

Mathematische Graphiken mit TikZ und PGFPLOTS

Markus Neher

Karlsruher Institut für Technologie
Institut für Angewandte und Numerische Mathematik

12.10.2017

- 1 TikZ von Till Tantau: TikZ ist kein Zeichenprogramm
Version 3.0.1a vom 29.08.2015. Basiert auf PGF.
Manual: 1161 Seiten.

- 2 PGFPLOTS von Christian Feuersänger: 2D und 3D
Funktionsgraphen
Version 1.15 vom 05.06.2017. Basiert auf TikZ.
Manual: 566 Seiten.

Versionen bis PGFPLOTS 1.10 interpretieren gewisse
Koordinaten anders (z.B. Radien von Kreisen).

- 1 Die Pakete bestehen je aus einem Grundpaket und Bibliotheken, die einzeln eingebunden werden können.

```
\usepackage{tikz}
\usetikzlibrary{arrows}
\usetikzlibrary{patterns}
\usepackage{pgfplots}
\pgfplotsset{compat=1.15,
  /pgfplots/every axis/.style={
    axis lines=center, scale only axis=true,
    enlargelimits=auto, font={\fs},
    xlabel style={below}, ylabel style={left},
    xlabel=\hspace*{-1.0ex}\raisebox{-1.5ex}{$x$},
    ylabel=\raisebox{-2.5ex}{$y$} },
  /pgfplots/every tick/.style={black} }
\usepgfplotslibrary{fillbetween}
```

- 2 TikZ-Graphiken stehen in einer `picture`-Umgebung, PGFPLOTS-Graphiken meist in einer `axis`-Umgebung.

```
\begin{tikzpicture}  
  < Befehle >  
\end{tikzpicture}
```

```
\begin{tikzpicture}  
  \begin{axis}[ Optionen ]  
    < Befehle >  
  \end{axis}  
\end{tikzpicture}
```

- 1 Linien, Kreise und Bögen
- 2 Kurven
- 3 Ebene Flächen
- 4 3D-Plots von Funktionsgraphen
- 5 Auslagern von Graphiken mit `externalize`
- 6 Plot-Export aus Matlab: `matlab2tikz`
- 7 Schleifen mit `\foreach`
- 8 Programmierung mit der TikZ-Bibliothek `math`
- 9 Animationen mit `animate`

Linien, Kreise und Bögen

Beispiel 1: Linien in TikZ



```
\draw (0,0) -- (2,0);
```



```
\draw[blue, thin, ->, >=latex] (0,0) -- (2.1,0);
```



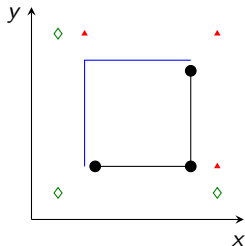
```
\draw[very thick, <->, >=stealth] (0,0) -- (2,0);
```



```
\draw[red, line width=2.5pt, dashed] (0,0) -- (2,0);
```

```
\node[above right] at (1,0) { \small Linien };
```

Beispiel 2: Linien in PGFPLOTS



```
\begin{tikzpicture}
  \begin{axis}[...]
    %
    \addplot[blue, mark=none]
      coordinates{ (0.5,0.5) (0.5,1.5) (1.5,1.5) };
    \addplot[mark=*]
      coordinates{ (0.6,0.5) (1.5,0.5) (1.5,1.4) };
    \addplot[red, only marks, mark=triangle*, mark size=1pt]
      coordinates{ (1.75,0.5) (1.75,1.75) (0.5,1.75) };
    \draw[darkgreen] plot[only marks, mark=diamond]
      coordinates{ (1.75,0.25) (0.25,0.25) (0.25,1.75) };
    %
  \end{axis}
\end{tikzpicture}
```

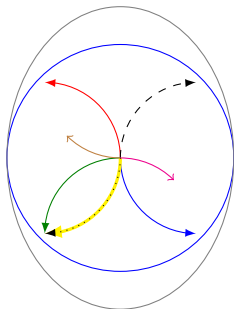

Beispiel 3: Bögen

Syntax: `arc(Winkel1 : Winkel2 : Radius)`

$w_2 > w_1$: Linkskurve, $w_2 < w_1$: Rechtskurve

Wählt man bei Linkskurven $r > 0$ und bei Rechtskurven $r < 0$, dann ist w_1 der Winkel, der im Anfangspunkt zwischen Kreisbogen und positiver y -Achse eingeschlossen wird (Polarwinkel+90°).

