

# Mathematik mit Mathematica

*Praktikum im Wintersemester 2021/22 an der TU Braunschweig  
betreut von Prof. Dr. Michael Herrmann*

---

## Tutorium 04: Graphiken und Animationen

---

Mathematica besitzt eine eigene und sehr intuitive Graphik-Sprache, wobei `Graphics` der Grundbefehl ist und eine Liste weiterer Befehle sequentiell abarbeitet.

---

### Graphik-Sprache

#### Graphikelemente 2D

`In[ ]:= (* Polygonzüge *)`

```
In[ ]:= p1 =  
Graphics[  
  {  
    RGBColor[1, 0.4, 0.3],  
    Thickness[0.015],  
    Line[{{0, 0}, {1, 0}, {1, 1}, {1/2, 1/2}}] (* Liste von 2D Punkten *)  
  }  
]
```

Out[ ]:=

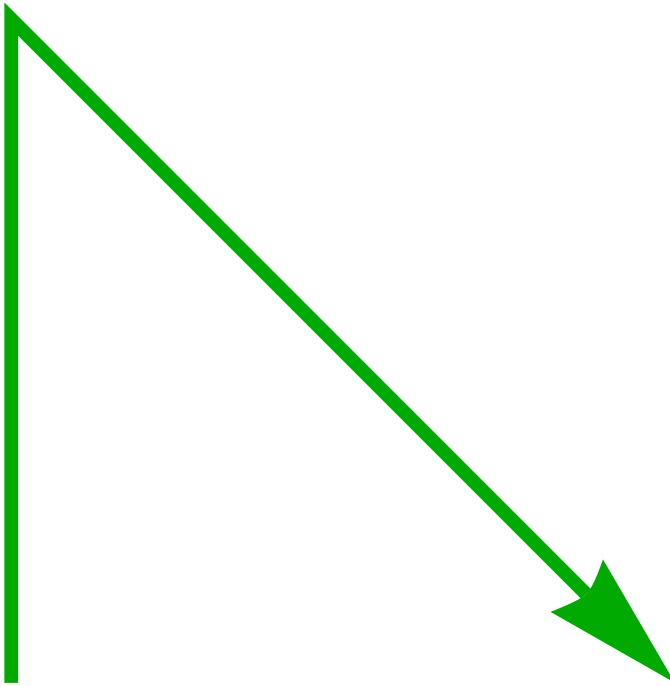


In[ ]:=

```
In[ ]:= (* Polygonzüge mit Pfeilen *)
```

```
In[*]:= p2 = Graphics[  
  {  
    Darker[Green],  
    Thickness[0.02],  
    Arrowheads[0.2],  
    Arrow[{{0, 0}, {0, 1}, {1, 0}}] (* Liste von 2D Punkten *)  
  }  
]
```

Out[\*]=

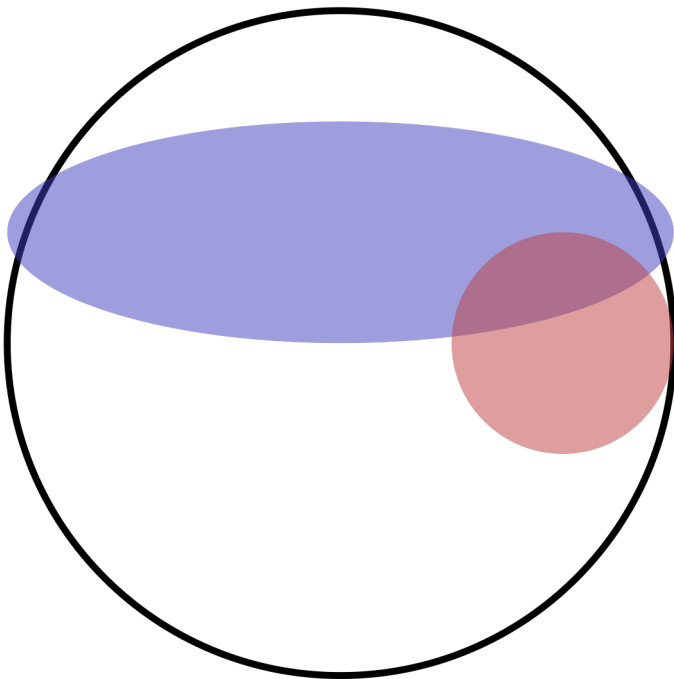


In[ ]:= (\* Kreise \*)

