

Wissenschaftl. Textverarbeitung mit \LaTeX

WS 2015/16 - 10. Vorlesung

Alexander Richter

Institut für Mathematische Optimierung

25. Januar 2016

In der letzten Woche wurde u.a. behandelt:

- ▶ beamer II
- ▶ $\text{Ti}k\text{Z}$ II
- ▶ spezielle Overlay-Effekte

- 1 Nachträge zu TikZ, beamer
 - Graph-Bibliothek
- 2 Erstellen von Plots
 - Plots in TikZ
 - Plots mit pgfplots
- 3 Weitere nützliche Klassen
- 4 Musterlösungen
- 5 Schluss
- 6 Organisation

- ▶ Option `remember picture` einer `tikzpicture` Umgebung: sorgt dafür, dass Knotennamen und Positionen auch in anderen Bildern zu Verfügung stehen.
- ▶ Option `overlay` für (Teil-)Pfade sorgt dafür, dass diese Teile nicht die Bounding-Box vergrößern

- 1 Nachträge zu TikZ, beamer
 - Graph-Bibliothek
- 2 Erstellen von Plots
 - Plots in TikZ
 - Plots mit pgfplots
- 3 Weitere nützliche Klassen
- 4 Musterlösungen
- 5 Schluss
- 6 Organisation

```
\usetikzlibrary{graphs}
```

und dann ...

```
\usetikzlibrary{graphs.standard}
```

....

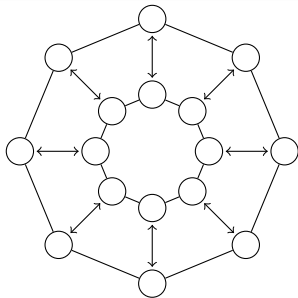
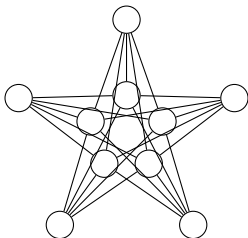
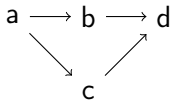
mit lualatex

```
\usetikzlibrary{graphdrawing.force}
```

```
\usetikzlibrary{graphdrawing.layered}
```

...

Beispiele



```
\graph { a -> {b, c} -> d };  
\graph [nodes={draw, circle}, clockwise,  
radius=.5cm, empty nodes, n=5] {  
  subgraph I_n [name=inner] --[complete bipartite]  
  subgraph I_n [name=outer]  };  
\graph [nodes={draw, circle}, clockwise,  
radius=.75cm, empty nodes, n=8] {  
  subgraph C_n [name=inner] <->[shorten <=1pt, shorten >=1pt]  
  subgraph C_n [name=outer] };
```

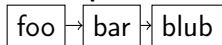
Benötigt wird hier: `\usetikzlibrary{graphs.standard}`

- ▶ kompakte Syntax, um Graphen händisch zu erzeugen
- ▶ Syntax vermischt die Philosophien von DOT (graphviz) und `TikZ`
- ▶ für Graphen bis ≈ 100 Knoten

Wie geht's?

- ▶ Intern wird alles über `nodes` und `edges` realisiert
- ▶ alles könnte auch manuell realisiert werden
- ▶ Positionen werden grob angegeben
- ▶ Ziel: Nutze Graph-Drawing-Algorithmen (nur `lualatex`)

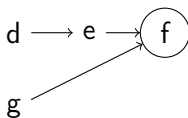
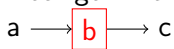
Konzept Node Chains



```
\tikz [every node/.style = draw]  
\graph { foo -> bar -> blub };
```

Knotenoptionen

in eckigen Klammern, nach Knotenspezifikation



```
\graph {  
a -> b[draw,red] -> c;  
d -> e -> f[draw, circle];  
g -> f; };
```

in eckigen Klammern, nach Kantenspezifikation

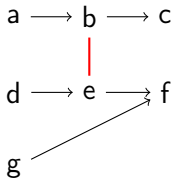
$a \xrightarrow{\text{red}} b \longrightarrow c$

$d \longrightarrow e \xrightarrow{\text{ultra thick}} f$

$g \longrightarrow f$

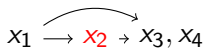
```
\graph {  
a ->[red] b -> c;  
d -> e ->[ultra thick] f;  
g -> f; };
```

Konzept Node Chains: Knotenname = Knoteninhalt



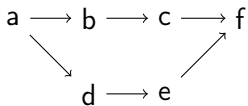
```
\graph {  
a -> b -> c;  
d -> e -> f;  
g -> f; };  
\draw[thick,red] (b)edge(e);
```

Konzept Node Chains: Label gleich Inhalt Mathemodus ?



