Zeichnen mit LaTeX

2. April 2010

Inhaltsverzeichnis

1 Nützliche Links

http://www.ctan.org/tex-archive/help/Catalogue/entries/pgf.html
http://www.texample.net/

2 Die benötigten Pakete

```
% Vorlage und globale Optionen
\documentclass
[
  draft      = true,
  fontsize   = 12pt,
  parskip    = half-,
  twoside    = false,
  dvipsnames % vermeidet 'option clash' mit xcolor
]
{scrartcl}

% Standardpakete
\usepackage[ascii]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[ngerman]{babel}
% Spezialpakete
\usepackage{fp}
\usepackage{tikz}
\usepackage{xcolor}
% TikZ-Bibliotheken
\usetikzlibrary{arrows}
\usetikzlibrary{shapes}
\usetikzlibrary{decorations.pathmorphing}
\usetikzlibrary{decorations.pathreplacing}
\usetikzlibrary{decorations.shapes}
\usetikzlibrary{decorations.text}
```

3 Die in `xcolor` vordefinierten Farben

4 Die erforderliche Umgebung

4.1 Vom Text abgesetzte Zeichnungen

PGF is a macro package for creating graphics. It is platform- and format-independent and works together with the most important TEX backend drivers, including pdftex and dvips. It comes with a user-friedly syntax layer called TikZ.

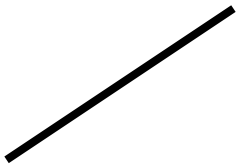


Abbildung 1: Die kürzeste Verbindung zwischen zwei Punkten

It comes with a user-friedly syntax layer called TikZ.

```
PGF is a macro package for creating graphics.
It is platform- and format-independent and works together with the most
important \TeX\ backend drivers, including pdftex and dvips.
It comes with a user-friedly syntax layer called TikZ.
\begin{figure}[H]
  \centering
  \begin{tikzpicture}
    \draw[line width=3pt] (0, 0) -- (3, 2);
  \end{tikzpicture}
  \caption{Die k\"urzeste Verbindung zwischen zwei Punkten}
  \label{fig:strecke}
\end{figure}
It comes with a user-friedly syntax layer called TikZ.
```

4.2 Zeichnungen im Text

PGF is a macro package for creating ⬭ graphics. It is platform- and format-independent and works together with the most important TEX backend drivers, including pdftex and dvips. It comes with a user-friedly syntax layer called TikZ.

```
PGF is a macro package for creating
\begin{tikzpicture}
  \draw[line width=0.8pt] (0, 0) ellipse (0.5em and 0.5ex);
\end{tikzpicture}
graphics.
It is platform- and format-independent and works together with the most
important \TeX\ backend drivers, including pdftex and dvips.
It comes with a user-friedly syntax layer called TikZ.
```

5 Die wichtigsten Befehle

```
% \draw[Optionen] <Pfad>;
% \fill[Optionen] <Pfad>;
\begin{tikzpicture}
  \draw[line width=1.5pt]
       (0, 0) -- (1, 0) -- (1, 1) -- (0, 1) -- cycle;
  \fill[fill=MidnightBlue]
       (2, 0) -- (3, 0) -- (3, 1) -- (2, 1) -- cycle;
\end{tikzpicture}
```

