

# Kurs ■ Cours

## 4. Graphiken ■ Graphiques

Die Gliederung dieses Kurses folgt in groben Zügen dem Buch von Nancy Blachman: A Practical Approach....

Hinweis: Kapitel 4 lesen!

Run mit WIN+*Mathematica* Version 5.2

■ L'articulation de ce cours correspond à peu près à celle du livre de Nancy Blachman: A Practical Approach....

Indication: Lire le chapitre 4.

Testé avec *Mathematica* version 5.2+WIN

WIR94/98/99/2000/2007 // Copyright Rolf Wirz

---

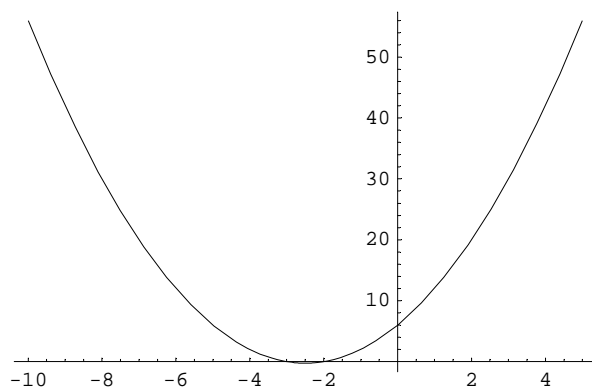
### 4.1. Zweidimensionale Plots

#### ■ Plots à deux dimensions

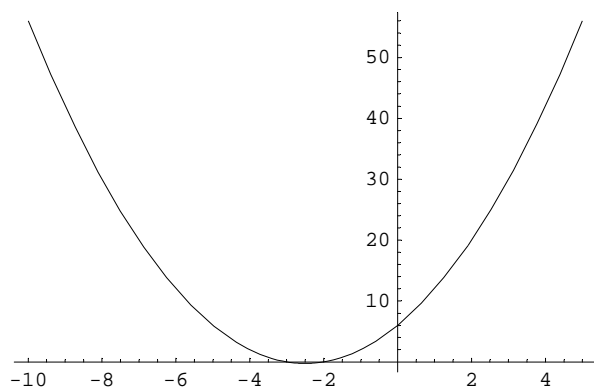
Gewöhnliche Funktion über einem Intervall. ";" unterdrückt den letzten Output:

■ Fonction commune sur un intervalle. ";" empêche le dernier output.

```
In[1]:= Plot[x^2 + 5x + 6, {x, -10, 5}];
```

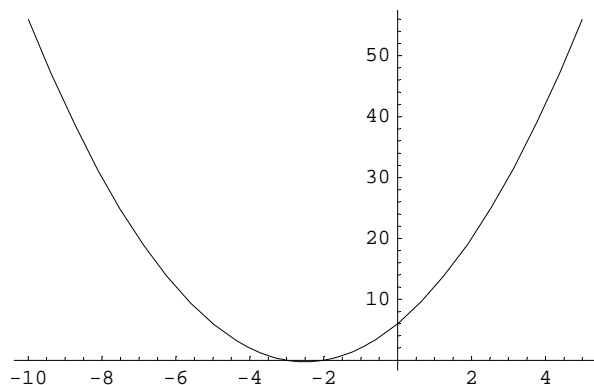


```
In[2]:= Plot[x^2 + 5x + 6, {x, -10, 5}]
```



```
Out[2]= - Graphics -
```

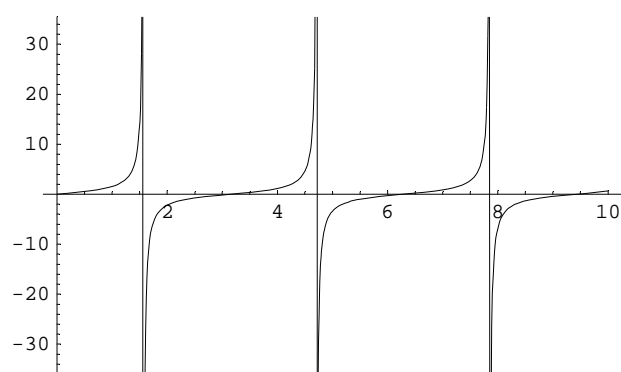
```
In[3]:= y = x^2 + 5x + 6;  
Plot[y, {x, -10, 5}];
```



Pole bieten keine Probleme:

■ Les pôles ne causent pas de problèmes:

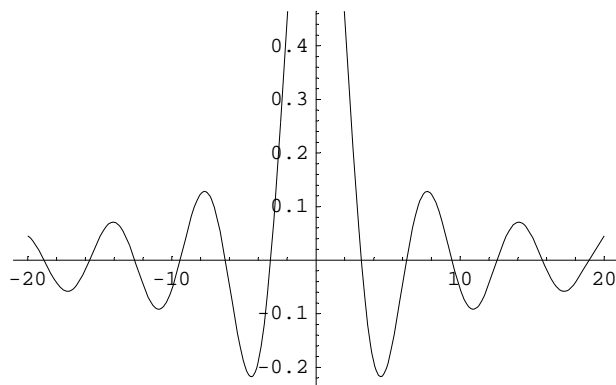
```
In[5]:= Plot[Tan[x], {x, 0, 10}];
```



Vielleicht wird etwas abgeschnitten:

■ *Mathematica* coupe peut-être quelque chose:

```
In[6]:= Plot[Sin[x]/x,{x, -20, 20}];
```



## 4.2. Options

### ■ Options

#### 4.2.1. Allgemeines

##### ■ En général

Ein Plot kann auf diverse Weise durch Angabe der Options verändert werden:

■ Un plot peut être transformé de différentes façons en indiquant les options:

```
In[7]:= ??Plot
```

```
Plot[f, {x, xmin, xmax}] generates a plot of f as a function of x from xmin to
xmax. Plot[{f1, f2, ... }, {x, xmin, xmax}] plots several functions fi. Mehr...
```

```
Attributes[Plot] = {HoldAll, Protected}
```

```
Options[Plot] = {AspectRatio ->  $\frac{1}{\text{GoldenRatio}}$ , Axes -> Automatic, AxesLabel -> None,
  AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic, ColorOutput -> Automatic,
  Compiled -> True, DefaultColor -> Automatic, DefaultFont -> $DefaultFont,
  DisplayFunction -> $DisplayFunction, Epilog -> {}, FormatType -> $FormatType, Frame -> False,
  FrameLabel -> None, FrameStyle -> Automatic, FrameTicks -> Automatic, GridLines -> None,
  ImageSize -> Automatic, MaxBend -> 10., PlotDivision -> 30., PlotLabel -> None,
  PlotPoints -> 25, PlotRange -> Automatic, PlotRegion -> Automatic, PlotStyle -> Automatic,
  Prolog -> {}, RotateLabel -> True, TextStyle -> $TextStyle, Ticks -> Automatic}
```

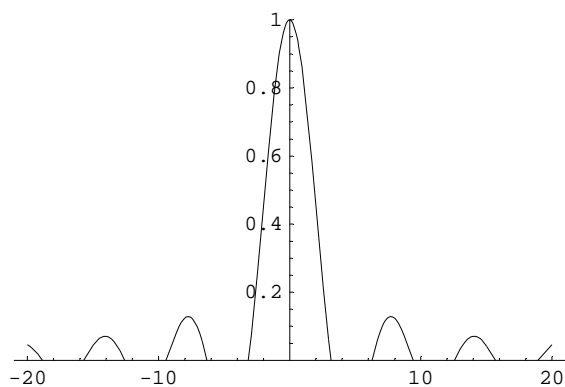
```
In[8]:= Options[Plot]
```

```
Out[8]= {AspectRatio ->  $\frac{1}{\text{GoldenRatio}}$ , Axes -> Automatic, AxesLabel -> None,
  AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic,
  ColorOutput -> Automatic, Compiled -> True, DefaultColor -> Automatic,
  DefaultFont -> $DefaultFont, DisplayFunction -> $DisplayFunction, Epilog -> {},
  FormatType -> $FormatType, Frame -> False, FrameLabel -> None, FrameStyle -> Automatic,
  FrameTicks -> Automatic, GridLines -> None, ImageSize -> Automatic,
  MaxBend -> 10., PlotDivision -> 30., PlotLabel -> None, PlotPoints -> 25,
  PlotRange -> Automatic, PlotRegion -> Automatic, PlotStyle -> Automatic,
  Prolog -> {}, RotateLabel -> True, TextStyle -> $TextStyle, Ticks -> Automatic}
```

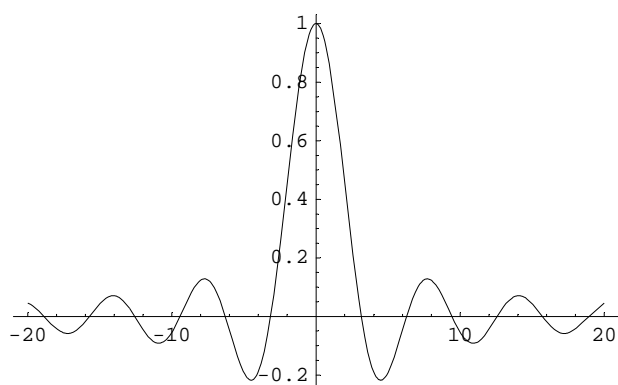
Beispiel mit dem y-Intervall:

■ Exemple avec l'intervalle y:

```
In[9]:= Plot[Sin[x]/x,{x, -20, 20}, PlotRange -> {0, 1}];
```