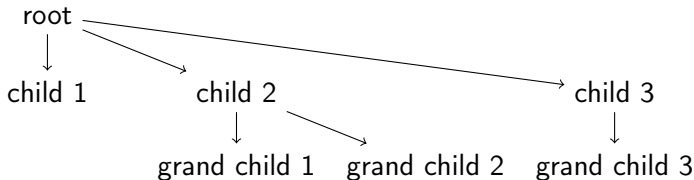
- ▶ mit `as=` Option
- ▶ oder mit `/`-Syntax

```
\graph {  
x1/$x_1$ -> x2 [as=$x_2$, red] -> x34/{$x_3,x_4$};  
x1 -> [bend left] x34;  
};
```



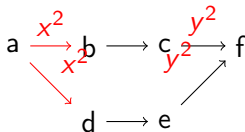
- ▶ mit klassischen Scopes { }

```
\graph {  
a -> {  
    b -> c,  
    d -> e  
}  
-> f  
};
```



```
\graph [grow down,  
branch right=2.5cm] {  
  root -> {  
    child 1,  
    child 2 -> {  
      grand child 1,  
      grand child 2  
    },  
    child 3 -> {  
      grand child 3  
    }  
  }  
};
```

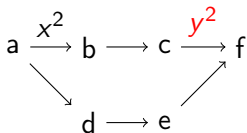
Gruppierungen und Kantenoptionen?



```
\graph {  
  a ->["$x^2$",red] {  
    b -> c,  
    d -> e  
  } ->["$y^2$"red] f  
};
```

Benötigt `\usetikzlibrary{quotes,babel}`

Gruppierungen und Kantenoptionen?



- ▶ Kantenoptionen können auch bei Endknoten spezifiziert werden
- ▶ $>$: eingehende Kanten der Gruppe
- ▶ $<$: ausgehende Kanten der Gruppe

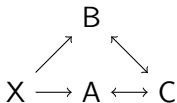
```
\graph {  
  a -> {  
    b[>"$x^2$"] -> c[<"$y^2$"red],  
    d -> e  
  } -> f  
};
```

Benötigt `\usetikzlibrary{quotes,babel}`

$A \longrightarrow B \longrightarrow C$

- Knoten können wie üblich referenziert werden

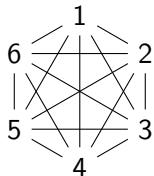
```
\node (a) at (0,0) {A};  
\node (b) at (1,0) {B};  
\node (c) at (2,0) {C};  
\graph { (a) -> (b) -> (c) };
```



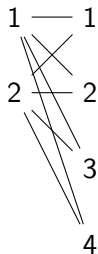
► oder mit Nodesets

```
\begin{scope}[new set=my nodes]
\node[set=my nodes] (a) at (1,0) {A};
\node[set=my nodes] (b) at (1,1) {B};
\node (c) at (2,0) {C};
\graph { X-> (my nodes) <-> (c) };
\end{scope}
```

Standard Graphen



```
\graph { subgraph K_n  
        [n=6, clockwise] };
```



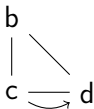
```
\graph { subgraph K_nm  
        [n=2, m=4 ] };
```

Benötigt wird hier: `\usetikzlibrary{graphs.standard}`

Color Classes

- ▶ wir können logische Klassen vergeben
- ▶ Connector-Syntax: Verbinde automatisch Knoten bestimmter Klassen (untereinander)

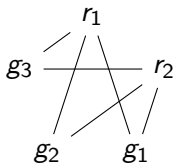
a



e

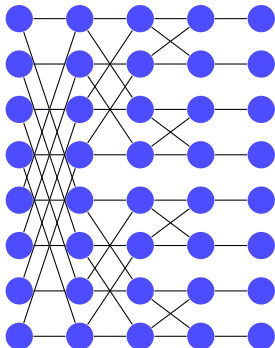
```
\tikz \graph [color class=red] { % rote Klasse eingefuehrt  
[cycle=red] % aus roten Knoten wird Zykel  
b [red], { [red] c ->[bend right] d }, e };
```

- ▶ Notiz: die Farbbedeutung "red" wird überlagert!



```
\tikz \graph [color class=red, color class=green,
math nodes, clockwise, n=5] {
[complete bipartite={red}{green}]
{ [red]
r_1, r_2 },
{ [green] g_1, g_2, g_3 }
};
```

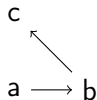
Connector ohne Colorclasses



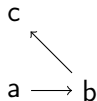
```
\tikz [x=8mm, y=6mm, circle]
\graph [nodes={fill=blue!70}, empty nodes, n=8] {
  subgraph I_n [name=A] --[butterfly={level=4}]
  subgraph I_n [name=B] --[butterfly={level=2}]
  subgraph I_n [name=C] --[butterfly]
  subgraph I_n [name=D] -- subgraph I_n [name=E]};
```

Etwas genauer: Platzierung der Knoten

- ▶ `\usetikzlibrary{graphs}`: online Platzierungsstrategien
(keine Information über zukünftige Teilgraphen)
- ▶ `\usetikzlibrary{graphdrawing}`: offline Platzierungsstrategien
Nachdem Graph spezifiziert wurde: finde beste Knotenpositionen
(Dokument muss `lualatex` interpretiert werden)
- ▶ bisher und in dieser VL: nur online!



```
\tikz \graph [no placement]
{
  a[at={(0:0)}]
    -> b[at={(1,0)}]
    -> c[yshift=1cm];
};
```

