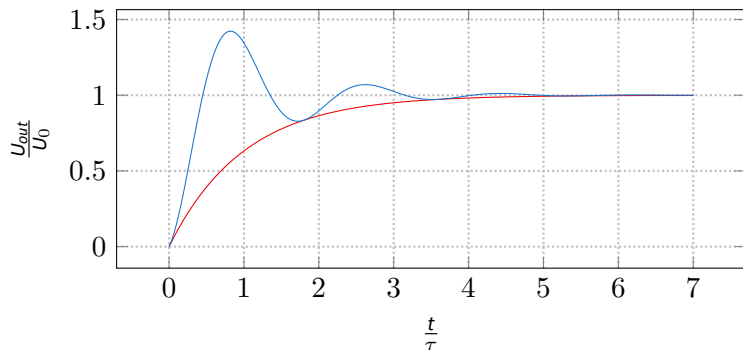
```
\begin{axis}[samples=200, domain=0:7]
  % Sprung erster Ordnung
  \addplot expression { 1 - exp(-x) };
  % Sprung zweiter Ordnung
  \addplot expression { 1 - exp(-x) * cos(200*x) };
\end{axis}
```



Beispiel: Sprungantwort

Achsenbeschriftung und Gitter

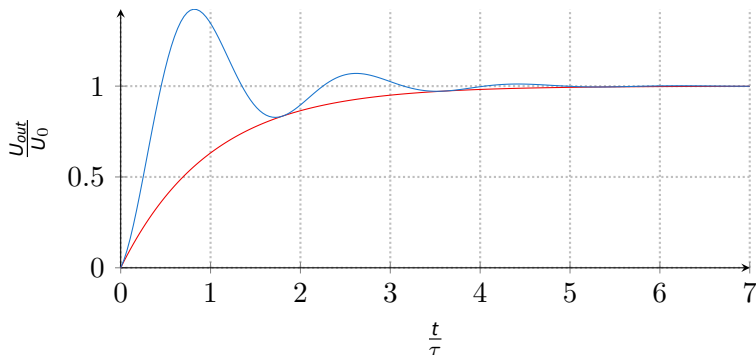
```
\begin{axis}[  
  samples=200, domain=0:7, grid,  
  xlabel={\frac{t}{\tau}},  
  ylabel={\frac{U_{out}}{U_0}}  
]  
  \addplot expression { 1 - exp(-x) };  
  \addplot expression { 1 - exp(-x) * cos(200*x) };  
\end{axis}
```



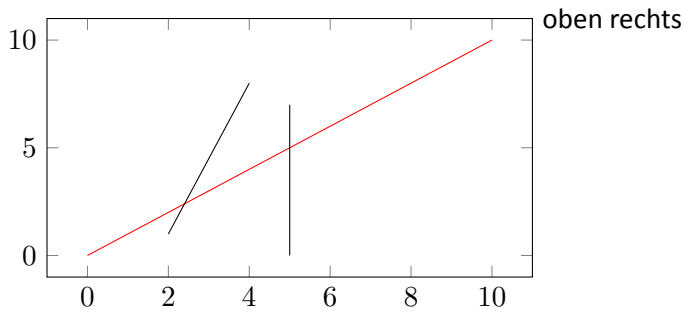
Beispiel: Sprungantwort

Art der Achsen

```
\begin{axis}[  
  % wie zuvor  
  axis x line=bottom, axis y line=left  
]  
  \addplot expression { 1 - exp(-x) };  
  \addplot expression { 1 - exp(-x) * cos(200*x) };  
\end{axis}
```



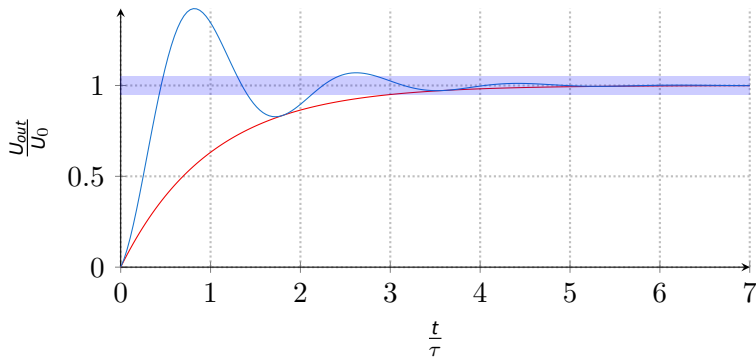
```
\addplot[red, domain=0:10] expression { x };  
\draw (axis cs:5, 0) -- (axis cs: 5, 7);  
\draw (2, 1) -- (4, 8); % implizit  
\draw (axis description cs: 0, -0.2)  
  -- (axis description cs: 1, -0.2);  
\node[right] at (axis description cs:1, 1) { oben rechts};
```