6 Manipulation von Linien


```
\draw[line width=5pt, draw=RubineRed]
     (0, 0) -- (2, 0);
```



```
\draw[line width=5pt, cap=round]
     (0, 0) -- (2, 0);
```



```
\draw[line width=1.5pt, style=dashed]
     (0, 0) -- (2, 0);
```



```
\draw[line width=1.5pt, style=densely dashed]
     (0, 0) -- (2, 0);
```



```
\draw[line width=1.5pt, style=loosely dashed]
     (0, 0) -- (2, 0);
```

```
\draw[line width=1.5pt, style=dotted]
     (0, 0) -- (2, 0);
```



```
\draw[line width=1.5pt, style=densely dotted]
     (0, 0) -- (2, 0);
```



```
\draw[line width=1.5pt, style=loosely dotted]
     (0, 0) -- (2, 0);
```

```
\draw[line width      = 5pt,
      draw            = Black,
      double          = LimeGreen,
      double distance = 5pt]
     (0, 0) -- (2, 0);
```

```
\draw[line width=1.5pt, decorate, decoration=zigzag]
     (0, 0) -- (2, 0);
```



```
\draw[line width=1.5pt, decorate, decoration=saw]
     (0, 0) -- (2, 0);
```



```
\draw[line width=1.5pt, decorate, decoration=bent]
     (0, 0) -- (2, 0);
```

```
\draw[line width=1.5pt, decorate, decoration=snake]
     (0, 0) -- (2, 0);
```

```
\draw[line width=1.5pt, decorate, decoration=coil]
     (0, 0) -- (2, 0);
```



```
\draw[line width=1.5pt, decorate, decoration=bumps]
     (0, 0) -- (2, 0);
```



```
\draw[line width=1.5pt, decorate, decoration=brace]
     (0, 0) -- (2, 0);
```



```
\draw[line width=1.5pt, decorate, decoration=ticks]
     (0, 0) -- (2, 0);
```



```
\draw[line width=1.5pt, decorate, decoration=border]
     (0, 0) -- (2, 0);
```



```
\draw[line width=1.5pt, decorate, decoration=waves]
     (0, 0) -- (2, 0);
```

```
\draw[line width=1.5pt,
      decorate,
      decoration=expanding waves]
     (0, 0) -- (2, 0);
```



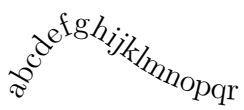
```
\draw[line width=1pt, decorate, decoration=triangles]
     (0, 0) -- (2, 0);
```



```
\draw[line width=1pt, decorate, decoration=crosses]
     (0, 0) -- (2, 0);
```



```
\draw[line width=1pt,
      decorate,
      decoration=shape backgrounds]
     (0, 0) -- (2, 0);
```




```
\draw[decorate,
      decoration={text along path,
                  text={abcdefghijklmnopqr}}]
     (0, 0) .. controls (1, 2) and (1, 0) .. (3, 0);
```



```
\draw[line width=1.5pt] (0, 1mm) -- (2, 1mm);
\draw[line width=1.5pt, shorten <=2mm, shorten >=5mm]
     (0, 0) -- (2, 0);
```

7 Pfeile

```
\draw[line width=1.5pt, ->]
     (0, 0) -- (1, 1) -- (3, 0);
```

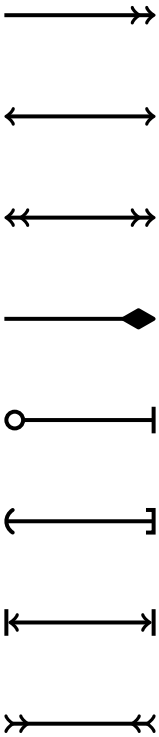


```
\draw[line width=1.5pt, ->, >=latex]
     (0, 0) -- (1, 1) -- (3, 0);
```

```
\draw[line width=1.5pt, ->, >=stealth]
     (0, 0) -- (1, 1) -- (3, 0);
```

```
\draw[line width=1.5pt, <-]
     (0, 0) -- (1, 1) -- (3, 0);
```

```
\draw[line width=1.5pt, <<-]
     (0, 0) -- (1, 1) -- (3, 0);
```



```
\draw[line width=1.5pt, ->>]
     (0, 0) -- (1, 1) -- (3, 0);
```

```
\draw[line width=1.5pt, <->]
     (0, 0) -- (1, 1) -- (3, 0);
```

```
\draw[line width=1.5pt, <<->>]
     (0, 0) -- (1, 1) -- (3, 0);
```

```
\draw[line width=1.5pt, ->, >=diamond]
     (0, 0) -- (1, 1) -- (3, 0);
```

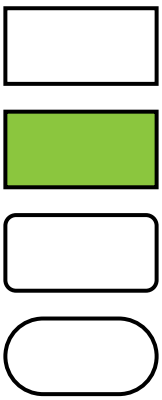
```
\draw[line width=1.5pt, o-|]
     (0, 0) -- (1, 1) -- (3, 0);
```

```
\draw[line width=1.5pt, (-{]}]
     (0, 0) -- (1, 1) -- (3, 0);
```

```
\draw[line width=1.5pt, |<->|]
     (0, 0) -- (1, 1) -- (3, 0);
```

```
\draw[line width=1.5pt, <<->>, >=to reversed]
     (0, 0) -- (1, 1) -- (3, 0);
```