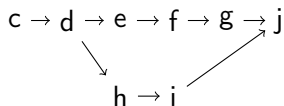


```
\tikz \graph [no placement]
{
  a[x=0,y=0]
    -> b[x=1,y=0]
    -> c[x=0,y=1];
};
```

Default: Cartesian placement

a

b

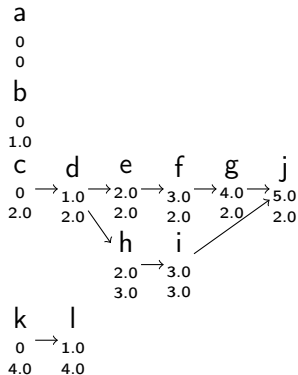


k → l

```
\tikz
\graph [grow right=7mm ]
{ a,
  b,
  c -> d -> {
    e -> f -> g,
    h -> i
  } -> j,
  k -> l };

```

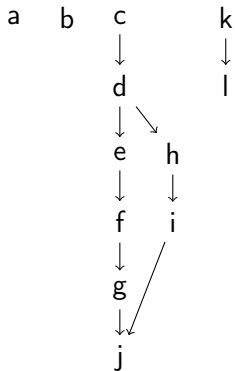
Default: Cartesian placement



```
\tikz
\graph [grow right=7mm
, ... zeige koordinaten ...]
{ a,
  b,
  c -> d -> {
    e -> f -> g,
    h -> i
  } -> j,
  k -> l    };
```

Default: Cartesian placement

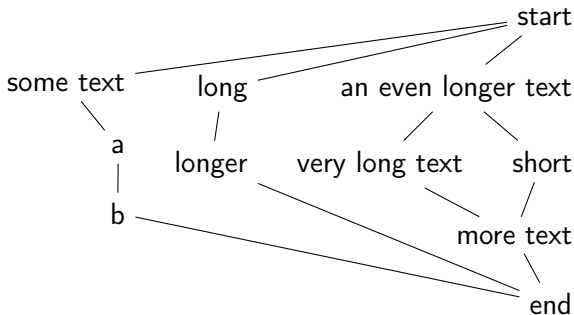
- ▶ `branch <left,right,up,down>=`: wie neue Chaingroups angelegt werden
- ▶ `grow <left,right,up,down>=`: wie Chaingroups wachsen



```
\tikz
\graph [branch right=7mm
        ,grow down=9mm]
{ a,
  b,
  c -> d -> {
    e -> f -> g,
    h -> i
  } -> j,
  k -> l    };
```

Default: Cartesian placement

- ▶ `branch <left,right,up,down> sep=`
- ▶ Knotengröße wird berücksichtigt, Abstand spezifiziert



```
\graph [grow down sep, branch left sep] {  
start -- { an even longer text -- {short, very long text}  
-- more text, long -- longer, some text -- a -- b  
} -- end };
```

Dieser Foliensatz ist **sehr** nah am pgfmanual 3.0.1a!

Dort:

- ▶ viel mehr Optionen
- ▶ viel mehr Möglichkeiten
 - ▶ kompakt zu spezifizieren
 - ▶ Stiele ect. differenziert anzupassen

- 1 Nachträge zu TikZ, beamer
 - Graph-Bibliothek
- 2 Erstellen von Plots
 - Plots in TikZ
 - Plots mit pgfplots
- 3 Weitere nützliche Klassen
- 4 Musterlösungen
- 5 Schluss
- 6 Organisation

- ▶ Viele Programme ermöglichen Visualisierung (Matlab, R, Mathematika, Excel, gnuplot, ...)
- ▶ Standardverfahren: pdf-Export und Import mit `\includegraphics`
- ▶ Probleme: Schriftart und -größe, Legenden, oft Export im Rasterformat (versteckt)

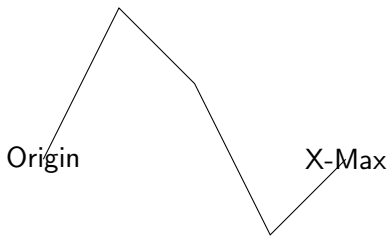
Alternative:

- ▶ Erstellen eines Plots in Latex (Ti~~k~~Z, pgfplots, gnuplot-Anbindung)
- ▶ Ermöglicht einheitlichen Stil, gute Qualität
- ▶ Nachteile: wie üblich ...