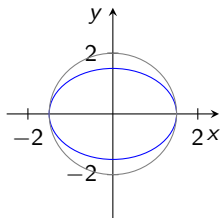
Radius $\times (-1)$ oder Winkel+180 Grad: Drehung um 180 Grad



```
\draw[blue] (0,0) circle (1.5);  
\draw[gray] (0,0) circle (1.5 and 2);  
  
\draw[red, ->, >=latex] (0,0) arc ( 0: 90: 1);  
\draw[blue, ->, >=latex] (0,0) arc ( 0: 90:-1);  
\draw[darkgreen, ... ] (0,0) arc ( 90:180: 1);  
  
\draw[black, dashed,...] (0,0) arc ( 0:-90:-1);  
\draw[yellow, ... ] (0,0) arc ( 0:-90: 1);  
\draw[black, dotted,...] (0,0) arc (180: 90:-1);  
  
\draw[brown, -> ] (0,0) arc ( 90: 45:-1);  
\draw[magenta, -> ] (0,0) arc ( 90: 45: 1);
```

Beispiel 4: Bögen in axis-Umgebung

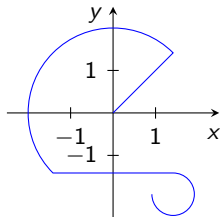
Verzerrung bei unterschiedlicher Skalierung der Achsen



```
\draw[blue] (0,0) circle (1.5);  
\draw[gray] (0,0) circle (1.5 and 2);
```

Beispiel 5: Anstückeln von Bögen

Beliebig:



```
\draw[blue] (0,0) -- (1.414,1.414) arc (45:225:2)  
-- (1.414,-1.414) arc (270:0:-0.5);
```

Beispiel 6: Dekorationen



```
\draw (0,0) -- (2,0);  
\draw[decorate, decoration={brace,mirror,raise=2pt}]  
      (0,0) -- (2,0);
```



```
\draw[decorate, decoration={zigzag, pre length=2.5  
      post length=2.5, segment length=7.5, amplitude=-2.5}]  
      (0,0) -- (2,0);
```



```
\draw[decorate, decoration={zigzag, pre length=8  
      post length=12, segment length=5, amplitude=1.5}]  
      (0,0) -- (2,0);
```



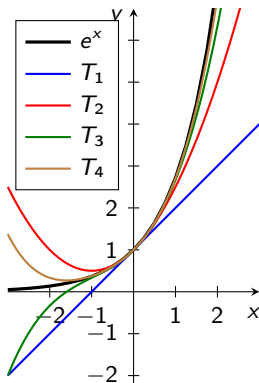
```
\draw[decorate, decoration={snake,amplitude=1pt}]  
      (0,0) -- (2,0);
```



```
\draw[decorate, decoration=bumps, segment length=15.5]  
      (0,0) -- (2,0);
```

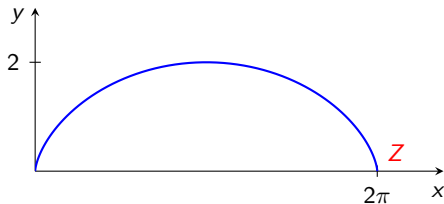
Kurven

Überlagerte Graphen mit Legende



```
\begin{tikzpicture}
  \begin{axis}[width=0.8\textwidth, height=...,
    xmin=-3, xmax=3, ymin=-2.25, ymax=6.75,
    xtick={-2,...,2}, ytick={-2,...,6},
    legend pos=north west]
    %
    \addplot[samples=50, domain=-3:3] { exp(x) };
    \addplot[blue, thick, domain=-3:3] { 1+x };
    ...
    \legend{ $e^x$, $T_1$, $T_2$, $T_3$, $T_4$ };
    %
  \end{axis}
\end{tikzpicture}
```