Beispiel: Sprungantwort

Füllfläche

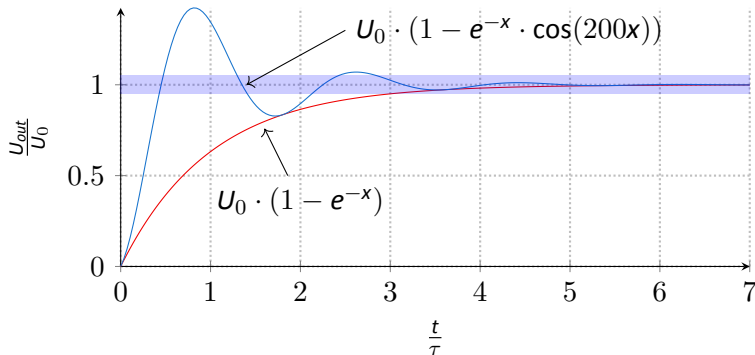
```
% Einstellungen wie zuvor  
\addplot expression { 1 - exp(-x) };  
\addplot expression { 1 - exp(-x) * cos(200*x) };  
\fill[blue, opacity=0.2] (0, 0.95) rectangle (7, 1.05);
```



Beispiel: Sprungantwort

Nodes in Plots

```
% Einstellungen und Plots wie zuvor
\node[below] at (2, 0.5) (p1) {  $U_0 \cdot (1 - e^{-x})$  };
\node[right] at (2.5, 1.3) (p2) {  $U_0 \cdot (1 - e^{-x})$ 
\cdot  $\cos(200x)$  };
\draw[->] (p1) -- +(-0.4, 0.4);
\draw[->] (p2.west) -- +(-1.1, -0.3);
```



Beispiel: Sprungantwort

Nodes in Plots – der bessere Weg

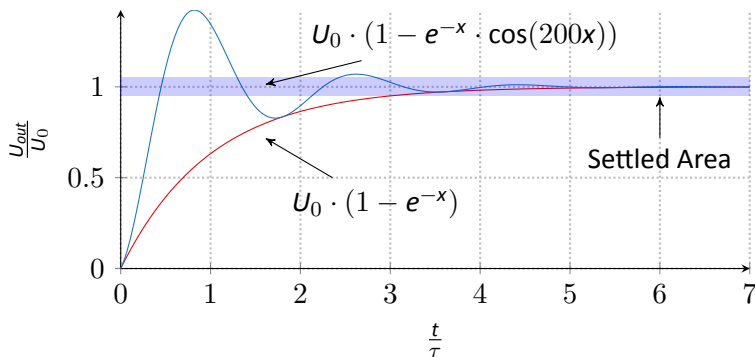
→ `\node[pin=angle:text] { };`

Beispiel: Sprungantwort

Nodes in Plots – der bessere Weg

→ `\node[pin=angle:text] { };`

```
% Einstellungen und Plots wie zuvor
\node[pin=-60:{$U_0 \cdot (1 - e^{-x})$}]
  at (1.5, 0.75) {};
\node[pin=30:{$U_0 \cdot (1 - e^{-x}) \cdot \cos(200x)$}]
  at (1.5, 1) {};
\node[pin=-90:{Settled Area}] at (6, 1) { };
```

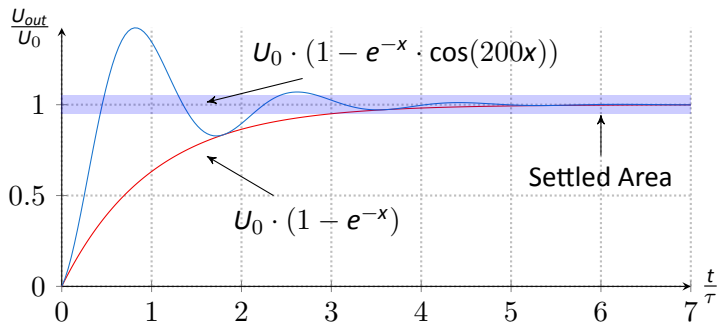


Beispiel: Sprungantwort

Platzierung der Achsenbeschriftungen

```
% Einstellungen wie zuvor
```

```
x label style={at={(axis description cs:1, 0)}, right},  
y label style={at={(axis description cs:0, 1)}, left,  
rotate=-90}
```