- 1 Nachträge zu TikZ, beamer
 - Graph-Bibliothek
- 2 Erstellen von Plots
 - Plots in TikZ
 - Plots mit pgfplots
- 3 Weitere nützliche Klassen
- 4 Musterlösungen
- 5 Schluss
- 6 Organisation

- ▶ Plot aus einzelnen Koordinaten (im $\text{T}_\text{E}\text{X}$ -Code)
- ▶ Plot Koordinaten einer externen Datei
- ▶ Plot einfacher Funktionen (evaluiert durch $\text{T}_\text{E}\text{X}$)
- ▶ Plot von Funktionen (evaluiert durch `gnuplot`)

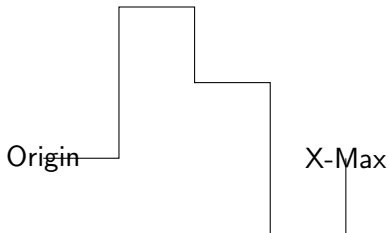
► plot coordinates { }



```
\draw (0,0) node{Origin} plot coordinates  
      {(0,0) (1,2) (2,1) (3,-1) (4,0)}  
      node{X-Max};
```

Die Plot-Pfadoperation

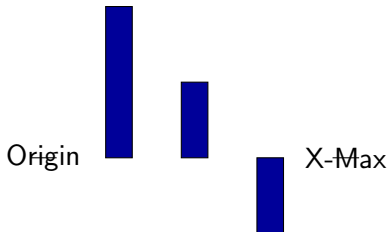
► `plot[Optionen] coordinates { }`



```
\draw (0,0) node{Origin} plot[const plot] coordinates
      {(0,0) (1,2) (2,1) (3,-1) (4,0)}
      node{X-Max};
```

Die Plot-Pfadoperation

► `plot[Optionen] coordinates { }`



```
\draw[fill=blue!60!black] (0,0) node{Origin}  
  plot[ybar] coordinates  
    {(0,0) (1,2) (2,1) (3,-1) (4,0)}  
  node{X-Max};
```

- ▶ `mark= <*,+,x>` oder `\usetikzlibrary{plotmarks}`
- ▶ `smooth` und weitere Optionen
- ▶ `const plot` und `jump mark`
- ▶ `ycomb` und `xcomb` (zusammen mit `color=`, `line width=`)
- ▶ `ybar` und `xbar` (Rechteckformen)

Plot aus Raw-Datei

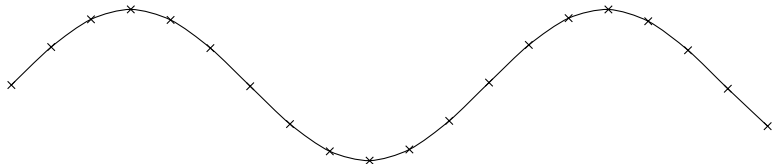
- ▶ **Ti***k***Z** unterstützt einfaches Dateiformat (nur engl.):
 - ▶ eine Zeile je Datenpunkt, Kommentarzeilen mit **#** oder **%**
 - ▶ Zahlen getrennt mit Leerzeichen
 - ▶ nur ersten beiden Zahlen werden berücksichtigt
- ▶ entspricht dem Datenformat von **gnuplot**

Inhalt von vl11-plots-bsp.txt:

```
#x y type
0.00000 0.00000 i
0.52632 0.50235 i
1.05263 0.86873 i
1.57895 0.99997 i
...
8.94737 0.45948 i
9.47368 -0.04889 i
10.00000 -0.54402 i
```

Die ‚i‘ werden ignoriert.

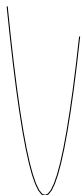
Plot aus Raw-Datei: Bsp.



```
\tikz \draw[mark=x,smooth] plot file{vl11-plots-bsp.txt};
```


Plot von Funktionen

- ▶ Es werden stets (intern) Datenpunkte berechnet (default: 25)
- ▶ Funktionsausdruck spezifiziert diese Datenpunkte explizit mit Variable `\x`:
Bsp.:
 - ▶ Ausdrücke mit Klammern (`...`) benötigen zusätzlich `{...}`



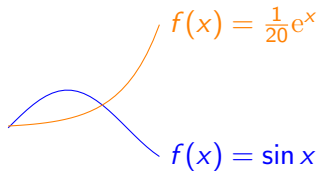
```
\tikz[scale=0.1] \draw plot (\x ,{(\x)*(\x)});
```

Optionen:

- ▶ `domain=xmin:xmax`
- ▶ `variable=\t`
- ▶ `samples=<number>` oder `samples at={ ... }`

Anbindung von gnuplot

- ▶ wird extern von \TeX aufgerufen
- ▶ es muss also gnuplot installiert sein!
- ▶ `pdflatex -shell-escape` (Linux)
(? oder `pdflatex -enable-write18` Windows ?)
- ▶ veränderte Syntax, insbesondere eindeutige `id=` Option !