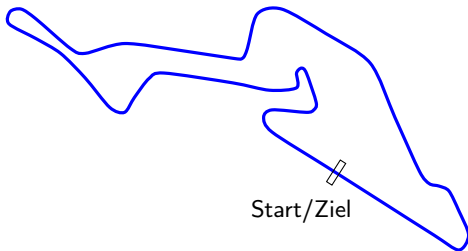
Kurve in Parameterdarstellung



```
\begin{tikzpicture}
  \begin{axis}[width=7cm, height=3cm, xmin=0, xmax=7, ymin=0, ymax=3,
    xtick={6.2832}, ytick={2}, xticklabels={{2\pi}}]
    \addplot[blue, thick, samples=100, domain=0:6.2832]
      ( { x - sin( deg(x) ) }, { 1 - cos( deg(x) ) } );
    \node[red, above right] at (axis cs:6.28,0) { $Z$ };
  \end{axis}
\end{tikzpicture}
```

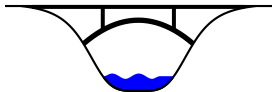
3D-Kurve analog.

Glatte Kurve durch Interpolation



```
\begin{tikzpicture}[scale=0.065] % Nürburgring
  \draw[color=blue, very thick] plot[smooth cycle]
    coordinates{ (58,28) (69,21) (80,14) (91, 7) (96, 4) (98, 7)
      (96,12) (94,16) (92,17) (90,17.5) (88,20) (83,31)
      (... )
      (66.5,32.5) (65,32) (58,32.5) (56.3,31) };
  \draw (69.1,17.9)--(70.2,17.2)--(73,21.6)--(71.9,22.3)--(69.1,17.9);
  \node at (64,12) {\fs Start/Ziel};
\end{tikzpicture}
```


Flächen zwischen Graphen füllen mit fill between



```
\addplot[blue, thick, decorate,  
  decoration={snake,amplitude=1pt}, name path=A]  
  coordinates { (-0.65,-1.59) (0.65,-1.59) };  
\addplot[thick, domain=-0.65:0.65, samples=100,  
  name path=B] { -2/(1+x^4) + 0.1 };  
\addplot[blue] fill between [of=A and B];
```

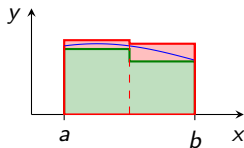
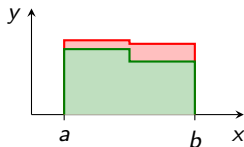
Ebene Flächen

Beispiel 1: Mit draw



```
\begin{tikzpicture}[scale=1.25]
  % Rechteck
  \draw[blue, thick, fill=gray!25!white] (0.5,0) rectangle (1.5,1.5);
  % Dreieck
  \draw[blue, thick, fill=gray!25!white] (3,0) -- (5,0) -- (4.5,1.5) -- (3,0);
\end{tikzpicture}
```

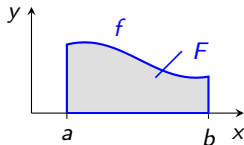
Beispiel 2: Mit addplot



```
\begin{axis}[...]
% Ober- und Untersumme
\addplot[red, thick, fill=red!25!white]
coordinates
{ (1,0) (1,2.1) (2,2.1) (2,2) (3,2) (3,0) };
\addplot[darkgreen, thick, fill=darkgreen!25!white]
coordinates
{ (1,0) (1,1.85) (2,1.85) (2,1.5) (3,1.5) (3,0) };

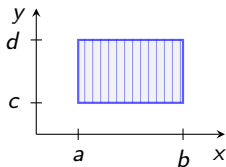
% Funktionsgraph
\addplot[blue, domain=1:3] { 1.5 + cos(deg(x-1.5))/2 };
% Rand links - unten - rechts
\addplot[red, thick] coordinates
{ (1,1.939) (1,0) (3,0) (3,2) };
% Rechtecke
\draw[red, dashed] (axis cs:2,0) -- (axis cs:2,2);
```