p3 =

```
Graphics[
  {
    Thickness[0.01],
    Circle[{0, 0}, 1.5],
    Opacity[0.5],
    Blend[{Gray, Blue}, 0.5],
    Disk[{0, .5}, {1.5, .5}],
    Blend[{Gray, Red}, 0.5],
    Disk[{1, 0}, 0.5]
  }
]
```

Out[ ]:=



In[ ]:= (\* Rechtecke \*)

p4 =

```
Graphics[
  {
    Orange,
    Darker[Orange],
    Opacity[0.5],
    Rectangle[{-1, -1}, {1, 1}]
  }
]
```

Out[ ]:=



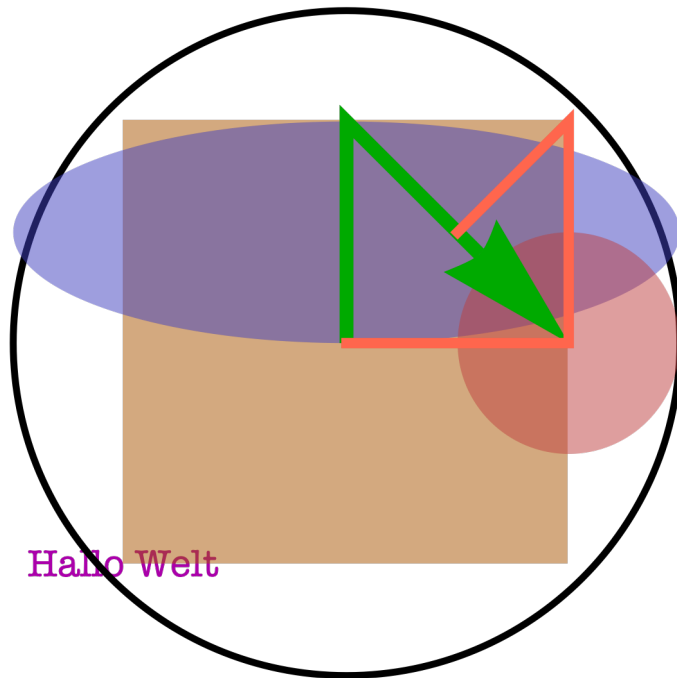
```
In[ ]:= (* etwas Text *)  
p5 =  
Graphics[  
  {  
    Text[  
      Style[  
        "Hallo Welt",  
        FontColor -> Darker[Magenta],  
        FontFamily -> "American Typewriter",  
        FontSize -> 20  
      ],  
      {-1, -1}  
    ]  
  }  
]
```

Out[ ]:=

Hallo Welt

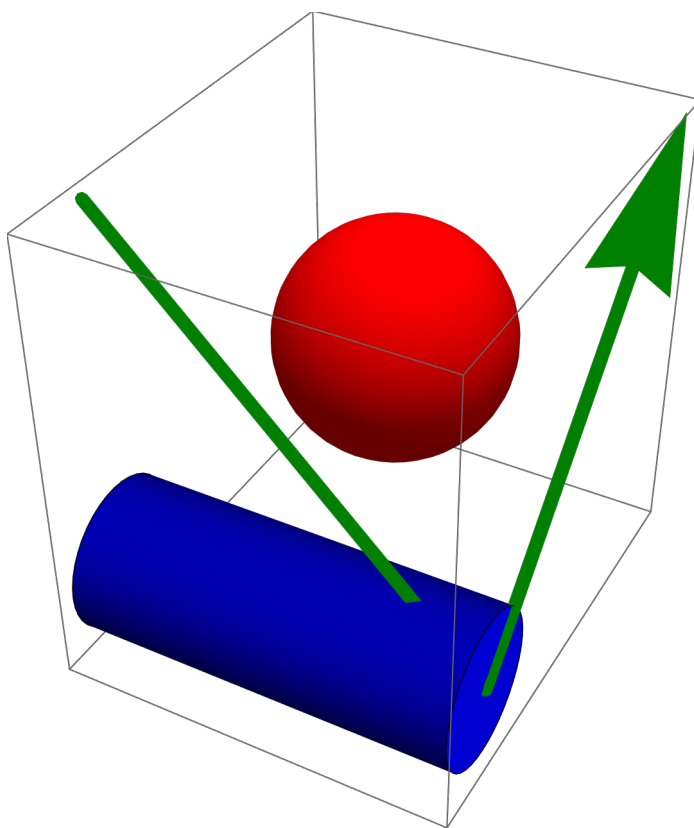
```
In[ ]:= (* alles zusammen *)  
pic = Show[{p5, p4, p3, p2, p1}]
```