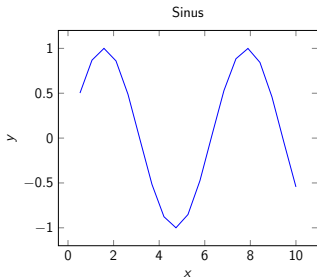


```
\tikz[domain=0:4,scale=0.5]{  
  \draw[color=blue]plot[id=sin]  
  function{sin(x)}node[right] {$f(x) = \sin x$};  
  \draw[color=orange] plot[id=exp] function{0.05*exp(x)}  
  node[right] {$f(x) = \frac{1}{20} \mathrm{e}^x$};}
```

- 1 Nachträge zu TikZ, beamer
 - Graph-Bibliothek
- 2 Erstellen von Plots
 - Plots in TikZ
 - Plots mit pgfplots
- 3 Weitere nützliche Klassen
- 4 Musterlösungen
- 5 Schluss
- 6 Organisation

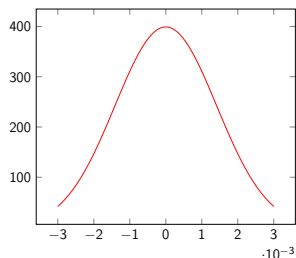
Paket: pgfplots.sty

- ▶ `\usepackage{pgfplots}` basiert auf `Ti \acute{k} Z`
- ▶ viele, viele typische Plotfunktionen (vgl. matlab)
- ▶ benötigt kein externes Programm
- ▶ Syntax sehr ähnlich zu `Ti \acute{k} Z`-internen Plots
- ▶ `axis`-Umgebung im `tikzpicture`, darin: `\addplot...`



```
\tikz[scale=0.5]{  
  \begin{axis}[  
    title=Sinus,  
    xlabel={ $x$ },  
    ylabel={ $y$ }]  
    \addplot[blue]  
      table {vl11-plots-bsp.txt};  
  \end{axis}}
```

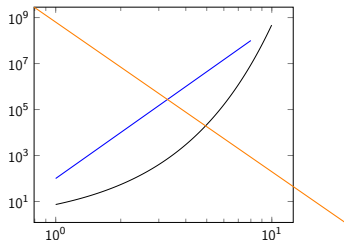
- Syntax: bessere Zahlendarstellung, Operatoren (z.B. \wedge) und weniger {...}-Paare



```
\tikz[scale=0.5]{\begin{axis}  
% density of Normal distribution:  
\addplot[red,  
domain=-3e-3:3e-3,  
samples=201]  
{exp(-x^2 / (2e-3^2)) / (1e-3 * sqrt(2*pi))};  
\end{axis}}
```

pgfplots spezielle Koordinaten

- ▶ `axis cs:` und `axis description cs:`
- ▶ `clip=` Option (default: true)



```
\tikz[scale=0.5]{  
  \begin{loglogaxis}[clip=false]  
    \addplot[  
      domain=1:10,  
      samples=201]  
      {(exp(x))^2};  
    \draw[thick, color=blue]  
      (axis cs: 1,100)--(axis cs: 8,10e+7);  
    \draw[thick, color=orange]  
      (axis description cs: 0,1)--  
      (axis description cs: 1.2,0);  
  \end{loglogaxis}}
```

- ▶ Tiks (Achsenbeschriftung, Abstände, Tik-Label, ...)
- ▶ viele Plotarten (2D-Kurven, 3D, Flächen, ybar, ...)
- ▶ verschiedene Achsumgebungen
- ▶ Grids, Grid-Linien
- ▶ Legenden, Farben
- ▶ Manual: \sim 500 Seiten

Tagesprogramm

- 1 Nachträge zu TikZ, beamer
 - Graph-Bibliothek
- 2 Erstellen von Plots
 - Plots in TikZ
 - Plots mit pgfplots
- 3 Weitere nützliche Klassen
- 4 Musterlösungen
- 5 Schluss
- 6 Organisation

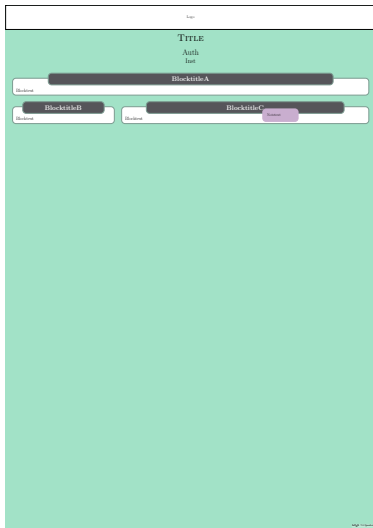
Beispiel-Poster

```
\documentclass[25pt, a0paper, portrait, margin=0mm  
    , innermargin=15mm, blockverticalspace=15mm  
    , colspace=15mm, subcolspace=8mm]{tikzposter}
```

► Handhabung ähnlich zu beamer

```
\title{Title} \institute{Inst} \author{Auth}  
\titlegraphic{Logo}  
\usetheme{Basic}  
\begin{document}  
\maketitle  
\block{BlocktitleA}{Blocktext}  
\begin{columns}  
    \column{0.3}  
        \block{BlocktitleB}{Blocktext}  
    \column{0.7}  
        \block{BlocktitleC}{Blocktext}  
        \note{Notetext}  
\end{columns} \end{document}
```

Minimales Beispiel



Wie ist es umgesetzt

- ▶ Das gesamte Poster ist in einer `tikzpicture` Umgebung
- ▶ Blöcke, ect. sind `TikZ`-Knoten (dynamische Größe) mit eigener Absatzumgebung.
- ▶ großes Papierformat, große Schriften, hohe Auflösung
- ▶ relativ neues Paket (bugs ?)