Beispiel 3: Mit Funktionsplot

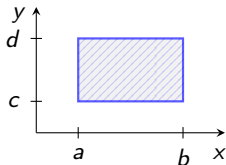


```
\addplot[blue, thick, fill=gray!25!white, domain=1:5]  
{ 1.5 + cos(deg(x-1.5))/2 } \closedcycle;
```

Schraffierte Flächen mit pattern



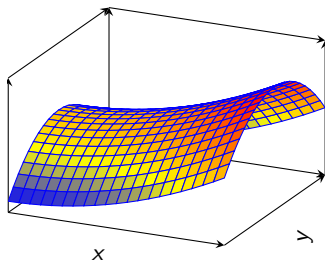
```
\draw[blue, thick, fill=gray!25!white, opacity=0.4]  
  (axis cs:1,0.75) rectangle (axis cs:3.5,2.25);  
\draw[blue, thick, pattern=vertical lines,  
  pattern color=blue, opacity=0.4]  
  (axis cs:1,0.75) rectangle (axis cs:3.5,2.25);
```



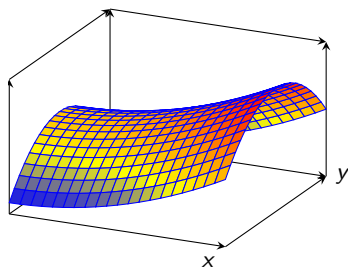
```
pattern=north east lines
```

3D-Plots

3D-Plot mit $z = f(x, y)$: axis lines=box

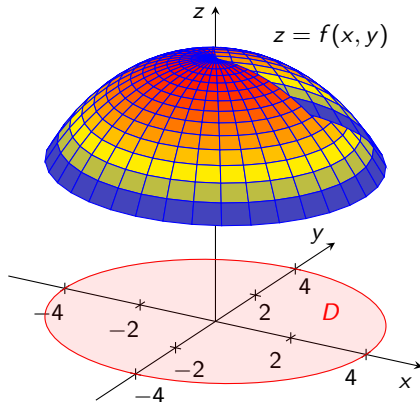


```
\begin{axis}[width=..., height=...,  
  axis lines=box,  
  xtick=\empty, ..., ztick=\empty]  
%  
\addplot3[surf, faceted color=blue,  
  samples=21, domain=0:1, y domain=-1:1]  
{ x^2 - y^2 };
```



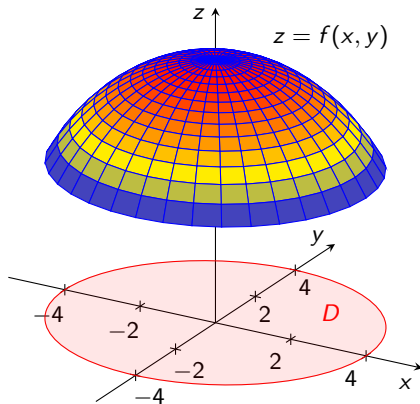
```
\begin{axis}[..., axis lines=box,  
  clip=false,  
  xlabel=\empty, ylabel=\empty,  
  ..., ztick=\empty]  
\addplot3[...] { x^2 - y^2 };  
\node[below left]  
  at (rel axis cs:1,0,0) {  $x$  };  
\node[right] at (rel axis cs:1,1,0) {  $y$  };
```


3D-Plot mit Parameterdarstellung: axis lines=center



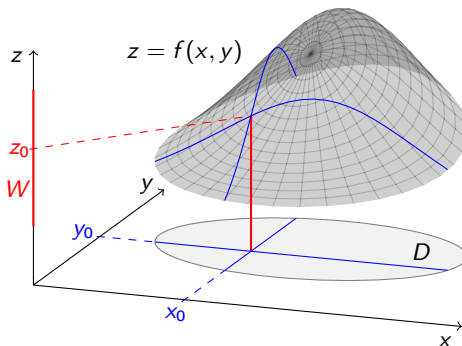
```
\begin{axis}[width=0.8\textwidth, clip=false, view={30}{30},  
  axis lines=center, tick label style={font=\fs}, ...]  
  \draw[fill=red, opacity=0.1] (axis cs:0,0,0) circle (4);  
  \draw[red] (axis cs:0,0,0) circle (4);  
  \addplot3[surf, faceted color=blue, ... domain=0:4, y domain=0:360]  
    ( { x*cos(y) }, { x*sin(y) }, { sqrt(25-x^2) } );
```