Out[ ]:=



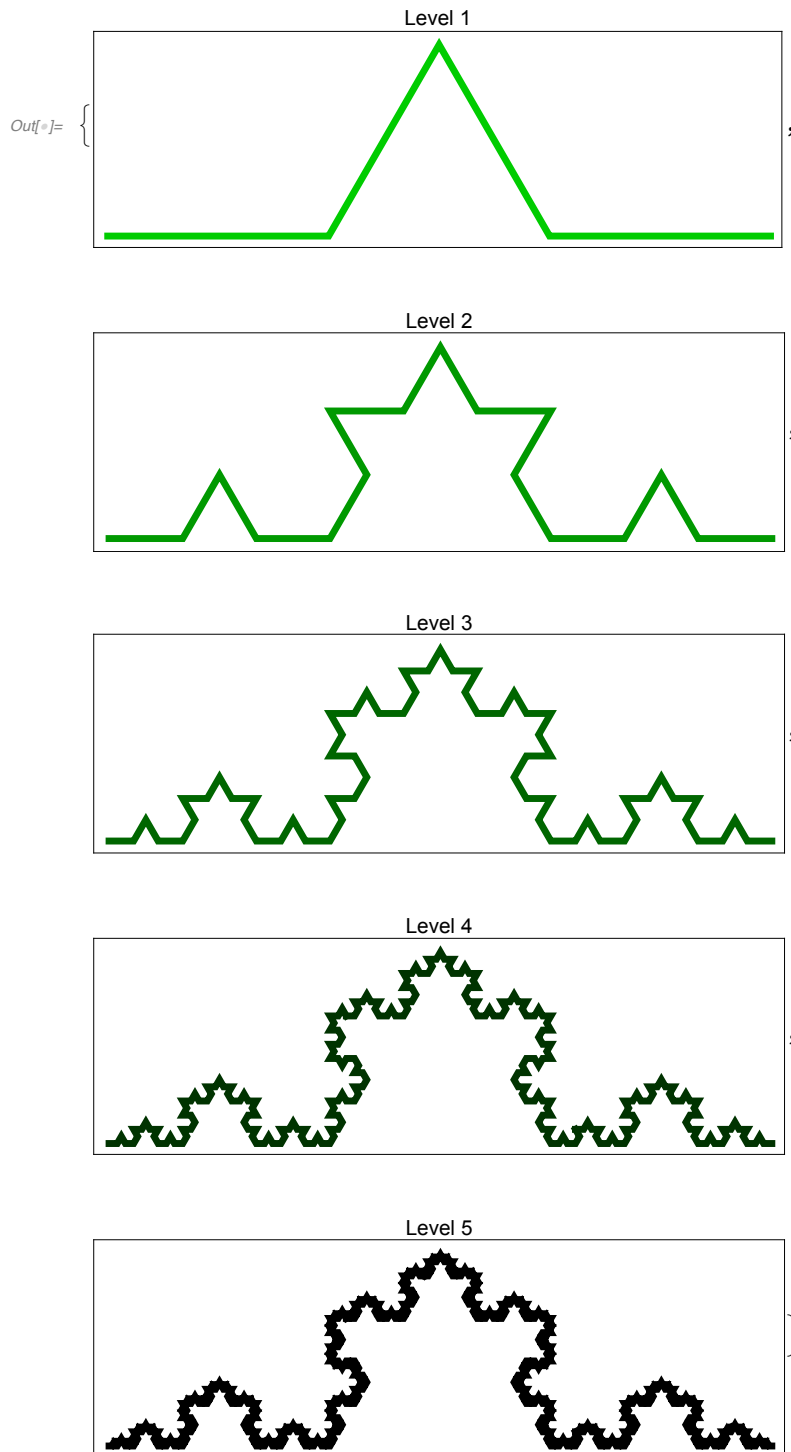
```
In[ ]:= (* geht auch in 3D *)  
Graphics3D[  
  {  
    RGBColor[0, 0.5, 0],  
    Thickness[0.02],  
    Arrowheads[0.2],  
    Arrow[{{0, 0, 1}, {1, 0, 0}, {1, 1, 1}}],  
    Red,  
    Sphere[{0.5, .5, .5}, .3],  
    Blue,  
    Cylinder[{{0, 0, 0}, {1, 0, 0}}, 0.2]  
  }  
]
```