Konsequenzen:

- ▶ Problem: Wiederverwendung von Tikzbildern:
 - ▶ jetzt: Schrift zu groß
 - ▶ absolute Längenangaben
 - ▶ Manche Attribute werden von umgebenem Knoten geerbt
Ausweg(?): `Saveboxen` und `Scaleboxen`
- ▶ außerhalb von Blöcken: Text führt zu Fehlern
- ▶ außerhalb von Blöcken: eigene Pfade möglich:

```
\draw (0.4 \textheight, 0.4\textwidth) rectangle  
++(0.1 \textheight, 0.1\textwidth);
```

- ▶ `\usepackagecalc`
- ▶ Neudefinition der Kommandos
 - ▶ `\setlength`
 - ▶ `\addtolength`
 - ▶ `\setcounter`
 - ▶ `\addtocounter`

- ▶ erlaubt einfache arithmetische Ausdrücke

`\setlength{\foo}{\textwidth-2\fbboxsep-2\fbboxrule}`

- ▶ Pakete, die z.B. `\setlength` benutzen „erben“ das
 - ▶ z.B. die Breitenangabe bei `\begin{minipage}`

- ▶ `\usepackage{siunitx}`
 - ▶ viele Optionen
- ▶ Angabe von Größen mit Einheiten

`\SI{120}{\kilo\meter\per\hour}`

ergibt: 120 km h^{-1} `\num{9999999}` ergibt 9 999 999

- ▶ chemische Formeln
- ▶ Sudokus
- ▶ bessere `include` Optionen
- ▶ automatische Paketreihenfolge
- ▶ zusätzliche Symbole
- ▶ bessere Brüche
- ▶ ...

- 1 Nachträge zu TikZ, beamer
 - Graph-Bibliothek
- 2 Erstellen von Plots
 - Plots in TikZ
 - Plots mit pgfplots
- 3 Weitere nützliche Klassen
- 4 Musterlösungen**
- 5 Schluss
- 6 Organisation

- ▶ 10. HA 13 Stimmen
- ▶ 9. HA 12 Stimmen
- ▶ 8. 7. 5. Ha: 9,8,7 Stimmen

- 1 Nachträge zu TikZ, beamer
 - Graph-Bibliothek
- 2 Erstellen von Plots
 - Plots in TikZ
 - Plots mit pgfplots
- 3 Weitere nützliche Klassen
- 4 Musterlösungen
- 5 **Schluss**
- 6 Organisation

Es gibt eine Vielzahl von Programmen, mit denen man sich das Arbeiten mit \LaTeX vereinfachen kann. Manche davon haben sie eventuell bereits in den letzten Wochen kennengelernt.

Besonders beliebt sind

- ▶ \TeX -Editoren (von recht einfach bis sehr umfangreich) bzw. \TeX -Umgebungen für allgemeine Editoren (z.B. Vim, Emacs)
- ▶ Editoren für Bib \TeX -Dateien (z.B. JabRef)
- ▶ Grafikprogramme mit \LaTeX -Unterstützung (z.b. xfig), generell Vektorgrafiken vorziehen
- ▶ Konverter, z.B. LaTeX2html, Word2LaTeX, Excel2LaTeX

Es gibt eine Vielzahl von Programmen, mit denen man sich das Arbeiten mit \LaTeX vereinfachen kann. Manche davon haben sie eventuell bereits in den letzten Wochen kennengelernt.

Besonders beliebt sind

- ▶ \TeX -Editoren (von recht einfach bis sehr umfangreich) bzw. \TeX -Umgebungen für allgemeine Editoren (z.B. Vim, Emacs)
- ▶ Editoren für Bib \TeX -Dateien (z.B. JabRef)
- ▶ Grafikprogramme mit \LaTeX -Unterstützung (z.b. xfig), generell Vektorgrafiken vorziehen
- ▶ Konverter, z.B. LaTeX2html, Word2LaTeX, Excel2LaTeX

Es gibt eine Vielzahl von Programmen, mit denen man sich das Arbeiten mit \LaTeX vereinfachen kann. Manche davon haben sie eventuell bereits in den letzten Wochen kennengelernt.

Besonders beliebt sind

- ▶ \TeX -Editoren (von recht einfach bis sehr umfangreich) bzw. \TeX -Umgebungen für allgemeine Editoren (z.B. Vim, Emacs)
- ▶ Editoren für Bib \TeX -Dateien (z.B. JabRef)
- ▶ Grafikprogramme mit \LaTeX -Unterstützung (z.b. xfig), generell Vektorgrafiken vorziehen
- ▶ Konverter, z.B. LaTeX2html, Word2LaTeX, Excel2LaTeX

Es gibt eine Vielzahl von Programmen, mit denen man sich das Arbeiten mit \LaTeX vereinfachen kann. Manche davon haben sie eventuell bereits in den letzten Wochen kennengelernt.