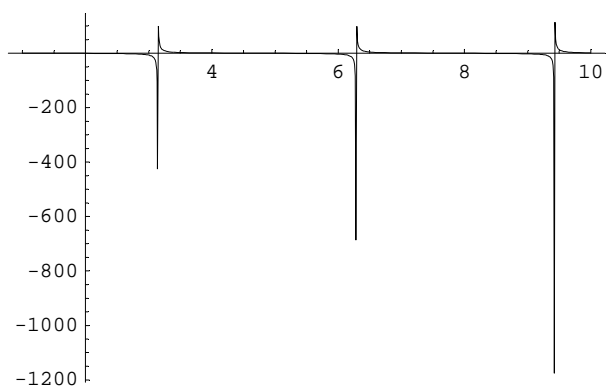
```
In[10]:= Plot[Sin[x]/x,{x, -20, 20}, PlotRange -> All];
```



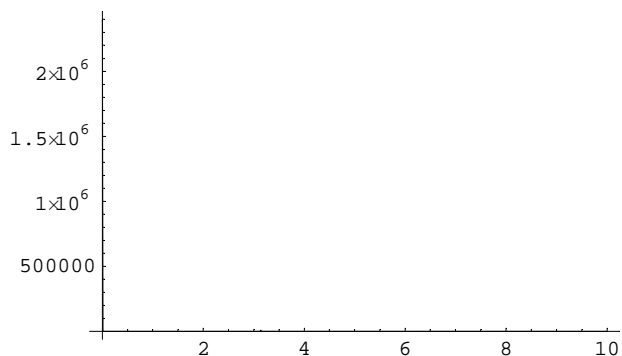
Vielleicht wird trotzdem "etwas" abgeschnitten...:

■ Peut-être *Mathematica* coupe quand-même "quelque chose"....:

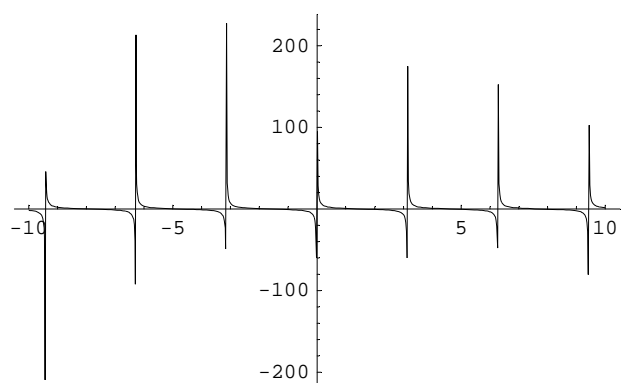
```
In[11]:= Plot[Cot[x],{x, 1, 10}, PlotRange -> All];
```



```
In[12]:= Plot[Cot[x],{x, 0, 10}, PlotRange -> All];
```



```
In[13]:= Plot[Cot[x],{x, -10, 10}, PlotRange -> All];
```



#### 4.2.2. Die Voreinstellung von Mathematica kann abgefragt werden:

##### ■ La mise au point préalable de *Mathematica* peut être interrogée:

```
In[14]:= Options[Plot]
```

```
Out[14]= {AspectRatio ->  $\frac{1}{\text{GoldenRatio}}$ , Axes -> Automatic, AxesLabel -> None,
  AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic,
  ColorOutput -> Automatic, Compiled -> True, DefaultColor -> Automatic,
  DefaultFont -> $DefaultFont, DisplayFunction -> $DisplayFunction, Epilog -> {},
  FormatType -> $FormatType, Frame -> False, FrameLabel -> None, FrameStyle -> Automatic,
  FrameTicks -> Automatic, GridLines -> None, ImageSize -> Automatic,
  MaxBend -> 10., PlotDivision -> 30., PlotLabel -> None, PlotPoints -> 25,
  PlotRange -> Automatic, PlotRegion -> Automatic, PlotStyle -> Automatic,
  Prolog -> {}, RotateLabel -> True, TextStyle -> $TextStyle, Ticks -> Automatic}
```

```
In[15]:= AspectRatio /. Options[Plot]
```

```
Out[15]=  $\frac{1}{\text{GoldenRatio}}$ 
```

```
In[16]:= ??AspectRatio
```

```
AspectRatio is an option for Graphics and related
  functions which specifies the ratio of height to width for a plot. Mehr...
```

```
Attributes[AspectRatio] = {Protected}
```

```
In[17]:= PlotPoints /. Options[Plot]
```

```
Out[17]= 25
```

Der Output kann unterdrückt werden:

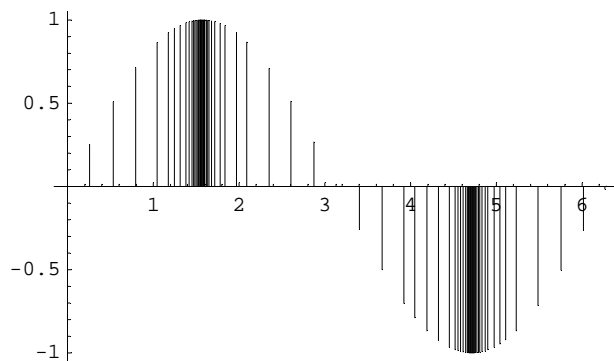
■ On peut réprimer l'output.

```
In[18]:= test = Plot[Sin[x],{x, 0, 2 Pi},
  DisplayFunction -> Identity];
```

Hier eine Zeichnung, die zeigt, in welchen Abständen *Mathematica* die Funktionswerte berechnet:

■ Voici un dessin qui montre à quelles distances *Mathematica* calcule les valeurs des fonctions:

```
In[19]:= Show[Graphics[{Thickness[0.001], Map[Line[{{#[[1]],
  0},#]}]&, Nest[First, test, 4]}],
  Axes->Automatic];
```



Was tun die einzelnen Befehle?

■ Que fait chaque ordre?

```
In[20]:= test
```

```
Out[20]= - Graphics -
```

```
In[21]:= ??Nest
```