3D-Plot mit Parameterdarstellung: axis lines=center



```
\begin{axis}[width=0.8\textwidth, clip=false, view={30}{30},  
  axis lines=center, tick label style={font=\fs}, z buffer=sort, ...]  
  \draw[fill=red, opacity=0.1] (axis cs:0,0,0) circle (4);  
  \draw[red] (axis cs:0,0,0) circle (4);  
  \addplot3[surf, faceted color=blue, ... domain=0:4, y domain=0:360]  
    ( { x*cos(y) }, { x*sin(y) }, { sqrt(25-x^2) } );
```

3D-Plot mit Parameterdarstellung: manuelle Achsen



```
\begin{axis}[..., view={20}{30}, axis lines=none, ...]  
% Achsen  
\draw[>] (axis cs:-2,-2,0) -- (axis cs:2.25,-2,0);  
\node[below left] at (axis cs:2.25,-2,0) {  $x$  }; ...  
% Kurven  
\addplot3[blue, samples y=0, domain=-1.45:1.45]  
    ( x, -0.25, { 1 / ( 1.25 + 4*x^2/9 - x/6 ) } );  
\addplot3[blue, samples y=0, domain=-0.945:0.945]  
    ( -0.5, x, { 1 / ( 43/36 + x^2 - 3*x/4 ) } );
```

Auslagern von Graphiken mit externalize

- Motivation: Graphiken zu übersetzen kostet Rechenzeit.
- Idee: Exportiere übersetzte Graphiken in separate PDF-Dateien, binde diese wieder ein, wenn sie nicht geändert wurden.
- Der KIT-Style sabotiert `externalize`, weil jede Seite als Grapik gesetzt wird.

- Empfehlung: Verzeichnis für separate Graphiken anlegen, z.B. Unterverzeichnis `figures`.
- PGFPLOTS-Bibliothek `external` im Dokument einbinden, `externalize`-Befehl aufrufen:

```
\usepgfplotslibrary{external}  
\tikzexternalize[prefix=figures/]
```

- Übersetzen auf der Kommandozeile mit Option `shell-escape`:

```
pdflatex --shell-escape dateiname.tex
```

(Einmalig bzw. nach Änderungen von Graphiken.)

- Lästig: Übersetzt man geänderte Graphiken versehentlich ohne `shell-escape`, wird die Änderung danach nicht mehr erkannt.

Abhilfe: Graphik-Datei löschen oder erneut Graphik ändern und neu übersetzen.

- In Verbindung mit pdfLatex schneidet `externalize` die bounding box der Graphik ab.

Abhilfe (Stand September 2017): unbefriedigend.

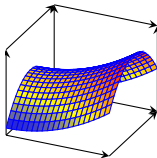
externalize: Probleme



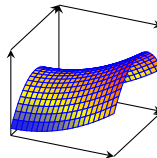
```
\draw[blue, <->, >=latex] (0,0) -- (2,0);
```



```
\draw[blue, <->, >=latex] (0,0) -- (2,0);  
\node[fit=(current bounding box), inner sep=0.5ex]{};
```



```
\begin{tikzpicture} \begin{axis} ... \end{axis}  
\end{tikzpicture}
```



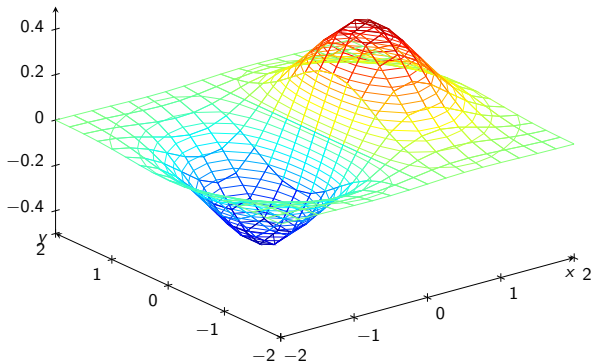
```
\begin{tikzpicture} \begin{axis} ... \end{axis}  
\node[fit=(current bounding box), inner sep=0.5ex]{};  
\end{tikzpicture}
```