Besonders beliebt sind

- ▶ \TeX -Editoren (von recht einfach bis sehr umfangreich) bzw. \TeX -Umgebungen für allgemeine Editoren (z.B. Vim, Emacs)
- ▶ Editoren für Bib \TeX -Dateien (z.B. JabRef)
- ▶ Grafikprogramme mit \LaTeX -Unterstützung (z.b. xfig), generell Vektorgrafiken vorziehen
- ▶ Konverter, z.B. LaTeX2html, Word2LaTeX, Excel2LaTeX

Es gibt eine Vielzahl von Programmen, mit denen man sich das Arbeiten mit \LaTeX vereinfachen kann. Manche davon haben sie eventuell bereits in den letzten Wochen kennengelernt.

Besonders beliebt sind

- ▶ \TeX -Editoren (von recht einfach bis sehr umfangreich) bzw. \TeX -Umgebungen für allgemeine Editoren (z.B. Vim, Emacs)
- ▶ Editoren für Bib \TeX -Dateien (z.B. JabRef)
- ▶ Grafikprogramme mit \LaTeX -Unterstützung (z.b. xfig), generell Vektorgrafiken vorziehen
- ▶ Konverter, z.B. LaTeX2html, Word2LaTeX, Excel2LaTeX

Im Internet sei auf folgende Quellen für weitere generelle Recherchen verwiesen:

- ▶ Deutschsprachige Anwendervereinigung TeX e.V.:
<http://www.dante.de>
- ▶ Comprehensive TeX Archive Network: <http://www.ctan.org>
- ▶ Recht umfangreiche Linksammlung zu verschiedenen LaTeX-Themen:
<http://www.siart.de/typografie/latexipps.xhtml>

Neben vielen anderen Druckwerken sind im Deutschsprachigen die LaTeX-Bücher (inzwischen 3 Bände) von Helmut Kopka als umfangreicheres Kompendium sehr zu empfehlen.

Dieser Kurs stellt eine Einführung in \LaTeX dar. Er vermittelt die grundlegenden Fertigkeiten, um eine (Abschluss-) Arbeit und Präsentationen zu verfassen. Darüber hinaus wird es immer wieder notwendig sein, zu bestimmten Fragestellungen die vorgestellten Websites zu befragen.

Weitergehende Informationen zur Typographie erhalten Sie auf folgenden Seiten:

- ▶ Eine Einführung in typographische Grundregeln
<http://www2.informatik.hu-berlin.de/sv/lehre/typographie.pdf>
- ▶ Beachten Sie auch bitte die \LaTeX -Tabus (nicht mehr zu verwendende \LaTeX -Befehle), das \LaTeX -Sündenregister:
<http://www.ctan.org/tex-archive/info/l2tabu/german/>

Vielen Dank für Ihre Aufmerksamkeit!
Und jetzt zur Organisation ...

- 1 Nachträge zu TikZ, beamer
 - Graph-Bibliothek
- 2 Erstellen von Plots
 - Plots in TikZ
 - Plots mit pgfplots
- 3 Weitere nützliche Klassen
- 4 Musterlösungen
- 5 Schluss
- 6 Organisation

Vorlesung, Hausaufgaben und betr. Rechnerzeiten

Mit dem Ende der Vorlesung ist diese Veranstaltung abgeschlossen.

Ab nächster Woche (1.2.++) werden keine weiteren

- ▶ Vorlesungen abgehalten,
- ▶ Hausaufgaben gestellt,
- ▶ betreuten Rechnerzeiten angeboten.

Je nach Bedarf werden aber noch Latex-Sprechstunden angeboten.

- ▶ Zur Zeit der VL am 1.2.2015 sowie 8.2.2015
- ▶ Hier können Sie Anregungen zu Ihren eigenen Arbeiten erhalten, Fragen stellen
- ▶ Falls gewünscht, tragen Sie sich in die Umfrage im StudIP ein

Weitere Ankündigungen (Fertigstellungstermin der Scheine zu dieser Veranstaltung, ...) erhalten Sie in den nächsten Wochen via (studip-Verteiler-) email.

⇒ Rufen Sie Ihr TU-Mailkonto am besten mindestens einmal pro Woche ab.

Vorlesung, Hausaufgaben und betr. Rechnerzeiten

Mit dem Ende der Vorlesung ist diese Veranstaltung abgeschlossen.

Ab nächster Woche (1.2.++) werden keine weiteren

- ▶ Vorlesungen abgehalten,
- ▶ Hausaufgaben gestellt,
- ▶ betreuten Rechnerzeiten angeboten.

Je nach Bedarf werden aber noch Latex-Sprechstunden angeboten.

- ▶ Zur Zeit der VL am 1.2.2015 sowie 8.2.2015
- ▶ Hier können Sie Anregungen zu Ihren eigenen Arbeiten erhalten, Fragen stellen
- ▶ Falls gewünscht, tragen Sie sich in die Umfrage im StudIP ein

Weitere Ankündigungen (Fertigstellungstermin der Scheine zu dieser Veranstaltung, ...) erhalten Sie in den nächsten Wochen via (studip-Verteiler-) email.

⇒ Rufen Sie Ihr TU-Mailkonto am besten mindestens einmal pro Woche ab.