Nest[f, expr, n] gives an expression with f applied n times to expr. Mehr...

```
Attributes[Nest] = {Protected}
```

```
In[22]:= Nest[First, test, 4]
```

```
Out[22]= {{2.61799×10-7, 2.61799×10-7}, {0.25489, 0.252139}, {0.532869, 0.508007},  
 {0.793939, 0.71312}, {1.04501, 0.864929}, {1.17413, 0.922355}, {1.24595, 0.947701},  
 {1.31226, 0.966765}, {1.37966, 0.981789}, {1.41517, 0.987915}, {1.45282, 0.993049},  
 {1.46855, 0.994777}, {1.48546, 0.996361}, {1.50009, 0.997501}, {1.51605, 0.998502},  
 {1.52572, 0.998984}, {1.53469, 0.999348}, {1.54327, 0.999621},  
 {1.54766, 0.999732}, {1.55242, 0.999831}, {1.5571, 0.999906}, {1.5614, 0.999956},  
 {1.56987, 1.}, {1.57756, 0.999977}, {1.58569, 0.999889}, {1.59373, 0.999737},  
 {1.59828, 0.999622}, {1.60249, 0.999498}, {1.6184, 0.998867}, {1.63286, 0.998075},  
 {1.64822, 0.997004}, {1.68008, 0.994035}, {1.71525, 0.989585},  
 {1.77763, 0.978685}, {1.8362, 0.964986}, {1.96935, 0.921622}, {2.09144, 0.867501},  
 {2.35285, 0.709467}, {2.60427, 0.511842}, {2.87186, 0.266474},  
 {3.12945, 0.0121375}, {3.40323, -0.258662}, {3.667, -0.501569},  
 {3.92078, -0.702701}, {4.04933, -0.788113}, {4.19073, -0.866996},  
 {4.32504, -0.925913}, {4.45069, -0.965952}, {4.51935, -0.981426},  
 {4.55317, -0.987352}, {4.58453, -0.991837}, {4.61287, -0.995052},  
 {4.64304, -0.997597}, {4.66029, -0.998643}, {4.66875, -0.999048},  
 {4.67652, -0.999357}, {4.68347, -0.999582}, {4.69109, -0.999773},  
 {4.69875, -0.999907}, {4.70683, -0.999985}, {4.71547, -0.999995},  
 {4.72345, -0.999939}, {4.72762, -0.999884}, {4.73221, -0.999803},  
 {4.74151, -0.999576}, {4.74967, -0.999305}, {4.75845, -0.998939},  
 {4.77433, -0.998082}, {4.78997, -0.996992}, {4.80641, -0.995583},  
 {4.83579, -0.992395}, {4.87, -0.987606}, {4.90178, -0.982119},  
 {4.97386, -0.96601}, {5.04347, -0.945692}, {5.10909, -0.92234},  
 {5.2318, -0.86811}, {5.48592, -0.715447}, {5.75622, -0.502911},  
 {6.01652, -0.263515}, {6.26682, -0.0163632}, {6.28319, -2.61799×10-7}}
```

```
In[23]:= Nest[First, test, 5]
```

```
Out[23]= {2.61799×10-7, 2.61799×10-7}
```

```
In[24]:= Nest[First, test, 3]
```

```
Out[24]= Line[{{2.61799×10-7, 2.61799×10-7}, {0.25489, 0.252139}, {0.532869, 0.508007},
  {0.793939, 0.71312}, {1.04501, 0.864929}, {1.17413, 0.922355}, {1.24595, 0.947701},
  {1.31226, 0.966765}, {1.37966, 0.981789}, {1.41517, 0.987915}, {1.45282, 0.993049},
  {1.46855, 0.994777}, {1.48546, 0.996361}, {1.50009, 0.997501}, {1.51605, 0.998502},
  {1.52572, 0.998984}, {1.53469, 0.999348}, {1.54327, 0.999621},
  {1.54766, 0.999732}, {1.55242, 0.999831}, {1.5571, 0.999906}, {1.5614, 0.999956},
  {1.56987, 1.}, {1.57756, 0.999977}, {1.58569, 0.999889}, {1.59373, 0.999737},
  {1.59828, 0.999622}, {1.60249, 0.999498}, {1.6184, 0.998867}, {1.63286, 0.998075},
  {1.64822, 0.997004}, {1.68008, 0.994035}, {1.71525, 0.989585},
  {1.77763, 0.978685}, {1.8362, 0.964986}, {1.96935, 0.921622}, {2.09144, 0.867501},
  {2.35285, 0.709467}, {2.60427, 0.511842}, {2.87186, 0.266474},
  {3.12945, 0.0121375}, {3.40323, -0.258662}, {3.667, -0.501569},
  {3.92078, -0.702701}, {4.04933, -0.788113}, {4.19073, -0.866996},
  {4.32504, -0.925913}, {4.45069, -0.965952}, {4.51935, -0.981426},
  {4.55317, -0.987352}, {4.58453, -0.991837}, {4.61287, -0.995052},
  {4.64304, -0.997597}, {4.66029, -0.998643}, {4.66875, -0.999048},
  {4.67652, -0.999357}, {4.68347, -0.999582}, {4.69109, -0.999773},
  {4.69875, -0.999907}, {4.70683, -0.999985}, {4.71547, -0.999995},
  {4.72345, -0.999939}, {4.72762, -0.999884}, {4.73221, -0.999803},
  {4.74151, -0.999576}, {4.74967, -0.999305}, {4.75845, -0.998939},
  {4.77433, -0.998082}, {4.78997, -0.996992}, {4.80641, -0.995583},
  {4.83579, -0.992395}, {4.87, -0.987606}, {4.90178, -0.982119},
  {4.97386, -0.96601}, {5.04347, -0.945692}, {5.10909, -0.92234},
  {5.2318, -0.86811}, {5.48592, -0.715447}, {5.75622, -0.502911},
  {6.01652, -0.263515}, {6.26682, -0.0163632}, {6.28319, -2.61799×10-7}}]
```

```
In[25]:= Nest[First, test, 2]
```

```
Out[25]= {Line[{{2.61799×10-7, 2.61799×10-7}, {0.25489, 0.252139}, {0.532869, 0.508007},
  {0.793939, 0.71312}, {1.04501, 0.864929}, {1.17413, 0.922355},
  {1.24595, 0.947701}, {1.31226, 0.966765}, {1.37966, 0.981789},
  {1.41517, 0.987915}, {1.45282, 0.993049}, {1.46855, 0.994777},
  {1.48546, 0.996361}, {1.50009, 0.997501}, {1.51605, 0.998502},
  {1.52572, 0.998984}, {1.53469, 0.999348}, {1.54327, 0.999621},
  {1.54766, 0.999732}, {1.55242, 0.999831}, {1.5571, 0.999906}, {1.5614, 0.999956},
  {1.56987, 1.}, {1.57756, 0.999977}, {1.58569, 0.999889}, {1.59373, 0.999737},
  {1.59828, 0.999622}, {1.60249, 0.999498}, {1.6184, 0.998867}, {1.63286, 0.998075},
  {1.64822, 0.997004}, {1.68008, 0.994035}, {1.71525, 0.989585},
  {1.77763, 0.978685}, {1.8362, 0.964986}, {1.96935, 0.921622}, {2.09144, 0.867501},
  {2.35285, 0.709467}, {2.60427, 0.511842}, {2.87186, 0.266474},
  {3.12945, 0.0121375}, {3.40323, -0.258662}, {3.667, -0.501569},
  {3.92078, -0.702701}, {4.04933, -0.788113}, {4.19073, -0.866996},
  {4.32504, -0.925913}, {4.45069, -0.965952}, {4.51935, -0.981426},
  {4.55317, -0.987352}, {4.58453, -0.991837}, {4.61287, -0.995052},
  {4.64304, -0.997597}, {4.66029, -0.998643}, {4.66875, -0.999048},
  {4.67652, -0.999357}, {4.68347, -0.999582}, {4.69109, -0.999773},
  {4.69875, -0.999907}, {4.70683, -0.999985}, {4.71547, -0.999995},
  {4.72345, -0.999939}, {4.72762, -0.999884}, {4.73221, -0.999803},
  {4.74151, -0.999576}, {4.74967, -0.999305}, {4.75845, -0.998939},
  {4.77433, -0.998082}, {4.78997, -0.996992}, {4.80641, -0.995583},
  {4.83579, -0.992395}, {4.87, -0.987606}, {4.90178, -0.982119},
  {4.97386, -0.96601}, {5.04347, -0.945692}, {5.10909, -0.92234},
  {5.2318, -0.86811}, {5.48592, -0.715447}, {5.75622, -0.502911},
  {6.01652, -0.263515}, {6.26682, -0.0163632}, {6.28319, -2.61799×10-7}}]}
```

```
In[26]:= Nest[First, test, 1]
```

```
Out[26]= {{Line[{{2.61799×10-7, 2.61799×10-7}, {0.25489, 0.252139}, {0.532869, 0.508007},
{0.793939, 0.71312}, {1.04501, 0.864929}, {1.17413, 0.922355},
{1.24595, 0.947701}, {1.31226, 0.966765}, {1.37966, 0.981789},
{1.41517, 0.987915}, {1.45282, 0.993049}, {1.46855, 0.994777},
{1.48546, 0.996361}, {1.50009, 0.997501}, {1.51605, 0.998502},
{1.52572, 0.998984}, {1.53469, 0.999348}, {1.54327, 0.999621},
{1.54766, 0.999732}, {1.55242, 0.999831}, {1.5571, 0.999906},
{1.5614, 0.999956}, {1.56987, 1.}, {1.57756, 0.999977}, {1.58569, 0.999889},
{1.59373, 0.999737}, {1.59828, 0.999622}, {1.60249, 0.999498},
{1.6184, 0.998867}, {1.63286, 0.998075}, {1.64822, 0.997004},
{1.68008, 0.994035}, {1.71525, 0.989585}, {1.77763, 0.978685},
{1.8362, 0.964986}, {1.96935, 0.921622}, {2.09144, 0.867501},
{2.35285, 0.709467}, {2.60427, 0.511842}, {2.87186, 0.266474},
{3.12945, 0.0121375}, {3.40323, -0.258662}, {3.667, -0.501569},
{3.92078, -0.702701}, {4.04933, -0.788113}, {4.19073, -0.866996},
{4.32504, -0.925913}, {4.45069, -0.965952}, {4.51935, -0.981426},
{4.55317, -0.987352}, {4.58453, -0.991837}, {4.61287, -0.995052},
{4.64304, -0.997597}, {4.66029, -0.998643}, {4.66875, -0.999048},
{4.67652, -0.999357}, {4.68347, -0.999582}, {4.69109, -0.999773},
{4.69875, -0.999907}, {4.70683, -0.999985}, {4.71547, -0.999995},
{4.72345, -0.999939}, {4.72762, -0.999884}, {4.73221, -0.999803},
{4.74151, -0.999576}, {4.74967, -0.999305}, {4.75845, -0.998939},
{4.77433, -0.998082}, {4.78997, -0.996992}, {4.80641, -0.995583},
{4.83579, -0.992395}, {4.87, -0.987606}, {4.90178, -0.982119},
{4.97386, -0.96601}, {5.04347, -0.945692}, {5.10909, -0.92234},
{5.2318, -0.86811}, {5.48592, -0.715447}, {5.75622, -0.502911},
{6.01652, -0.263515}, {6.26682, -0.0163632}, {6.28319, -2.61799×10-7}}]}
```

```
In[27]:= First[test]
```

```
Out[27]= {{Line[{{2.61799×10-7, 2.61799×10-7}, {0.25489, 0.252139}, {0.532869, 0.508007},
  {0.793939, 0.71312}, {1.04501, 0.864929}, {1.17413, 0.922355},
  {1.24595, 0.947701}, {1.31226, 0.966765}, {1.37966, 0.981789},
  {1.41517, 0.987915}, {1.45282, 0.993049}, {1.46855, 0.994777},
  {1.48546, 0.996361}, {1.50009, 0.997501}, {1.51605, 0.998502},
  {1.52572, 0.998984}, {1.53469, 0.999348}, {1.54327, 0.999621},
  {1.54766, 0.999732}, {1.55242, 0.999831}, {1.5571, 0.999906},
  {1.5614, 0.999956}, {1.56987, 1.}, {1.57756, 0.999977}, {1.58569, 0.999889},
  {1.59373, 0.999737}, {1.59828, 0.999622}, {1.60249, 0.999498},
  {1.6184, 0.998867}, {1.63286, 0.998075}, {1.64822, 0.997004},
  {1.68008, 0.994035}, {1.71525, 0.989585}, {1.77763, 0.978685},
  {1.8362, 0.964986}, {1.96935, 0.921622}, {2.09144, 0.867501},
  {2.35285, 0.709467}, {2.60427, 0.511842}, {2.87186, 0.266474},
  {3.12945, 0.0121375}, {3.40323, -0.258662}, {3.667, -0.501569},
  {3.92078, -0.702701}, {4.04933, -0.788113}, {4.19073, -0.866996},
  {4.32504, -0.925913}, {4.45069, -0.965952}, {4.51935, -0.981426},
  {4.55317, -0.987352}, {4.58453, -0.991837}, {4.61287, -0.995052},
  {4.64304, -0.997597}, {4.66029, -0.998643}, {4.66875, -0.999048},
  {4.67652, -0.999357}, {4.68347, -0.999582}, {4.69109, -0.999773},
  {4.69875, -0.999907}, {4.70683, -0.999985}, {4.71547, -0.999995},
  {4.72345, -0.999939}, {4.72762, -0.999884}, {4.73221, -0.999803},
  {4.74151, -0.999576}, {4.74967, -0.999305}, {4.75845, -0.998939},
  {4.77433, -0.998082}, {4.78997, -0.996992}, {4.80641, -0.995583},
  {4.83579, -0.992395}, {4.87, -0.987606}, {4.90178, -0.982119},
  {4.97386, -0.96601}, {5.04347, -0.945692}, {5.10909, -0.92234},
  {5.2318, -0.86811}, {5.48592, -0.715447}, {5.75622, -0.502911},
  {6.01652, -0.263515}, {6.26682, -0.0163632}, {6.28319, -2.61799×10-7}}]}
```

```
In[28]:= Map[Line[{{#[[1]],0},#}]&, Nest[First, test, 4]]
```

```
Out[28]= {Line[{{2.61799×10-7, 0}, {2.61799×10-7, 2.61799×10-7}}],
  Line[{{0.25489, 0}, {0.25489, 0.252139}}],
  Line[{{0.532869, 0}, {0.532869, 0.508007}}],
  Line[{{0.793939, 0}, {0.793939, 0.71312}}],
  Line[{{1.04501, 0}, {1.04501, 0.864929}}],
  Line[{{1.17413, 0}, {1.17413, 0.922355}}],
  Line[{{1.24595, 0}, {1.24595, 0.947701}}],
  Line[{{1.31226, 0}, {1.31226, 0.966765}}],
  Line[{{1.37966, 0}, {1.37966, 0.981789}}],
  Line[{{1.41517, 0}, {1.41517, 0.987915}}],
  Line[{{1.45282, 0}, {1.45282, 0.993049}}],
  Line[{{1.46855, 0}, {1.46855, 0.994777}}],
  Line[{{1.48546, 0}, {1.48546, 0.996361}}],
  Line[{{1.50009, 0}, {1.50009, 0.997501}}],
  Line[{{1.51605, 0}, {1.51605, 0.998502}}],
  Line[{{1.52572, 0}, {1.52572, 0.998984}}],
  Line[{{1.53469, 0}, {1.53469, 0.999348}}],
  Line[{{1.54327, 0}, {1.54327, 0.999621}}],
  Line[{{1.54766, 0}, {1.54766, 0.999732}}],
  Line[{{1.55242, 0}, {1.55242, 0.999831}}],
  Line[{{1.5571, 0}, {1.5571, 0.999906}}], Line[{{1.5614, 0}, {1.5614, 0.999956}}],
  Line[{{1.56987, 0}, {1.56987, 1.}}], Line[{{1.57756, 0}, {1.57756, 0.999977}}],
  Line[{{1.58569, 0}, {1.58569, 0.999889}}],
  Line[{{1.59373, 0}, {1.59373, 0.999737}}],
  Line[{{1.59828, 0}, {1.59828, 0.999622}}],
  Line[{{1.60249, 0}, {1.60249, 0.999498}}],
```

```

Line[{{1.6184, 0}, {1.6184, 0.998867}}], Line[{{1.63286, 0}, {1.63286, 0.998075}}],
Line[{{1.64822, 0}, {1.64822, 0.997004}}],
Line[{{1.68008, 0}, {1.68008, 0.994035}}],
Line[{{1.71525, 0}, {1.71525, 0.989585}}],
Line[{{1.77763, 0}, {1.77763, 0.978685}}],
Line[{{1.8362, 0}, {1.8362, 0.964986}}], Line[{{1.96935, 0}, {1.96935, 0.921622}}],
Line[{{2.09144, 0}, {2.09144, 0.867501}}],
Line[{{2.35285, 0}, {2.35285, 0.709467}}],
Line[{{2.60427, 0}, {2.60427, 0.511842}}],
Line[{{2.87186, 0}, {2.87186, 0.266474}}],
Line[{{3.12945, 0}, {3.12945, 0.0121375}}],
Line[{{3.40323, 0}, {3.40323, -0.258662}}],
Line[{{3.667, 0}, {3.667, -0.501569}}], Line[{{3.92078, 0}, {3.92078, -0.702701}}],
Line[{{4.04933, 0}, {4.04933, -0.788113}}],
Line[{{4.19073, 0}, {4.19073, -0.866996}}],
Line[{{4.32504, 0}, {4.32504, -0.925913}}],
Line[{{4.45069, 0}, {4.45069, -0.965952}}],
Line[{{4.51935, 0}, {4.51935, -0.981426}}],
Line[{{4.55317, 0}, {4.55317, -0.987352}}],
Line[{{4.58453, 0}, {4.58453, -0.991837}}],
Line[{{4.61287, 0}, {4.61287, -0.995052}}],
Line[{{4.64304, 0}, {4.64304, -0.997597}}],
Line[{{4.66029, 0}, {4.66029, -0.998643}}],
Line[{{4.66875, 0}, {4.66875, -0.999048}}],
Line[{{4.67652, 0}, {4.67652, -0.999357}}],
Line[{{4.68347, 0}, {4.68347, -0.999582}}],
Line[{{4.69109, 0}, {4.69109, -0.999773}}],
Line[{{4.69875, 0}, {4.69875, -0.999907}}],
Line[{{4.70683, 0}, {4.70683, -0.999985}}],
Line[{{4.71547, 0}, {4.71547, -0.999995}}],
Line[{{4.72345, 0}, {4.72345, -0.999939}}],
Line[{{4.72762, 0}, {4.72762, -0.999884}}],
Line[{{4.73221, 0}, {4.73221, -0.999803}}],
Line[{{4.74151, 0}, {4.74151, -0.999576}}],
Line[{{4.74967, 0}, {4.74967, -0.999305}}],
Line[{{4.75845, 0}, {4.75845, -0.998939}}],
Line[{{4.77433, 0}, {4.77433, -0.998082}}],
Line[{{4.78997, 0}, {4.78997, -0.996992}}],
Line[{{4.80641, 0}, {4.80641, -0.995583}}],
Line[{{4.83579, 0}, {4.83579, -0.992395}}],
Line[{{4.87, 0}, {4.87, -0.987606}}], Line[{{4.90178, 0}, {4.90178, -0.982119}}],
Line[{{4.97386, 0}, {4.97386, -0.96601}}],
Line[{{5.04347, 0}, {5.04347, -0.945692}}],
Line[{{5.10909, 0}, {5.10909, -0.92234}}], Line[{{5.2318, 0}, {5.2318, -0.86811}}],
Line[{{5.48592, 0}, {5.48592, -0.715447}}],
Line[{{5.75622, 0}, {5.75622, -0.502911}}],
Line[{{6.01652, 0}, {6.01652, -0.263515}}],
Line[{{6.26682, 0}, {6.26682, -0.0163632}}],
Line[{{6.28319, 0}, {6.28319, -2.61799 × 10-7}}]