Out[ ]:=





```
In[ ]:= (* gut für fraktale Gebiete, wobei vieles schon vorimplementiert ist *)
Table[
  Graphics[
    {
      Thickness[0.01],
      Blend[{Black, Green}, 1 - n / 5],
      KochCurve[n]
    },
    ImageSize → Medium,
    PlotLabel → "Level " <> ToString[n],
    Frame → True, FrameTicks → None
  ],
  {n, 1, 5}
]
```



*In[\*]:=* (\* Kochsche Kurve ist als Liste von Graphik-Kommandos vorhanden \*)

**KochCurve[2]**

*Out[\*]=* Line[{{0., 0.}, {0.111111, 0.}, {0.166667, 0.096225}, {0.222222, 0.},  
 {0.333333, 0.}, {0.388889, 0.096225}, {0.333333, 0.19245}, {0.444444, 0.19245},  
 {0.5, 0.288675}, {0.555556, 0.19245}, {0.666667, 0.19245}, {0.611111, 0.096225},  
 {0.666667, 0.}, {0.777778, 0.}, {0.833333, 0.096225}, {0.888889, 0.}, {1., 0.}}]

## Datenausgabe

```
In[ ]:= (* Ausgabe eines Bildes in verschiedene Formate *)
SetDirectory["~/Desktop"]
Export["pic.pdf", pic] (* Vektorgraphik *)
Export["pic.eps", pic] (* Vektorgraphik *)
Export["pic.png", pic]
Export["pic.bmp", pic]
Export["pic.jpeg", pic]
```

```
Out[ ]:= /Users/Herrmann/Desktop
```

```
Out[ ]:= pic.pdf
```

```
Out[ ]:= pic.eps
```

```
Out[ ]:= pic.png
```

```
Out[ ]:= pic.bmp
```

```
Out[ ]:= pic.jpeg
```

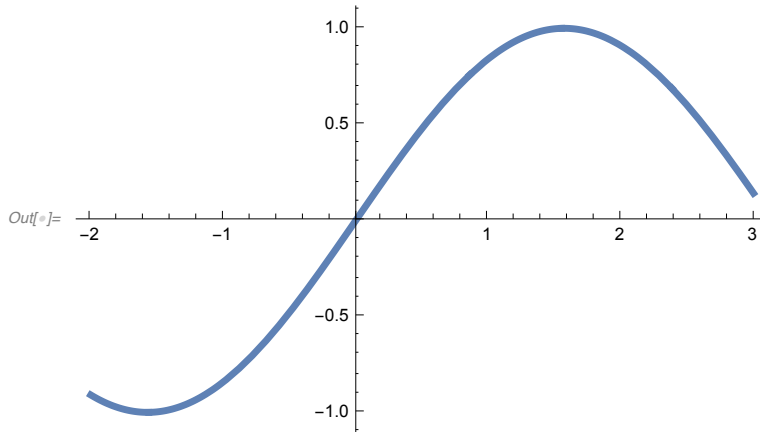
```
In[ ]:= (* Ausgabe von Rohdaten erfolgt auch mit Export *)
data =
  Table[
    RandomReal[],
    {i1, 1, 10},
    {i2, 1, 5}
  ]
Export["test.dat", data]
```

```
Out[ ]:= {{0.463469, 0.0990244, 0.75995, 0.695191, 0.408961},
{0.584461, 0.928443, 0.928134, 0.0597562, 0.767859},
{0.642311, 0.565796, 0.610376, 0.771447, 0.149527},
{0.939592, 0.17432, 0.582963, 0.749503, 0.0157166},
{0.715184, 0.286578, 0.975918, 0.49188, 0.365546},
{0.463831, 0.292947, 0.693286, 0.910821, 0.513437},
{0.510926, 0.192775, 0.596512, 0.962377, 0.45326},
{0.402209, 0.954882, 0.118135, 0.412038, 0.628001},
{0.219287, 0.82878, 0.204411, 0.726666, 0.889045},
{0.834616, 0.754987, 0.434456, 0.895131, 0.554884}}
```