8 Rechtecke



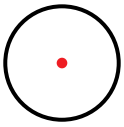
```
\draw[line width=1.5pt]
     (0, 0) rectangle (2, 1);
```

```
\draw[line width=1.5pt, fill=LimeGreen]
     (0, 0) rectangle (2, 1);
```

```
\draw[line width=1.5pt, rounded corners]
     (0, 0) rectangle (2, 1);
```

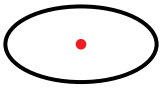
```
\draw[line width=1.5pt, rounded corners=5mm]
     (0, 0) rectangle (2, 1);
```

9 Kreise und Ellipsen

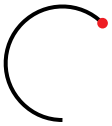


```
\draw[line width=1.5pt]
     (0, 0) circle (0.75);
\fill[fill=Red]
     (0, 0) circle (2pt);
```

```
\draw[line width      = 1pt,
      draw            = Black,
      double          = LimeGreen,
      double distance = 3mm]
     (0, 0) circle (0.75);
```



```
\draw[line width=1.5pt]
     (0, 0) ellipse (1 and 0.5);
```

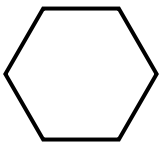


```
\draw[line width=1.5pt]
     (0, 0) arc (45:270:0.75);
```

```
\draw[line width=1.5pt]
     (0, 0) arc (45:270:1 and 0.5);
```

```
\draw[line width=1.5pt, xscale=-1]
     (0, 0) arc (45:270:1 and 0.5);
```

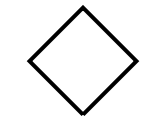
10 Polarkoordinaten



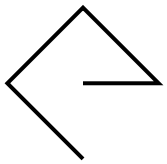
```
\draw[line width=1.5pt]
       (0:1cm) --  (60:1cm) -- (120:1cm) --
     (180:1cm) -- (240:1cm) -- (300:1cm) -- cycle;
```

11 Relative Koordinaten

```
\draw[line width=1.5pt]
     (0, 0) -- ++( 1, 0) -- ++(0,  1)
            -- ++(-1, 0) -- ++(0, -1);
```

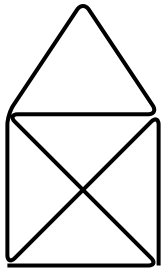


```
\draw[line width=1.5pt]
     (0, 0) -- ++( 45:1) -- ++(135:1)
            -- ++(225:1) -- ++(315:1);
```



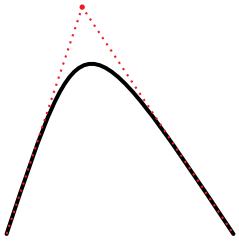
```
\draw[line width=1.5pt]
     (0, 0) -- +( 1, 0) -- +(0,  1)
            -- +(-1, 0) -- +(0, -1);
```

12 Koordinaten definieren

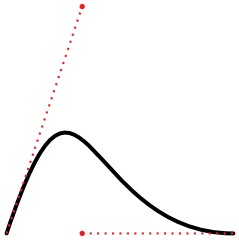


```
\coordinate (A) at (0, 0);
\coordinate (B) at (2, 0);
\coordinate (C) at (2, 2);
\coordinate (D) at (0, 2);
\coordinate (E) at (1, 3.5);
\draw[line width=1.5pt, rounded corners]
     (A) -- (B) -- (D) -- (C) --
     (E) -- (D) -- (A) -- (C) -- (B);
```

13 Bézierkurven



```
\coordinate (A) at (0, 0);
\coordinate (B) at (3, 0);
\coordinate (a) at (1, 3);
\draw[line width=1.5pt, cap=round]
     (A) .. controls (a) .. (B);
```



```
\coordinate (A) at (0, 0);
\coordinate (B) at (3, 0);
\coordinate (a) at (1, 3);
\coordinate (b) at (1, 0);
\draw[line width=1.5pt, cap=round]
     (A) .. controls (a) and (b) .. (B);
```