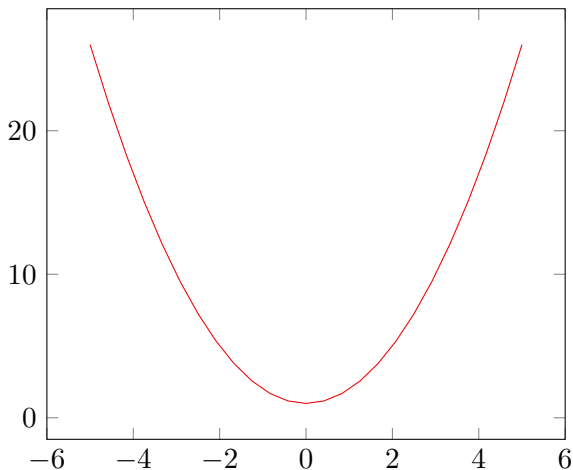


Beispiel: Sprungantwort

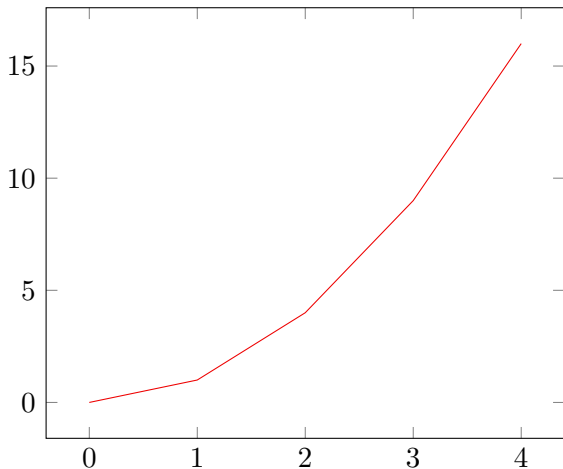
Der gesamte Code

```
\begin{axis}
[
  grid,
  grid style={semithick, densely dotted},
  xlabel={\frac{t}{\tau}},
  ylabel={\frac{U_{out}}{U_0}},
  axis x line=bottom,
  axis y line=left,
  xtick={0, 1, 2, 3, 4, 5, 6},
  ytick={0.25, 0.5, 0.75, 1, 1.25},
  x label style={at={(axis description cs:1, 0)}, below},
  y label style={at={(axis description cs:0, 1)}, left, rotate=-90},
  width=0.9\textwidth,
  height=0.9\textheight
]
% Plots wie zuvor
\end{axis}
```

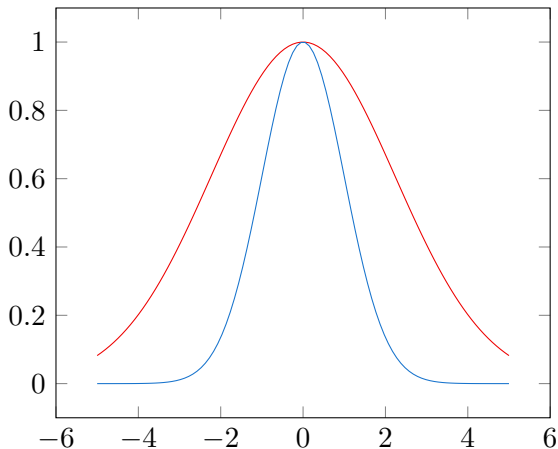
```
\begin{axis}  
  \addplot expression { x^2 + 1 };  
\end{axis}
```



```
\begin{axis}  
  \addplot coordinates { (0, 0) (1, 1) (2, 4) (3, 9) (4, 16) };  
\end{axis}
```



```
\begin{axis}  
  \addplot table[x=x, y=dist] {table_example.dat};  
  \addplot table[x index=0, y index=2] {table_example.dat};  
\end{axis}
```



| x | dist | dist2 | x,dist,dist2 |
|-------|------|-------|-----------------|
| -5.00 | 0.08 | 3.72 | -5.00,0.08,3.72 |
| -4.89 | 0.09 | 6.14 | -4.89,0.09,6.14 |
| -4.79 | 0.10 | 1.00 | -4.79,0.10,1.00 |
| -4.69 | 0.11 | 1.61 | -4.69,0.11,1.61 |
| -4.59 | 0.12 | 2.58 | -4.59,0.12,2.58 |
| -4.49 | 0.13 | 4.09 | -4.49,0.13,4.09 |
| -4.39 | 0.14 | 6.42 | -4.39,0.14,6.42 |
| -4.29 | 0.15 | 9.95 | -4.29,0.15,9.95 |

Optionen

- `table[x=name], table[y=name]`
- `table[x index=number], table[y index=number]`
- `table[col sep=space], table[col sep=comma]`

Farben, Größen, Liniendicken ...