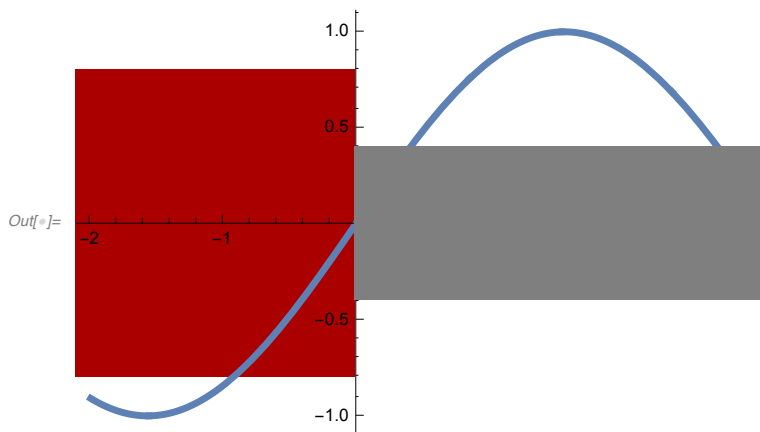
```
Out[ ]:= test.dat
```

## Graphiken und Plots

```
In[ ]:= Plot[
  Sin[x], {x, -2, 3},
  PlotStyle → Thickness[0.01]
]
```

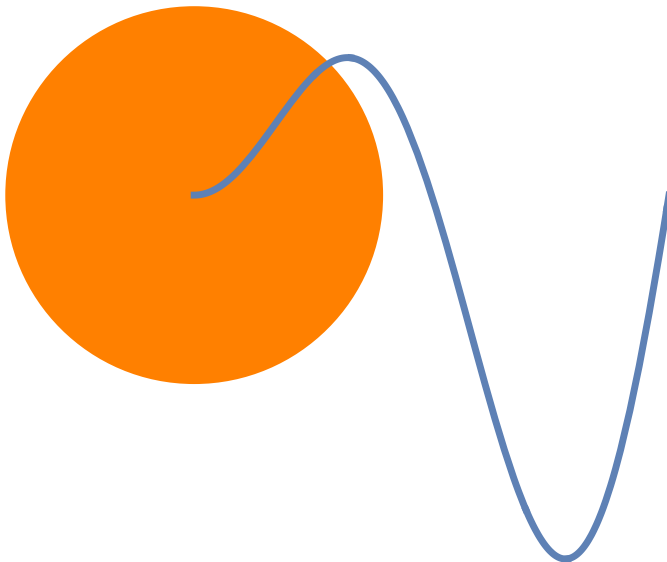


```
In[ ]:= (* Prolog und Epilog *)
p1 =
Plot[
  Sin[x], {x, -2, 3},
  PlotStyle → Thickness[0.01],
  Prolog → (* vor dem Plot *)
  {
    Darker[Red],
    Rectangle[{-Pi, -0.8}, {0, 0.8}]
  },
  Epilog → (* nach dem Plot *)
  {
    Gray,
    Rectangle[{0, -0.4}, {Pi, 0.4}]
  }
]
```



```
In[ ]:=  
(* oder mit Show *)  
Show[  
  {  
    Graphics[  
      {  
        Orange,  
        Disk[{0, 0}, 2.5]  
      }  
    ],  
    Plot[  
      x * Sin[x],  
      {x, 0, 2 Pi},  
      PlotStyle -> Thickness[0.01]  
    ]  
  }  
]
```