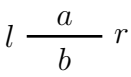
14 Beschriftung

14.1 Im Pfad

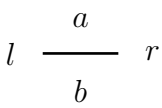
```
\draw[line width=1pt] (1, 0) node{$x$} -- (2, 0);
```

```
\draw[line width=1pt] (1, 0) -- node{$x$} (2, 0);
```

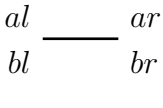
```
\draw[line width=1pt] (1, 0) -- (2, 0) node{$x$};
```



```
\draw[line width=1pt]
     (1, 0) node[left] {$l$} --
            node[above]{$a$}
            node[below]{$b$}
     (2, 0) node[right]{$r$};
```



```
\draw[line width=1pt]
     (1, 0) node[left=2mm] {$l$} --
            node[above=2mm]{$a$}
            node[below=2mm]{$b$}
     (2, 0) node[right=2mm]{$r$};
```



```
\draw[line width=1pt]
     (1, 0) node[above left] {$al$}
            node[below left] {$bl$} --
     (2, 0) node[above right]{$ar$}
            node[below right]{$br$};
```

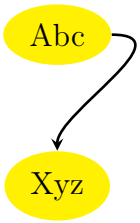
14.1.1 Einige `node`-Optionen

draw	fill	
rectangle diamond	circle	ellipse
text width	text centered	text badly centered
inner sep minimum width	outer sep minimum height	shape aspect

14.2 Eigenständig


```
\node[circle, fill=Yellow] at (0, 0) {Hallo, Welt};
```

14.3 Benannte `nodes`

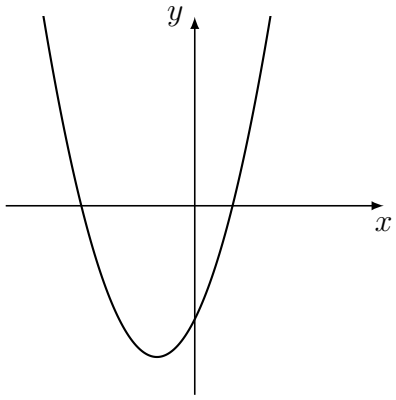



```
\node[ellipse, fill=Yellow] (abc) at (0,  1) {Abc};
\node[ellipse, fill=Yellow] (xyz) at (0, -1) {Xyz};
\draw[line width=1pt, ->, >=latex]
     (abc.0) .. controls +(0:1cm) and +(90:5mm) ..
     (xyz.90);
```

15 Clipping

```
\clip (-5mm, -1cm) rectangle (5mm, 1cm);
\node[ellipse, fill=Yellow] (abc) at (0,  1) {Abc};
\node[ellipse, fill=Yellow] (xyz) at (0, -1) {Xyz};
\draw[line width=1pt, ->, >=latex]
     (abc.0) .. controls +(0:1cm) and +(90:5mm) ..
     (xyz.90);
```

16 Kurven plotten

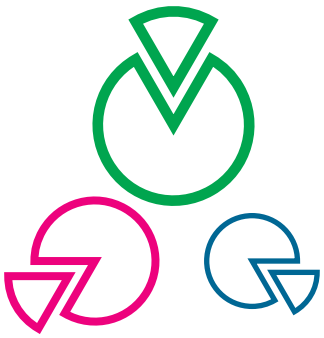



```
\draw[line width=1pt]
     plot[smooth]
     file{fx.table};
```

octave

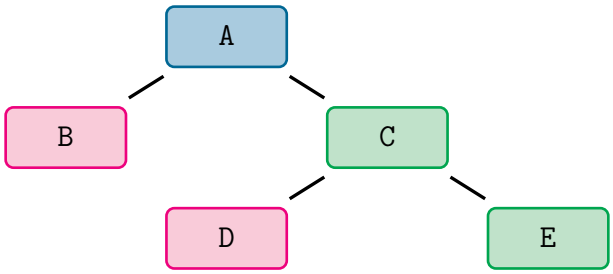
```
% Ausgabeformat definieren
format short g

% Vektor mit allen x-Werten erzeugen
x = [-5:0.1:5]';

% Vektor mit allen y-Werten berechnen
y = (x .+ 3) .* (x .- 1);

% kleine Werte auf Null setzen
for i = 1:rows(x)
  if (abs(x(i)) < 0.0001)
    x(i) = 0;
  end
  if (abs(y(i)) < 0.0001)
    y(i) = 0;
  end
end

% Wertetabelle exportieren
FID = fopen(''fx.table'', ''w'');
for i = 1:rows(x)
  fprintf(FID, ''%10.3f'', x(i));
  fprintf(FID, ''%10.3f'', y(i));
  fprintf(FID, ''\n'');
end
fclose(FID);
```