Plot-Export aus Matlab: matlab2tikz

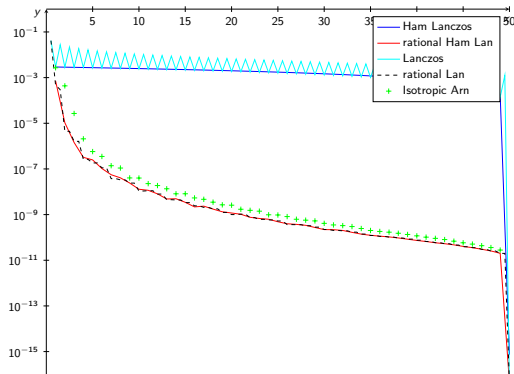
- 1 Mit dem Matlab-Skript matlab2tikz von Nico Schlömer lassen sich Plots aus Matlab exportieren.
- 2 Version 1.0.0 (09/2017): <http://www.mathworks.com/matlabcentral/fileexchange/22022-matlab2tikz>
- 3 Jeder Plot wird in eine separate Textdatei geschrieben, welche editiert werden kann.
- 4 Danach bindet man die Plot-Dateien entweder mit `\input` ein oder kopiert sie in das Texfile.

Matlab-Plot: Beispiel 1

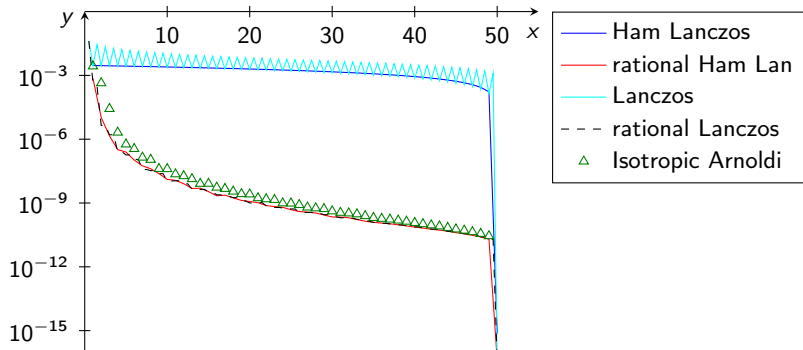
```
[X,Y] = meshgrid([-2:.25:2]);  
Z = X.*exp(-X.^2-Y.^2);  
contour3(X,Y,Z,30)  
surface(X,Y,Z,'EdgeColor',[.8 .8 .8],'FaceColor','none')  
grid off  
matlab2tikz;
```