```

```

In[29]:= Graphics[{Thickness[0.001],
Map[Line[{{#[[1]],0},#]}&,
Nest[First, test, 4]]]

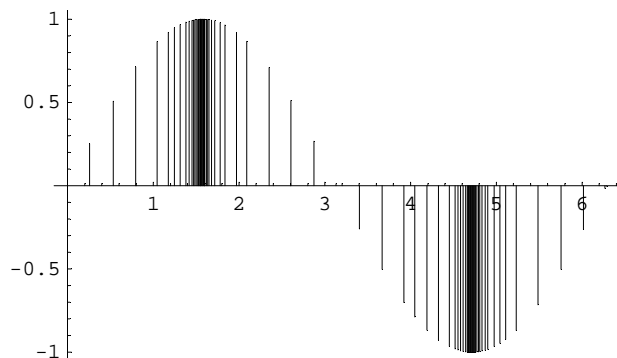
```

```

Out[29]= - Graphics -

```

```
In[30]:= Show[Graphics[{Thickness[0.001],
  Map[Line[{{#[[1]],0},#]}&,
  Nest[First, test, 4]]], Axes->Automatic];
```

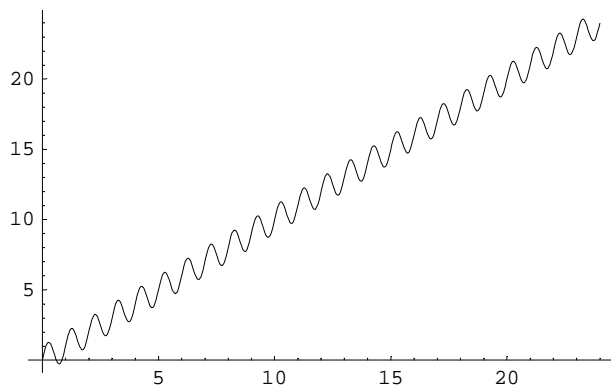


### 4.2.3. Probleme mit den Plot-Punkten: ■ Problèmes avec les points-"plot":

Kann das stimmen?