17 Die scope-Umgebung



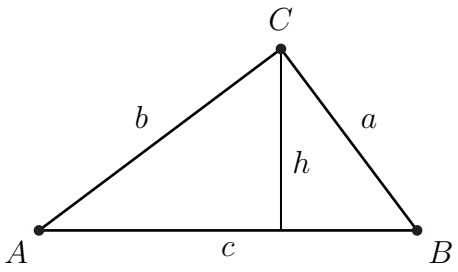
```
\begin{tikzpicture}
  \newcommand{\zeichnung}
  {
    \draw (0, 0) -- (30:1) arc (30:330:1) -- cycle;
    \begin{scope}[xshift=5mm, xscale=-1]
      \draw (0, 0) -- (150:1) arc (150:210:1) -- cycle;
    \end{scope}
  }
  \begin{scope}[line width = 4pt,
                draw       = Green,
                shift      = {(90:1.2cm)},
                rotate     = 90]
    \zeichnung
  \end{scope}
  \begin{scope}[line width = 3pt,
                draw       = RubineRed,
                shift      = {(210:1.2cm)},
                rotate     = 210,
                scale      = 0.8]
    \zeichnung
  \end{scope}
  \begin{scope}[line width = 2pt,
                draw       = MidnightBlue,
                shift      = {(330:1.2cm)},
                rotate     = 330,
                scale      = 0.6]
    \zeichnung
  \end{scope}
\end{tikzpicture}
```

18 Graphen




```
\begin{tikzpicture}
  \ttfamily
  \tikzstyle{every node}=
  [
    shape           = rectangle,
    rounded corners = 3pt,
    line width      = 1pt,
    text width      = 3em,
    text badly centered
  ];
  \tikzstyle{root}= [draw=MidnightBlue, fill=MidnightBlue!25!White];
  \tikzstyle{yes} = [draw=Green,        fill=Green!25!White];
  \tikzstyle{no}  = [draw=RubineRed,    fill=RubineRed!25!White];
  \newcommand{\rt}[2]{\node[root](#1){\rule[-0.8ex]{0pt}{2.8ex}#2};}
  \newcommand{\ys}[2]{\node[yes] (#1){\rule[-0.8ex]{0pt}{2.8ex}#2};}
  \newcommand{\no}[2]{\node[no]  (#1){\rule[-0.8ex]{0pt}{2.8ex}#2};}
  \matrix[row sep=5mm, column sep=5mm]
  {
                & \rt{A}{A}; &             &             \\
    \no{B}{B}; &             & \ys{C}{C}; &             \\
                & \no{D}{D}; &             & \ys{E}{E}; \\
  };
  \begin{scope}[line width=1.25pt, shorten >=2mm, shorten <=2mm]
    \draw (A) -- (B);
    \draw (A) -- (C);
    \draw (C) -- (D);
    \draw (C) -- (E);
  \end{scope}
\end{tikzpicture}
```

19 Rechnen mit LaTeX




```
\begin{tikzpicture}
  % Seitenlaengen
  \FPset{\a}{3}           % a = 3
  \FPset{\b}{4}           % b = 4
  \FPset{\c}{5}           % c = 5
  % Kathetensatz: p
  \FPmul{\p}{\a}{\a}      % p = a * a
  \FPdiv{\p}{\p}{\c}      % p = p / c
  % Kathetensatz: q
  \FPmul{\q}{\b}{\b}      % q = b * b
  \FPdiv{\q}{\q}{\c}      % q = q / c
  % Hoehensatz: h
  \FPmul{\h}{\q}{\p}      % h = p * q
  \FProot{\h}{\h}{2}      % h = 2-te wurzel aus h
  % Koordinaten
  \coordinate (A) at ( 0,  0);
  \coordinate (B) at (\c,  0);
  \coordinate (C) at (\q, \h);
  % Dreieck zeichnen
  \draw[line width=1pt]
       (A) -- node[below]       {$c$}
       (B) -- node[above right] {$a$}
       (C) -- node[above left]  {$b$}
       (A);
  % Hoehe zeichnen
  \draw[line width=0.75pt]
       (C) -- node[below right] {$h$} (\q, 0);
  % Punkte zeichnen
  \fill[fill=Black] (A) node[below left]  {$A$} circle (2pt);
  \fill[fill=Black] (B) node[below right] {$B$} circle (2pt);
  \fill[fill=Black] (C) node[above=3pt]   {$C$} circle (2pt);
\end{tikzpicture}
```