Out[ ]:=



---

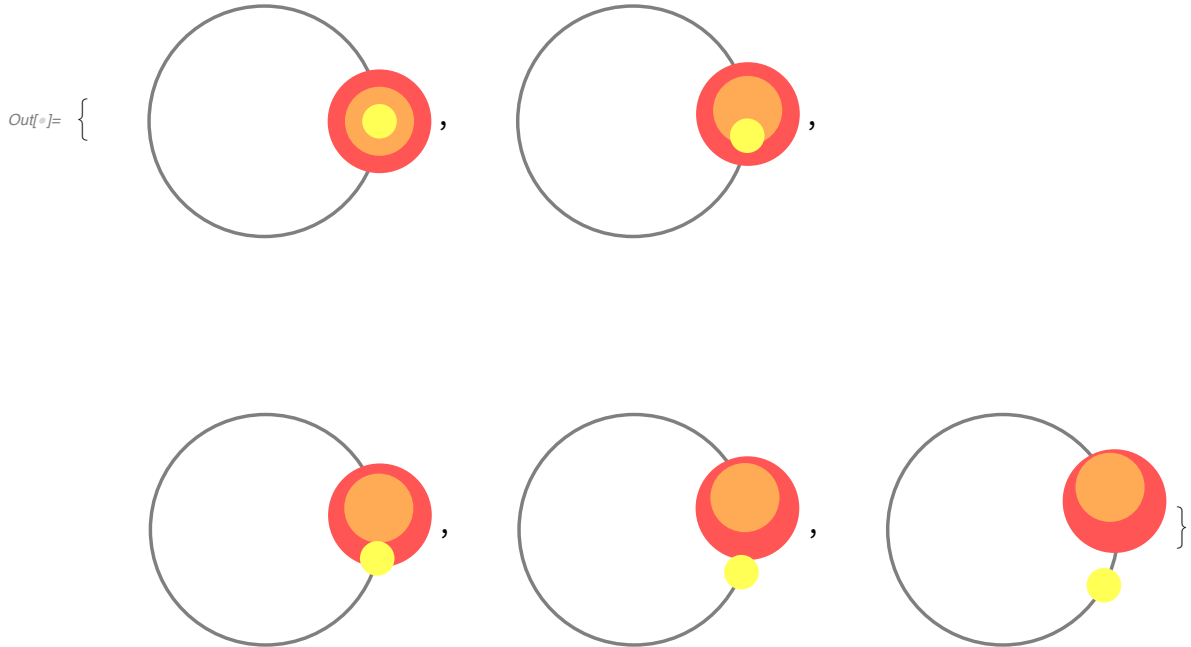
## Animationen

```
In[ ]:= (* Idee: Film = Liste von Bildern *)
```

```
In[ ]:= (* erzeuge Liste von Bilder *)
frames =
Table[
Graphics[
{
(* ein Kreis *)
Gray,
Thickness[0.01],
Circle[{0, 0}, 1],
(* Punkt 1 auf Kreis *)
PointSize[.3],
Lighter[Red],
Point[{Cos[2 * Pi * t], Sin[2 * Pi * t]}],
(* Punkt 2 auf Kreis *)
PointSize[.2],
Lighter[Orange],
Point[{Cos[3 * Pi * t], Sin[3 * Pi * t]}],
(* Punkt 3 auf Kreis *)
PointSize[.1],
Lighter[Yellow],
Point[{Cos[-4 * Pi * t], Sin[-4 * Pi * t]}]
},
PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}}
],
{t, 0, 1, 0.01} (* Laufvariable für 101 Bilder*)
];
```