- `\pgfplotsset`: Analog zu `\tikzset`
- Einheitliches Aussehen aller Graphen
- Wichtige Optionen:
 - ▶ `width` und `height`
 - ▶ `cycle list`
 - ▶ `grid style, every axis plot, ...`

Vorschlag

```
\pgfplotsset{
  every axis/.append style={grid, width=0.6\textwidth, height=5cm},
  grid style={densely dotted, thick},
  every axis plot/.append style={no markers, thick},
  label style={font=\small},
  tick label style={font=\small},
  legend pos={outer north east},
  cycle list={red, green, cyan, yellow}
}
```

Graphen können hierdurch sehr einfach erzeugt werden

```
\begin{axis}[
  xlabel={x},
  ylabel={y}
]
\addplot table {data.dat};
\end{axis}
```


- Table-Eingabedatei ohne Dateiendung: Annahme der Endung `.tex`
 - Dateiendung verwenden, z.B. `.txt`, `.dat`, ...
- Zeichnungen außerhalb des Graphen nicht sichtbar
 - Standardmäßiges Clipping: `\begin{axis}[clip=false]`
- Legende einiger Plots in Graph
 - Ignorieren einzelner Plots: `\addplot[forget plot]`
- Zwei y-Achsen in einem Graph
 - Verwendung zweier `axis`-Umgebungen, siehe Handbuch (*Two Ordinates (y axis) or Multiple Axes*)

Weitere Informationen

Probleme

- lange Übersetzungszeit (meine Bachelorarbeit: ca. eine Minute)
- auch nicht veränderte Grafiken müssen neu gesetzt werden

Lösung: tikzexternalize

- Export aller TikZ-Grafiken
- Import der Grafiken als PDF bei Übersetzung des Dokuments
- Automatische Erkennung von Veränderungen

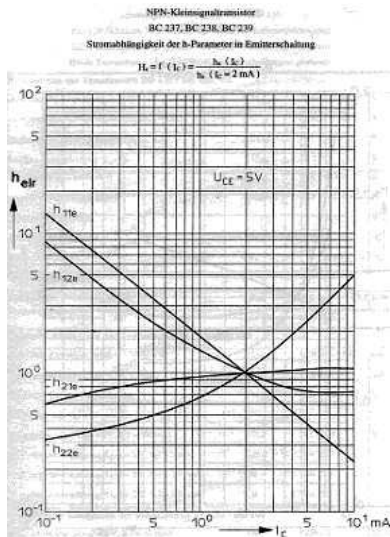
- Einfache Verwendung:

```
\usetikzlibrary{external}  
% bzw.  
\usepgfplotslibrary{external}  
\tikzexternalize
```

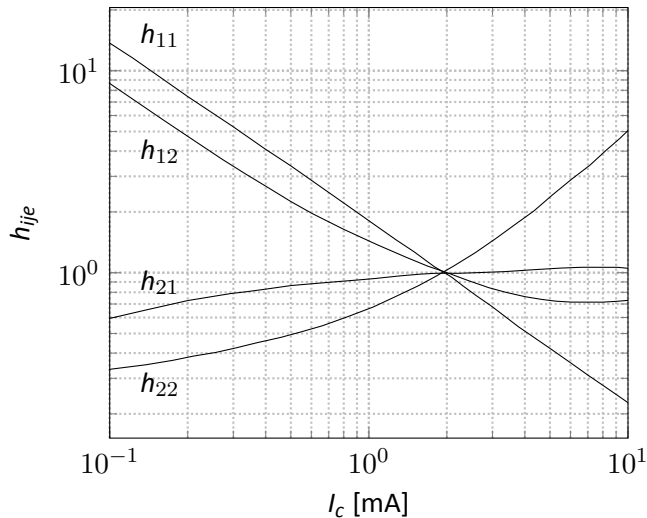
- externe Programme erlauben: `pdflatex -shell-escape`

Einbinden hässlicher Plots

Quelle



h-Paramter in Abhängigkeit des Kollektorstroms I_c



Tipps

- Gute Tutorials in den Handbüchern
- explizite Koordinaten vermeiden
- Erweiterungspakete: TikZ-Timingdiagramme, `tcolorbox` (Beispiele in diesen Folien)

Hinweise

- Verwendete PGFPLOTS-Version: 1.12 (`\pgfplotsset{compat=1.12}`)
- Verwendete TikZ-Version: 3.0.0

- TikZ-Handbuch
- PGFPLOTS-Handbuch
- Stackoverflow: `tex.stackexchange.com`
- `www.texample.net`
- Email an mich: pschulz@posteo.de

Vielen Dank für die Aufmerksamkeit!

Fragen?