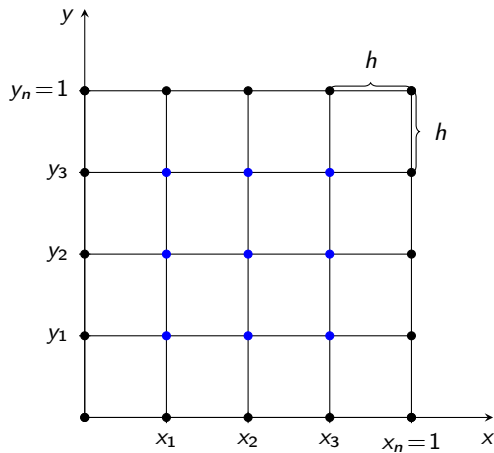
Beispiel 2: Original



Beispiel 2: Editiert

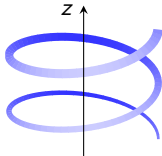


Programmierung mit foreach



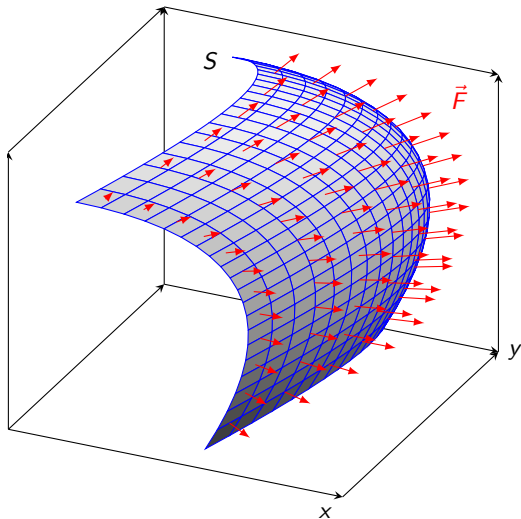
```
\begin{axis}[...]
  \draw[decorate, decoration={brace,raise=0.5pt}]
    (axis cs:3,4) -- (axis cs:4,4);
  \node[above] at (axis cs:3.5,4.15) {  $\$$  };
  % Gitter und Randpunkte
  \foreach \x in {0,...,4}
  { \edef\temp{ % Makro \temp wegen axis-Befehl
    \noexpand \draw (axis cs:\x,0) -- (axis cs:\x,4);
    \noexpand \draw (axis cs:0,\x) -- (axis cs:4,\x);
    \noexpand \filldraw (axis cs:\x,0) circle (1.5pt);
    \noexpand \filldraw (axis cs:\x,4) circle (1.5pt);
    \noexpand \filldraw (axis cs:0,\x) circle (1.5pt);
    \noexpand \filldraw (axis cs:4,\x) circle (1.5pt); }
    \temp }
  % Innere Gitterpunkte
  \foreach \x in {1,2,3}
  \foreach \y in {1,2,3}
  { \edef\temp{ % Makro \temp wegen axis-Befehl
    \noexpand \filldraw[blue] (axis cs:\x,\y) circle (1.5pt); }
    \temp }
\end{axis}
```


Kurve mit Variation von Farbe und Durchmesser



```
\foreach \x
  [evaluate=\shade using 50+30*sin( deg(\x) )]
  [evaluate=\w using sqrt( 1+\x )]
in {0,0.1,...,12.5} {
  \edef\temp{ % Makro \temp wegen Auswertung von \w und \x
    \noexpand \addplot3[blue!\shade!white, line width=\w pt,
      domain=\x:\x+0.11, samples y=0]
      ( { cos( deg(x) ) }, { sin( deg(x) ) }, { x } );
  } \temp }
% samples y=0 ist notwendig, um eine Kurve zu plotten
```