■ Est-ce correct?

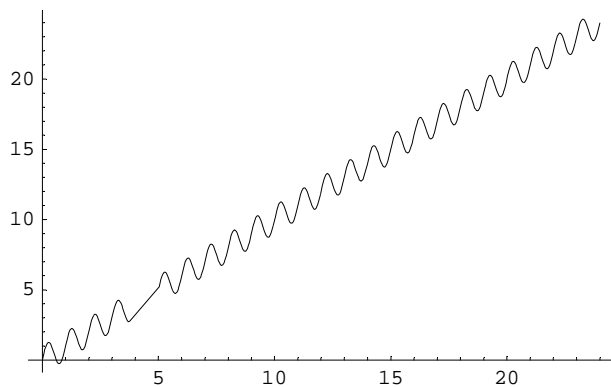
```
In[31]:= Plot[x + Sin[2 Pi x], {x, 0, 24}];
```



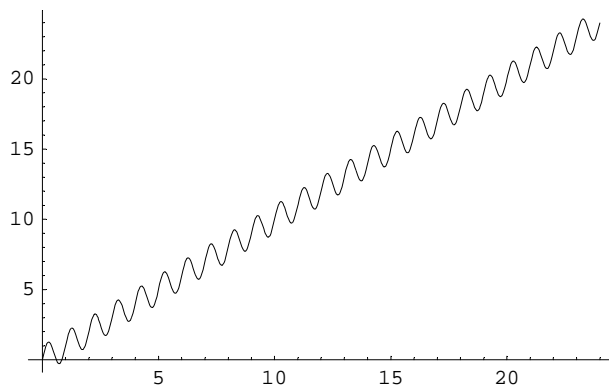
Andere Anzahl Punkte:

■ Autre nombre ce points:

```
In[32]:= Plot[x + Sin[2 Pi x], {x, 0, 24}, PlotPoints -> 20];
```



```
In[33]:= Plot[x + Sin[2 Pi x], {x, 0, 24}, PlotPoints -> 50];
```



Welche Punkte rechnet *Mathematica*?

■ Quels points sont calculés par *Mathematica*?

```
In[34]:= Table[{x, x + Sin[2 Pi x]}, {x, 0, 24}]
```

```
Out[34]= {{0, 0}, {1, 1}, {2, 2}, {3, 3}, {4, 4}, {5, 5}, {6, 6}, {7, 7}, {8, 8},
          {9, 9}, {10, 10}, {11, 11}, {12, 12}, {13, 13}, {14, 14}, {15, 15}, {16, 16},
          {17, 17}, {18, 18}, {19, 19}, {20, 20}, {21, 21}, {22, 22}, {23, 23}, {24, 24}}
```

#### 4.2.4. Wechsel des Plot-Stils:

##### ■ Changement du style "plot":

Liniendicke, Graustufe, Farbe, Linienart, etc. können angepasst werden:

■ Epaisseur de la ligne, nuance de gris, couleur, type de ligne etc. peuvent être adaptés:

```
In[35]:= Options[Plot]
```

```
Out[35]= {AspectRatio ->  $\frac{1}{\text{GoldenRatio}}$ , Axes -> Automatic, AxesLabel -> None,
          AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic,
          ColorOutput -> Automatic, Compiled -> True, DefaultColor -> Automatic,
          DefaultFont -> $DefaultFont, DisplayFunction -> $DisplayFunction, Epilog -> {},
          FormatType -> $FormatType, Frame -> False, FrameLabel -> None, FrameStyle -> Automatic,
          FrameTicks -> Automatic, GridLines -> None, ImageSize -> Automatic,
          MaxBend -> 10., PlotDivision -> 30., PlotLabel -> None, PlotPoints -> 25,
          PlotRange -> Automatic, PlotRegion -> Automatic, PlotStyle -> Automatic,
          Prolog -> {}, RotateLabel -> True, TextStyle -> $TextStyle, Ticks -> Automatic}
```

```
In[36]:= ??PlotStyle
```

```
PlotStyle is an option for Plot and ListPlot
that specifies the style of lines or points to be plotted. Mehr...
```

```
Attributes[PlotStyle] = {Protected}
```

```
In[37]:= ??Graphics
```

Graphics[primitives, options] represents a two-dimensional graphical image. Mehr...

```
Attributes[Graphics] = {Protected, ReadProtected}
```

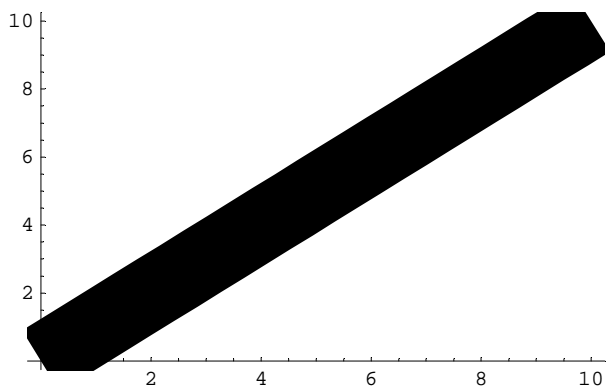
```
Options[Graphics] = {AspectRatio ->  $\frac{1}{\text{GoldenRatio}}$ , Axes -> False, AxesLabel -> None,
  AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic,
  ColorOutput -> Automatic, DefaultColor -> Automatic, DefaultFont -> $DefaultFont,
  DisplayFunction -> $DisplayFunction, Epilog -> {}, FormatType -> $FormatType, Frame -> False,
  FrameLabel -> None, FrameStyle -> Automatic, FrameTicks -> Automatic, GridLines -> None,
  ImageSize -> Automatic, PlotLabel -> None, PlotRange -> Automatic, PlotRegion -> Automatic,
  Prolog -> {}, RotateLabel -> True, TextStyle -> $TextStyle, Ticks -> Automatic}
```

```
In[38]:= PlotStyle /. Options[Plot]
```

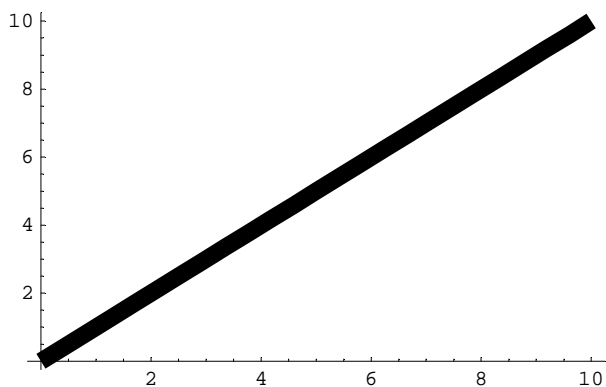
```
Out[38]= Automatic
```

Probiere aus: ■ Essaie:

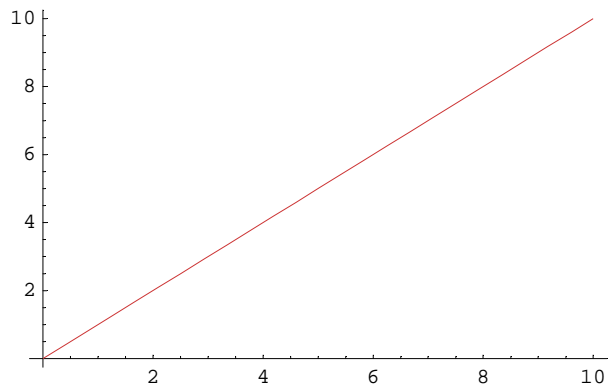
```
In[39]:= Plot[x, {x, 0, 10}, PlotStyle -> Thickness[0.125]];
```



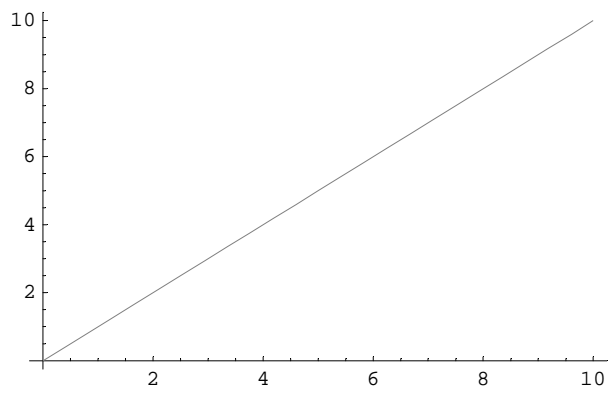
```
In[40]:= Plot[x, {x, 0, 10}, PlotStyle -> Thickness[0.03]];
```



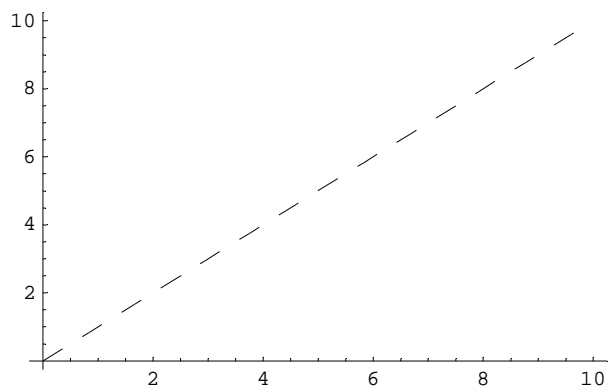
```
In[41]:= Plot[x, {x, 0, 10}, PlotStyle -> RGBColor[0.8,  
0.2, 0.2]];
```