19.1 Einige Makros aus dem Paket `fp`

```
% Konstanten
\FPe                    % 2.718281828459045235
\FPpi                   % 3.141592653589793238
% Zuweisungen
\FPset    {\x}{2}       % x := 2
\FPset    {\y}{2.5}     % y := 2.5
% unaere Operationen
\FPabs    {\a}{\x}      % a := abs(x)
\FPneg    {\a}{\x}      % a := -x
% binaere Operationen
\FPadd    {\a}{\x}{\y}  % a := x + y
\FPsub    {\a}{\x}{\y}  % a := x - y
\FPmul    {\a}{\x}{\y}  % a := x * y
\FPdiv    {\a}{\x}{\y}  % a := x / y
\FPmin    {\a}{\x}{\y}  % a := min(x,y)
\FPmax    {\a}{\x}{\y}  % a := max(x,y)
% Nachkommastellen
\FPround {\a}{\x}{\y}  % a := x auf y Nachkommastellen gerundet
\FPtrunc {\a}{\x}{\y}  % a := x nach y Nachkommastellen abgeschnitten
\FPclip  {\a}{\x}      % a := x nur mit signifikanten Nachkommastellen
% Potenzen und Wurzeln
\FPpow    {\a}{\x}{\y}  % a := x^y
\FProot   {\a}{\x}{\y}  % a := x^(1/y)
% Trigonometrische Funktionen
\FPsin    {\a}{\x}      % a := sin(x)
\FPcos    {\a}{\x}      % a := cos(x)
\FPtan    {\a}{\x}      % a := tan(x)
\FPcot    {\a}{\x}      % a := cot(x)
\FParcsin{\a}{\x}       % a := arcsin(x)
\FParccos{\a}{\x}       % a := arccos(x)
\FParctan{\a}{\x}       % a := arctan(x)
\FParccot{\a}{\x}       % a := arccot(x)
% Exponential- und Logarithmusfunktion
\FPexp    {\a}{\x}      % a := exp(x)
\FPln     {\a}{\x}      % a := ln(x)
```

19.2 Berechnung von Termen

```
% Ausdruecke auswerten
\FPeval{\a}{inf}    % a := eval(Infix-Notation)
                    %
                    % Infix  : +,-,*,/,^
                    % Praefix: op(arg1, arg2, ...)

\FPupn{\a}{upn}     % a := eval(Umgekehrte Polnische Notation)
                    %
                    % +,add,-,sub,*,mul,/,div,abs,neg,min,max,
                    % round,trunc,clip,e,exp,ln,pow,root,pi,sin,cos,
                    % sincos,tan,cot,tancot,arcsin,arccos,arcsincos,
                    % arctan,arccot,arctancot,pop,swap,copy
                    %
                    % wobei:
                    %   pop   entfernt das oberste Element vom Stapel
                    %   swap  vertauscht die beiden obersten Elemente
                    %   copy  kopiert das oberste Element
```

19.3 Fallunterscheidungen

```
\FPiflt {\x}{\y} ... \else ... \fi   % ist (x < y) ?
\FPifeq {\x}{\y} ... \else ... \fi   % ist (x = y) ?
\FPifgt {\x}{\y} ... \else ... \fi   % ist (x > y) ?
\FPifint{\x}     ... \else ... \fi   % ist x eine ganze Zahl?
```

2 ist nicht kleiner als 1.5.

```
\FPset{\x}{2}
\FPset{\y}{1.5}

\FPiflt{\x}{\y}
  \x\ ist kleiner als \y.
\else
  \x\ ist nicht kleiner als \y.
\fi
```