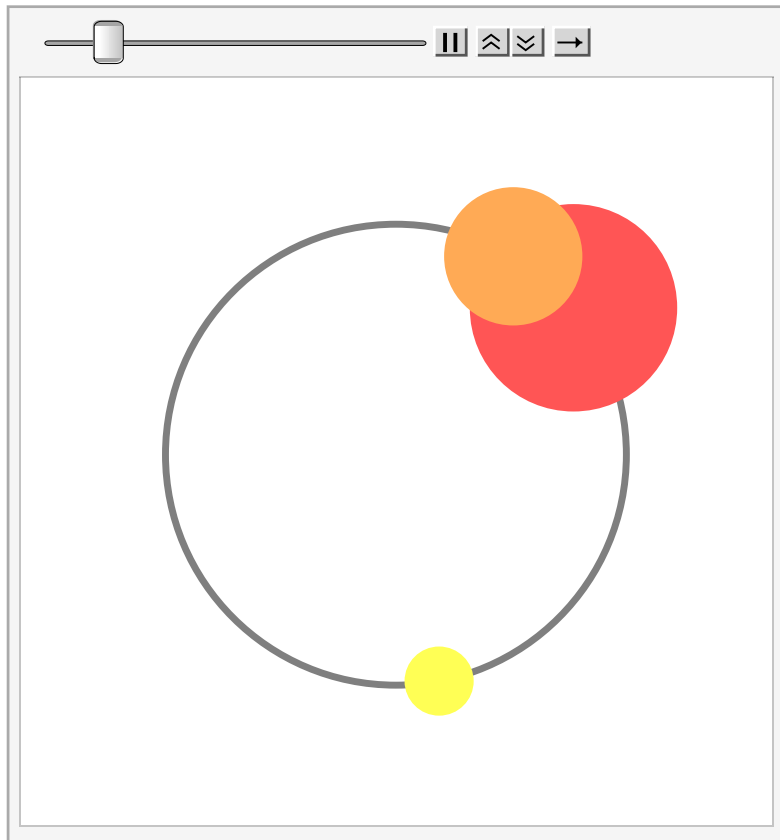
```
In[ ]:= (* einige der vielen Bilder *)  
Dimensions[frames]  
Take[frames, {1, 5}]
```

```
Out[ ]:= {101}
```



`In[*]:= (* eine Animation ist eine Liste von Graphiken *)`  
`ListAnimate[frames]`

`Out[*]=`



`In[*]:= (* Exportiere die Liste der Bilder (bei uns frames),`  
 jedoch nicht(!) das Ergebnis von ListAnimate \*)