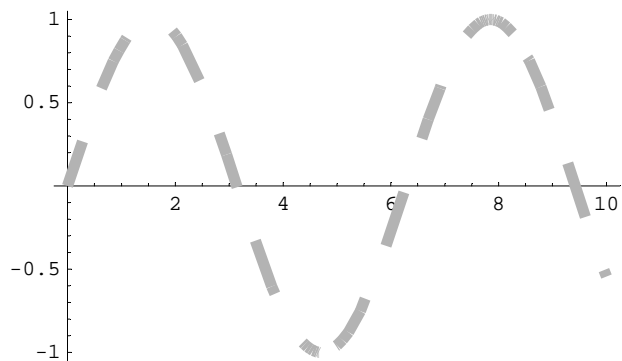
```
In[42]:= Plot[x, {x, 0, 10}, PlotStyle -> GrayLevel[0.5]];
```



```
In[43]:= Plot[x, {x, 0, 10}, PlotStyle -> Dashing[{0.04}]];
```



```
In[44]:= Plot[Sin[x], {x, 0, 10}, PlotStyle -> {{Thickness[
0.02], GrayLevel[0.7], Dashing[{0.1}]}}];
```



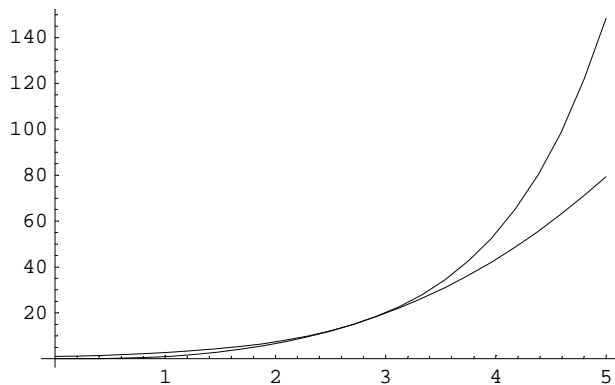
### 4.3. Mehrere Graphen in einem Bild

#### ■ Plusieurs graphes en une image

Man kann auch eine Menge von Funktionen auf einmal ausgeben:

■ On peut aussi sortir un ensemble de fonctions à la fois:

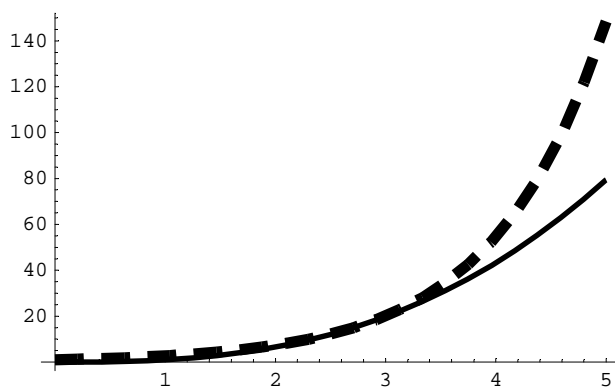
```
In[45]:= Plot[{E^x, x^E}, {x, 0, 5};
```



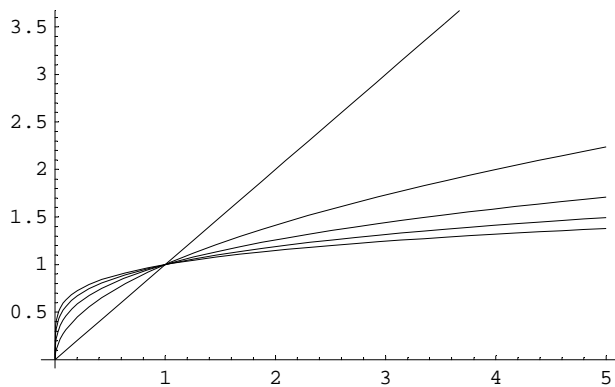
Da es manchmal schwierig ist, die Kurven zu unterscheiden, ist es angebracht, diese verschieden darzustellen:

■ Comme il est parfois difficile de distinguer les courbes, il est préférable de les représenter de différentes façons:

```
In[46]:= Plot[{E^x, x^E}, {x, 0, 5}, PlotStyle -> {
{Thickness[0.02], Dashing[{0.05, 0.03]}],
{Thickness[0.01]} }
];
```



```
In[47]:= Plot[Evaluate[Table[x^(1/n), {n,5}]], {x, 0, 5}];
```



```
In[48]:= Evaluate[Table[x^(1/n), {n,5}]]
```

```
Out[48]= {x, sqrt(x), x^(1/3), x^(1/4), x^(1/5)}
```

```
In[49]:= Table[x^(1/n), {n,5}]
```

```
Out[49]= {x, sqrt(x), x^(1/3), x^(1/4), x^(1/5)}
```

"Evaluate" steuert die Operationshierarchie. Bevor die Plotpunkte berechnet werden können, muss die Tabelle generiert werden! Versuche Dich nun in eigenen Plots!

■ "Evaluate" dirige la hiérarchie des opérations. Avant de pouvoir calculer les points "plot", il faut générer le tableau! Essaie des plots à toi!

## 4.4. Parametrisierte Kurven

### ■ Courbes paramétrisées

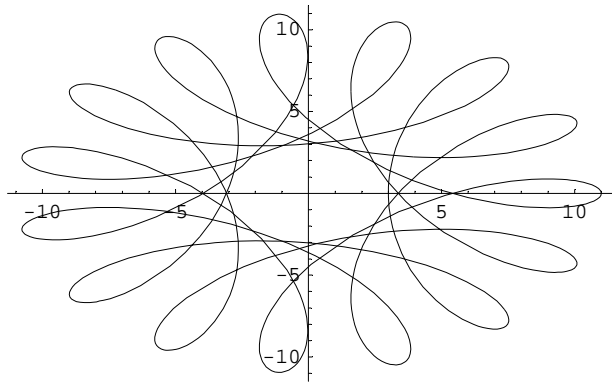
In einem kartesischen Koordinatensystem in der Ebene kann eine Kurve durch einen Vektor gegeben werden, der von einer Variable (Parameter) abhängt. Beispiel:

■ Dans un système de coordonnées cartésien, dans le plan, une courbe peut être donnée par un vecteur qui dépend d'une variable (paramètre). Exemple:

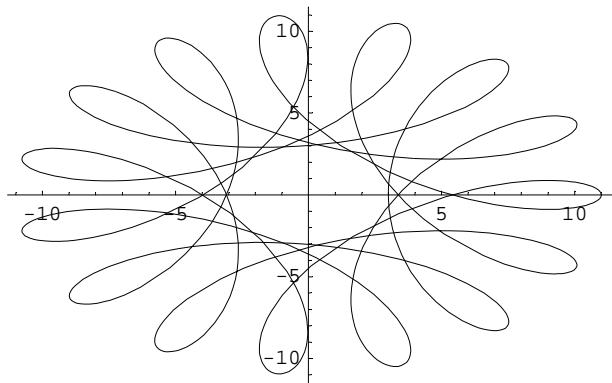
```
In[50]:= v[t_]:= {4 Cos[-11 t/4] + 7 Cos[t], 4 Sin[-11 t/4]
                + 7 Sin[t]}; v[t]
```

```
Out[50]= {7 Cos[t] + 4 Cos[11 t/4], 7 Sin[t] - 4 Sin[11 t/4]}
```

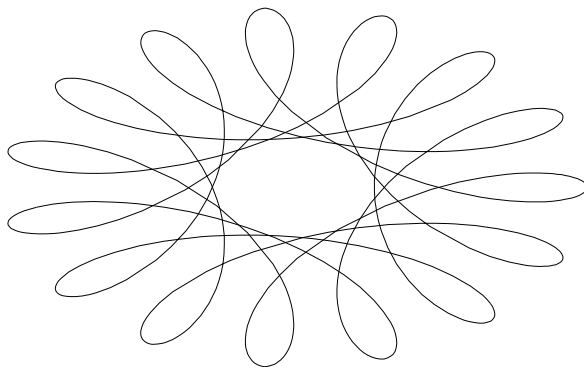
```
In[51]:= ParametricPlot[v[t], {t, 0, 8 Pi}];
```



```
In[52]:= ParametricPlot[{4 Cos[-11 t/4] + 7 Cos[t],
  4 Sin[-11 t/4] + 7 Sin[t]},
  {t, 0, 8 Pi}];
```



```
In[53]:= ParametricPlot[{4 Cos[-11 t/4] + 7 Cos[t],
  4 Sin[-11 t/4] + 7 Sin[t]},
  {t, 0, 8 Pi}, Axes -> None];
```



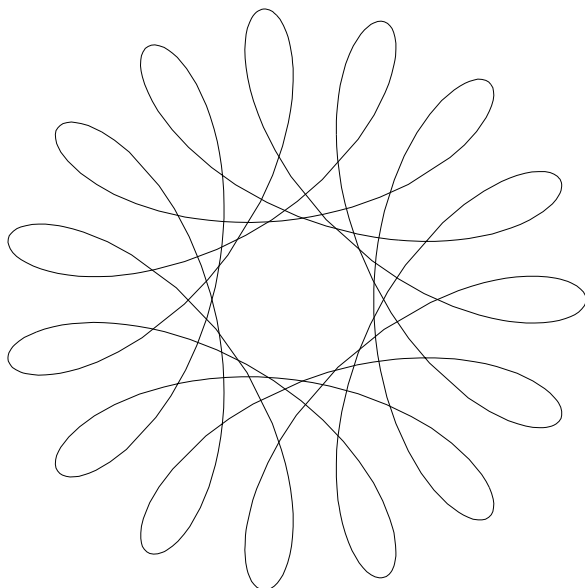
Das Achsenverhältnis:

■ Rapport axial:

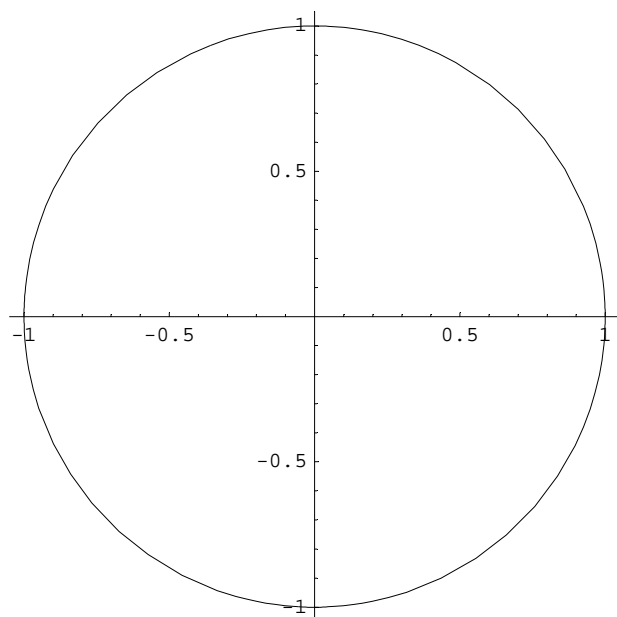
```
In[54]:= AspectRatio /. Options[Plot]
```

```
Out[54]=  $\frac{1}{\text{GoldenRatio}}$ 
```

```
In[55]:= ParametricPlot[{4 Cos[-11 t/4] + 7 Cos[t],  
4 Sin[-11 t/4] + 7 Sin[t]},  
{t, 0, 8 Pi}, Axes -> None,  
AspectRatio -> Automatic];
```



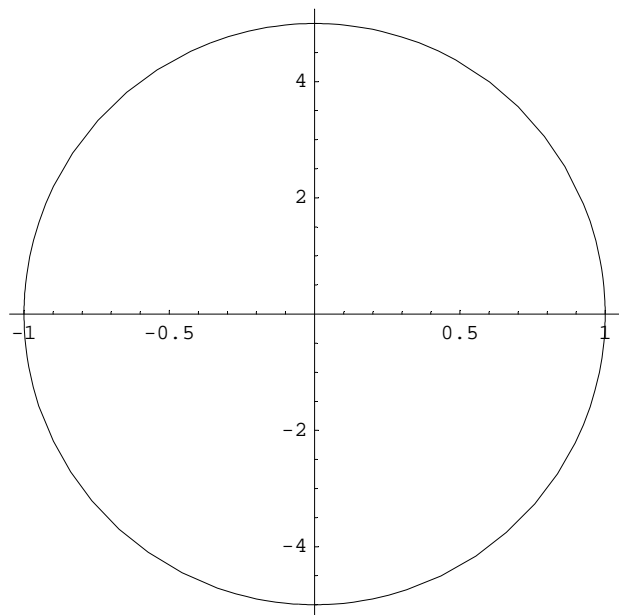
```
In[56]:= ParametricPlot[{Cos[t], Sin[t]}, {t, 0, 2 Pi},  
AspectRatio -> 1];
```



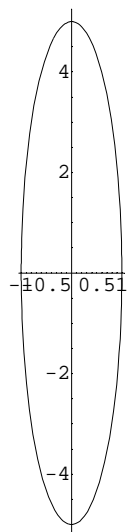
Man beachte den Massstab:

■ Tenir compte de l'échelle:

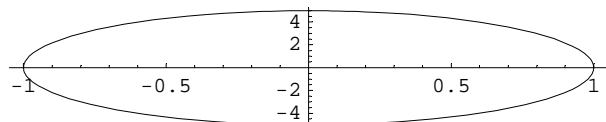
```
In[57]:= ParametricPlot[{Cos[t], 5 Sin[t]}, {t, 0, 2 Pi},  
  AspectRatio -> 1];
```



```
In[58]:= ParametricPlot[{Cos[t], 5 Sin[t]}, {t, 0, 2 Pi},  
  AspectRatio -> 5];
```



```
In[59]:= ParametricPlot[{Cos[t], 5 Sin[t]}, {t, 0, 2 Pi},  
  AspectRatio -> 1/5];
```



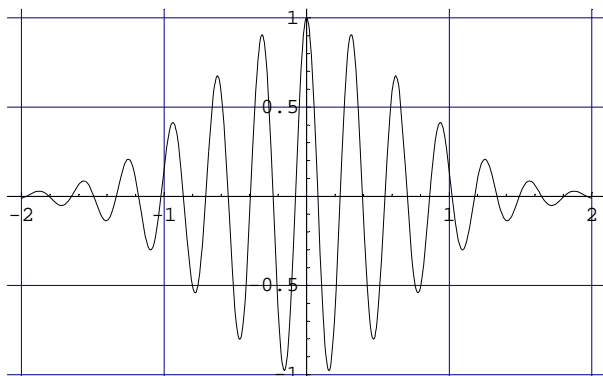
## 4.5. Weitere Optionen

### ■ Autres options

Z.B. ein Gitter:

■ P. ex. une grille:

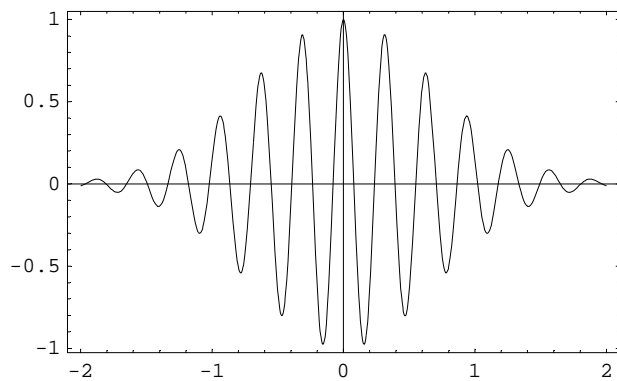
```
In[60]:= Plot[E^-x^2 Cos[20x], {x,-2,2},  
            GridLines -> Automatic];
```



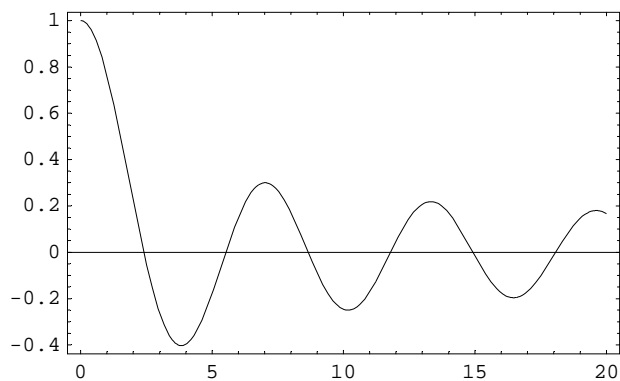
Oder ein Rahmen:

■ Ou un cadre:

```
In[61]:= Plot[E^-x^2 Cos[20x], {x,-2,2}, Frame -> True];
```



```
In[62]:= Plot[BesselJ[0, x], {x, 0, 20}, Frame -> True];
```



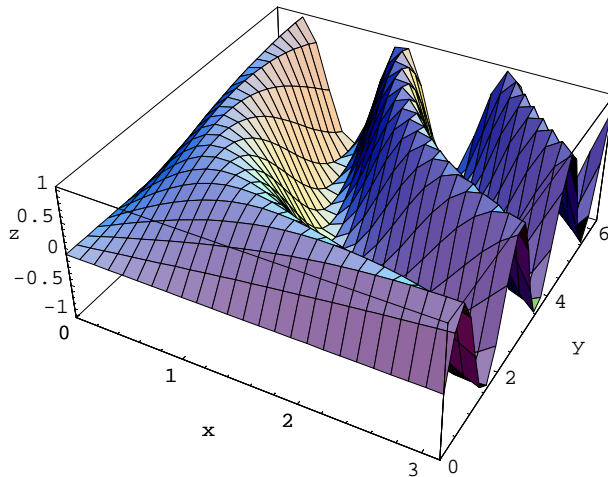
## 4.6. Plots von Flächen im Raum

### ■ Plots de surfaces dans l'espace

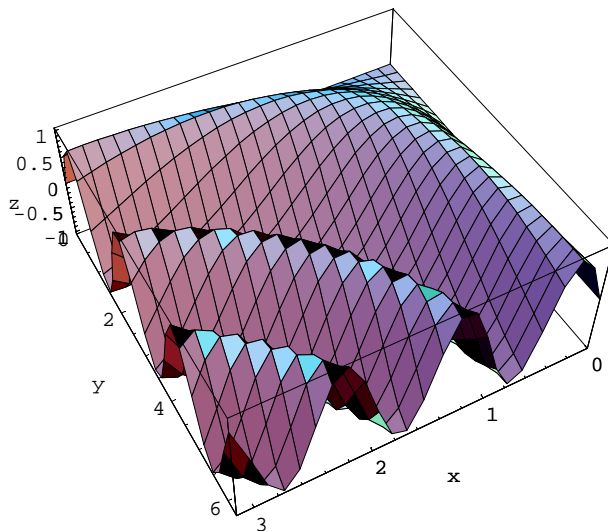
Probiere aus! (Hier werden die Achsen beschriftet):