Vorlesung, Hausaufgaben und betr. Rechnerzeiten

Mit dem Ende der Vorlesung ist diese Veranstaltung abgeschlossen.

Ab nächster Woche (1.2.++) werden keine weiteren

- ▶ Vorlesungen abgehalten,
- ▶ Hausaufgaben gestellt,
- ▶ betreuten Rechnerzeiten angeboten.

Je nach Bedarf werden aber noch Latex-Sprechstunden angeboten.

- ▶ Zur Zeit der VL am 1.2.2015 sowie 8.2.2015
- ▶ Hier können Sie Anregungen zu Ihren eigenen Arbeiten erhalten, Fragen stellen
- ▶ Falls gewünscht, tragen Sie sich in die Umfrage im StudIP ein

Weitere Ankündigungen (Fertigstellungstermin der Scheine zu dieser Veranstaltung, ...) erhalten Sie in den nächsten Wochen via (studip-Verteiler-) email.

⇒ Rufen Sie Ihr TU-Mailkonto am besten mindestens einmal pro Woche ab.

Zu den Scheinen ...

- ▶ Sämtliche Scheine werden schriftlich ausgestellt (voraussichtlich Anfang März).
- ▶ Die Veranstaltung hat einen Umfang von 2 SWS bzw. 3 Credit Points.

Die Hausaufgabenleistung gilt als bestanden, wenn folgende Kriterien erfüllt sind

- ▶ bei mindestens 10 Hausaufgaben jeweils mindestens ein Punkt erreicht wurde
- ▶ bei mindestens 8 Hausaufgaben jeweils mindestens die Hälfte der Punktzahl des Aufgabenblatts sowie
- ▶ eine Gesamtpunktzahl von mindestens 50% der erreichbaren Maximalpunktzahl aller Aufgabenblätter erreicht wurden.

Wer diese Kriterien erfüllt hat, erhält einen benoteten Schein.

Zu den Scheinen ...

- ▶ Sämtliche Scheine werden schriftlich ausgestellt (voraussichtlich Anfang März).
- ▶ Die Veranstaltung hat einen Umfang von 2 SWS bzw. 3 Credit Points.

Die Hausaufgabenleistung gilt als bestanden, wenn folgende Kriterien erfüllt sind

- ▶ bei mindestens 10 Hausaufgaben jeweils mindestens ein Punkt erreicht wurde
- ▶ bei mindestens 8 Hausaufgaben jeweils mindestens die Hälfte der Punktzahl des Aufgabenblatts sowie
- ▶ eine Gesamtpunktzahl von mindestens 50% der erreichbaren Maximalpunktzahl aller Aufgabenblätter erreicht wurden.

Wer diese Kriterien erfüllt hat, erhält einen benoteten Schein.

Zu den Scheinen ...

- ▶ Sämtliche Scheine werden schriftlich ausgestellt (voraussichtlich Anfang März).
- ▶ Die Veranstaltung hat einen Umfang von 2 SWS bzw. 3 Credit Points.

Die Hausaufgabenleistung gilt als bestanden, wenn folgende Kriterien erfüllt sind

- ▶ bei mindestens 10 Hausaufgaben jeweils mindestens ein Punkt erreicht wurde
- ▶ bei mindestens 8 Hausaufgaben jeweils mindestens die Hälfte der Punktzahl des Aufgabenblatts sowie
- ▶ eine Gesamtpunktzahl von mindestens 50% der erreichbaren Maximalpunktzahl aller Aufgabenblätter erreicht wurden.

Wer diese Kriterien erfüllt hat, erhält einen benoteten Schein.

Zu den Scheinen ...

- ▶ Sämtliche Scheine werden schriftlich ausgestellt (voraussichtlich Anfang März).
- ▶ Die Veranstaltung hat einen Umfang von 2 SWS bzw. 3 Credit Points.

Die Hausaufgabenleistung gilt als bestanden, wenn folgende Kriterien erfüllt sind

- ▶ bei mindestens 10 Hausaufgaben jeweils mindestens ein Punkt erreicht wurde
- ▶ bei mindestens 8 Hausaufgaben jeweils mindestens die Hälfte der Punktzahl des Aufgabenblatts sowie
- ▶ eine Gesamtpunktzahl von mindestens 50% der erreichbaren Maximalpunktzahl aller Aufgabenblätter erreicht wurden.

Wer diese Kriterien erfüllt hat, erhält einen benoteten Schein.

Viel Erfolg!

- ▶ bei möglichen anstehenden Prüfungen zu anderen Lehrveranstaltungen
- ▶ bei der möglichen Erstellung einer Abschlussarbeit
- ▶ (mit Hilfe von \LaTeX ?)

Vielen Dank für Ihre Aufmerksamkeit . . .

Viel Erfolg!

- ▶ bei möglichen anstehenden Prüfungen zu anderen Lehrveranstaltungen
- ▶ bei der möglichen Erstellung einer Abschlussarbeit
- ▶ (mit Hilfe von \LaTeX ?)

Vielen Dank für Ihre Aufmerksamkeit . . .