`In[*]:= Export["spheres.avi", frames]`  
`Export["spheres.mov", frames]`

`Out[*]= spheres.avi`

`Out[*]= spheres.mov`

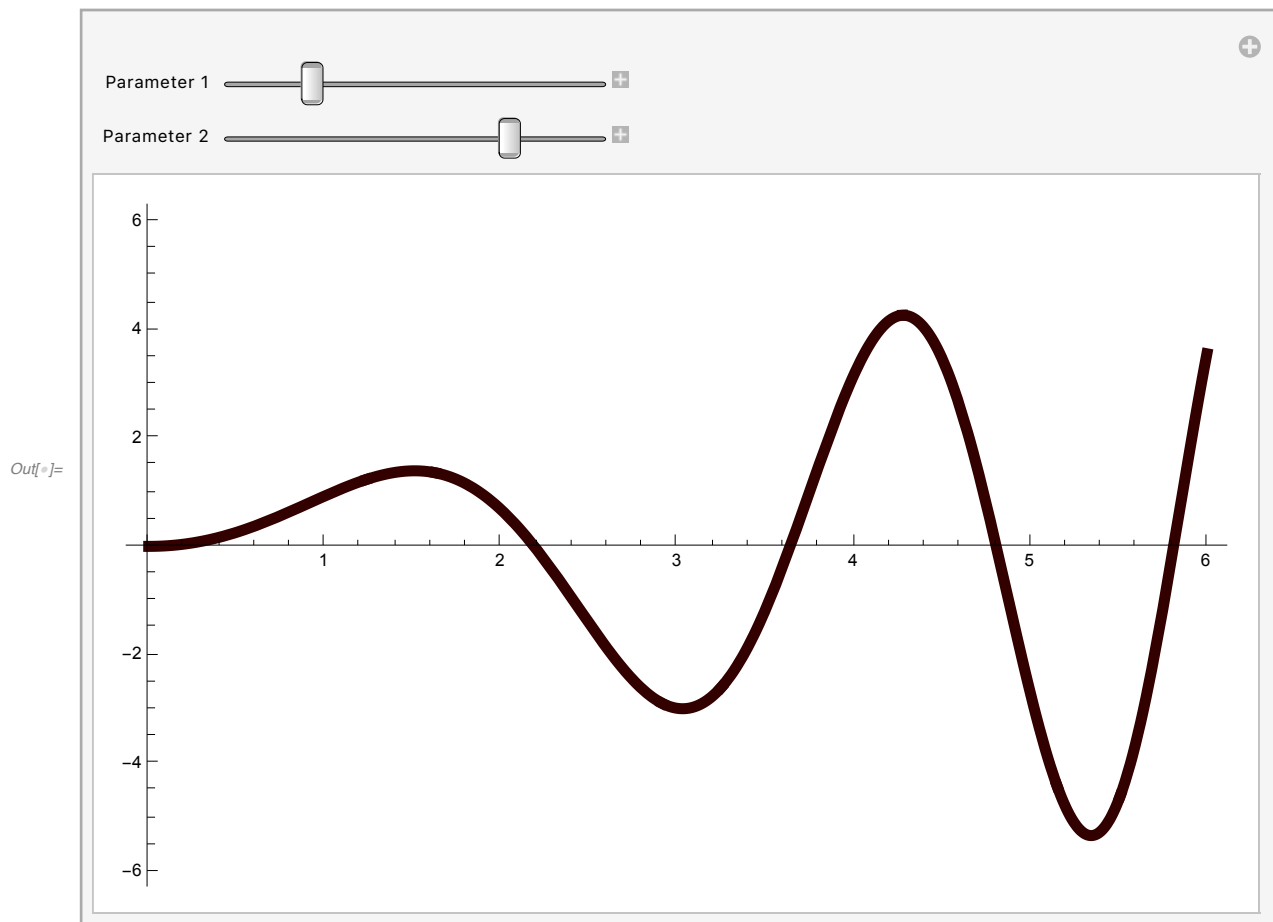


## Interaktive Elemente

```

In[ ]:= Manipulate[
  (* was soll manipuliert werden *)
  Plot[
    prm2 * x * Sin[x (1 + prm1 * x)], (* prm ist eine Parameter *)
    {x, 0, 6},
    PlotStyle -> {
      Blend[{Black, Red}, prm1 / 1],
      Thickness[0.01]
    },
    PlotRange -> {-2 Pi, 2 Pi},
    ImageSize -> Large
  ],
  (* Parameter, die manipuliert werden sollen *)
  {{prm1, .2, "Parameter 1"}, 0, 1},
  {{prm2, 1., "Parameter 2"}, .3, 1.2}
]

```



```
In[ ]:= (* es muss keine Graphik sein *)
```

```
Manipulate[
```

```
  Solve[x * x + x * prmP + prmQ == 1, x],
```

```
  {{prmP, 2, "p"}, 0, 10},
```

```
  {{prmQ, 1, "q"}, 0, 10},
```

```
  FrameLabel -> Style["pq-Formel",
```

```
    FontFamily -> "Alegreya SC", FontSize -> 14, FontColor -> Darker[Red]]
```

```
]
```

```
Out[ ]:=
```

The screenshot shows a Mathematica Manipulate interface. At the top, there are two sliders: one for parameter 'p' and one for parameter 'q'. Below the sliders is a text box containing the solutions  $\{x \rightarrow -2\}, \{x \rightarrow 0\}$ . At the bottom of the interface, the text "PQ-FORMEL" is displayed in red. The interface also includes a plus sign icon in the top right corner and a plus sign icon next to each slider.

```
In[ ]:=
```