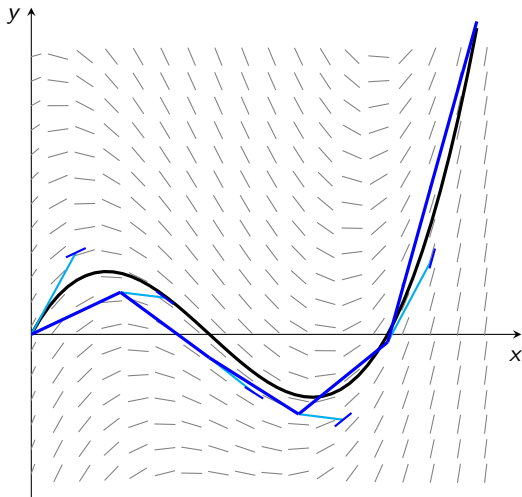
3D-Plot mit Vektorfeld



3D-Plot mit Vektorfeld

```
\begin{axis}[..., colormap={bw}{gray(0cm)=(0.2); gray(1cm)=(0.95)}, ...]
%
\addplot3[surf, faceted color=blue, samples=21,
    domain=-1:2, y domain=-1:1]
    ( { 0.707*( x + 4 - x^2/2 - y^2) }, {y},
      { 0.707*( -x + 4 - x^2/2 - y^2) } );
\foreach \x [evaluate=\xstart using 0.707*(\x+4-((\x)^2)/2)]
    [evaluate=\xtip using 0.778*(\x+4-((\x)^2)/2)]
    [evaluate=\zstart using 0.707*(-\x+4-((\x)^2)/2)]
    [evaluate=\ztip using 0.778*(-\x+4-((\x)^2)/2)]
in {-0.9,-0.65,...,1.85} {
    \foreach \y [evaluate=\ysqr using 0.707*((\y)^2)]
        [evaluate=\ysqtip using 0.778*((\y)^2)]
        [evaluate=\ystart using \y]
        [evaluate=\ytip using 1.1*\y]
    in {-0.95,-0.7,...,-0.2,0.1,0.5} {
        \edef\temp{ % Makro \temp wegen axis-Befehl
            \noexpand \draw[red, ->, >=latex]
                (axis cs:\xstart-\ysqr,\ystart,\zstart-\ysqr) --
                (axis cs:\xtip-\ysqtip,\ytip,\ztip-\ysqtip); }
        \temp } } % foreach \x
\end{axis}
```

Verfahren von Runge



Verfahren von Runge

```
% DGL:  $x' = f(x,y) = 1$ ;  $y' = g(x,y) = -y+x^3-4x+2$ 
\foreach \y in {-0.85,-0.7,...,1.85} {
  \foreach \x [evaluate=\dxx using 1]
    [evaluate=\dyy using  $-\y+(\x)^3-4*\x+2$ ]
    [evaluate=...] in {0,0.15,...,2.56} {
    \edef\temp{ \noexpand \draw[gray] (axis cs:\s,\u) -- (axis cs:\t,\v); }
    \temp } } % \foreach \y

% Lösung
\addplot[very thick, samples=50, domain=0:2.5] { x*(x-1)*(x-2) };

% Verfahren von Runge zu Anfangswerten ( $x,y$ )=(0,0)
\foreach \x/\y [evaluate=\h using 0.5] in {0/0} {
  \foreach \n [evaluate=\u using \unext]
    [evaluate=\v using \vnext]
    [evaluate=\uu using  $\u + \h/2$ ]
    [evaluate=...]
    [evaluate=\tnext using  $\vv+\h*\dv/8$ ]
    [remember=\unext (initially \x)]
    [remember=\vnext (initially \y)]
  in {1,...,5}
  { \edef\temp{
    \noexpand \draw[cyan, thick] (axis cs:\u,\v) -- (axis cs:\uu,\vv);
    \noexpand \draw[blue, thick] (axis cs:\s,\t) -- (axis cs:\snext,\tnext);
    \noexpand \draw[blue, very thick] (axis cs: ... \vnext); }
    \temp } } % \foreach \x/\y
```

- ❶ Die Arithmetik hat nur ca. 5 Dezimalstellen.
- ❷ Numerischen Überlauf vermeiden, evtl. skalieren.
- ❸ Statt $\backslash x^2$ stets $(\backslash x)^2$ verwenden, da bei negativen Werten das Vorzeichen sonst nicht quadriert wird.

Programmierung mit der TikZ-Bibliothek math

- Variablen und arrays
- Benutzerdefinierte Funktionen
- Schleifen und Verzweigungen
- Sehr zickig bei Syntaxfehlern, unklare Fehlermeldungen
- Verwendet PGF math engine und damit reines Tex für alle Berechnungen \leadsto langsam und ungenau (5 Dezimalstellen)
- Genauere Arithmetik (reine Berechnungen, in Graphiken nur eingeschränkt nutzbar): fp-Bibliothek (fixed point arithmetic), fpu-Bibliothek (float)

- Radiodromen
- Wellengleichung

Animationen mit animate

- Das Latex-Paket animate ermöglicht steuerbare Animationen
- Verzeigungen mit dem Latex-Paket ifthen
- Code-Beispiel:

```
\setcounter{m}{0}
\begin{animateinline}[poster=first, controls]{40}
  \whiledo{ \them < 101 }{
    ...
    \stepcounter{m}
    \ifthenelse{ \them < 101 }
      % branching is used to show last slide after exit
      { \newframe }
    { \end{animateinline} \relax }
  }
```

- Beispiel: Abkühlprozess