■ Essaie! (Ici on met une inscription sur les axes:)

```
In[63]:= Plot3D[Sin[x y],{x, 0, Pi}, {y, 0, 2 Pi},
  AxesLabel -> {"x", "y", "z"}];
```



```
In[64]:= Show[%, ViewPoint->{1.050, 1.910, 2.440}];
```

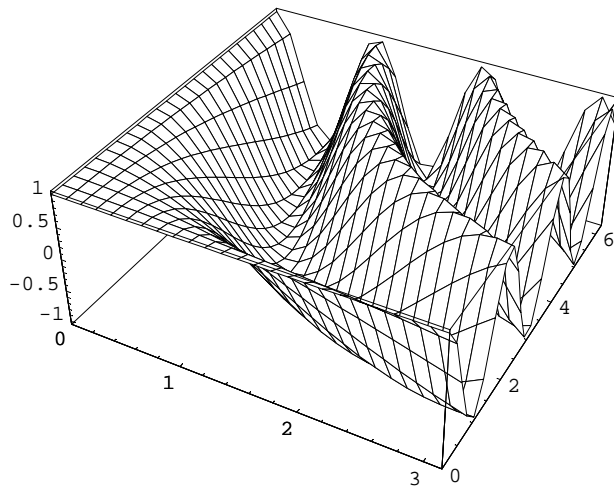


Im Menu unter ("Action", "Prepare Input") neu: "Input", "3D ViewPoint Selector" findest Du eine Hilfe, um den Blickpunkt zu verändern. Du kannst da schieben und dann z.B. die Insert-Taste (Copy-Paste) betätigen, nachdem allerdings der Cursor in einer neuen Zelle parkiert worden ist. Du bekommst dann ein lauffähiges Programm. Probiere die Sache aus!

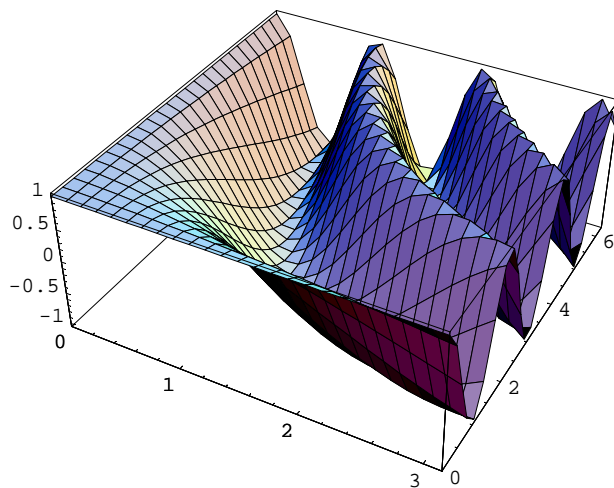
■ Tu trouves dans le menu sous ("Action", "Prepare Input") nouveau: "Input", "3D ViewPoint Selector" une aide pour changer le point de vue. Là tu peux déplacer et ensuite p.ex. appuyer sur la touche insert, après avoir note bien parqué le curseur dans une nouvelle cellule. Tu obtiendras un programme qui fonctionne. Essaie-le!

```
In[65]:= Remove["Global`*"]
```

```
In[66]:= Plot3D[Cos[x y],{x, 0, Pi}, {y, 0, 2 Pi},  
Shading -> False];
```



```
In[67]:= Plot3D[Cos[x y],{x, 0, Pi}, {y, 0, 2 Pi},  
Shading -> True];
```



---

## 4.7. Options für Farben, Beleuchtung, ....

### ■ Options pour couleurs, éclairage, ....

Probiere aus: ■ Essaie!

```
In[68]:= Options[Plot3D]
```

```
Out[68]= {AmbientLight → GrayLevel[0], AspectRatio → Automatic, Axes → True,
  AxesEdge → Automatic, AxesLabel → None, AxesStyle → Automatic,
  Background → Automatic, Boxed → True, BoxRatios → {1, 1, 0.4},
  BoxStyle → Automatic, ClipFill → Automatic, ColorFunction → Automatic,
  ColorFunctionScaling → True, ColorOutput → Automatic,
  Compiled → True, DefaultColor → Automatic, DefaultFont → $DefaultFont,
  DisplayFunction → $DisplayFunction, Epilog → {}, FaceGrids → None,
  FormatType → $FormatType, HiddenSurface → True, ImageSize → Automatic,
  Lighting → True, LightSources → {{{1., 0., 1.}, RGBColor[1, 0, 0]},
  {{1., 1., 1.}, RGBColor[0, 1, 0]}, {{0., 1., 1.}, RGBColor[0, 0, 1]}},
  Mesh → True, MeshStyle → Automatic, Plot3Matrix → Automatic, PlotLabel → None,
  PlotPoints → 25, PlotRange → Automatic, PlotRegion → Automatic, Prolog → {},
  Shading → True, SphericalRegion → False, TextStyle → $TextStyle, Ticks → Automatic,
  ViewCenter → Automatic, ViewPoint → {1.3, -2.4, 2.}, ViewVertical → {0., 0., 1.}}
```

```
In[69]:= LightSources /. Options[Plot3D]
```

```
Out[69]= {{{1., 0., 1.}, RGBColor[1, 0, 0]},
  {{1., 1., 1.}, RGBColor[0, 1, 0]}, {{0., 1., 1.}, RGBColor[0, 0, 1]}}
```

```
In[70]:= Lighting /. Options[Plot3D]
```

```
Out[70]= True
```

Probieren aus: ■ Essai!

```
In[71]:= ??SetOptions
```

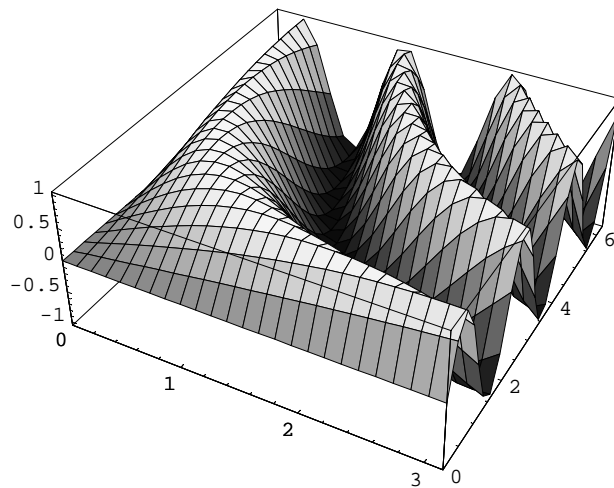
```
SetOptions[s, name1->value1, name2->value2, ... ] sets the specified default
options for a symbol s. SetOptions[stream, ... ] or SetOptions["name", ... ]
sets options associated with a particular stream. SetOptions[object, ... ] sets
options associated with an external object such as a NotebookObject. Mehr...
```

```
Attributes[SetOptions] = {Protected}
```

```
In[72]:= SetOptions[Plot3D, Lighting -> False]
```

```
Out[72]= {AmbientLight → GrayLevel[0], AspectRatio → Automatic, Axes → True,
  AxesEdge → Automatic, AxesLabel → None, AxesStyle → Automatic,
  Background → Automatic, Boxed → True, BoxRatios → {1, 1, 0.4},
  BoxStyle → Automatic, ClipFill → Automatic, ColorFunction → Automatic,
  ColorFunctionScaling → True, ColorOutput → Automatic,
  Compiled → True, DefaultColor → Automatic, DefaultFont → $DefaultFont,
  DisplayFunction → $DisplayFunction, Epilog → {}, FaceGrids → None,
  FormatType → $FormatType, HiddenSurface → True, ImageSize → Automatic,
  Lighting → False, LightSources → {{{1., 0., 1.}, RGBColor[1, 0, 0]},
  {{1., 1., 1.}, RGBColor[0, 1, 0]}, {{0., 1., 1.}, RGBColor[0, 0, 1]}},
  Mesh → True, MeshStyle → Automatic, Plot3Matrix → Automatic, PlotLabel → None,
  PlotPoints → 25, PlotRange → Automatic, PlotRegion → Automatic, Prolog → {},
  Shading → True, SphericalRegion → False, TextStyle → $TextStyle, Ticks → Automatic,
  ViewCenter → Automatic, ViewPoint → {1.3, -2.4, 2.}, ViewVertical → {0., 0., 1.}}
```

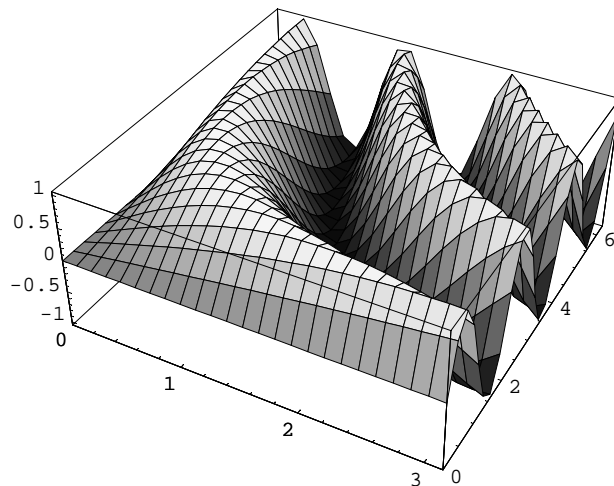
```
In[73]:= Plot3D[Sin[x y],{x, 0, Pi}, {y, 0, 2 Pi}];
```



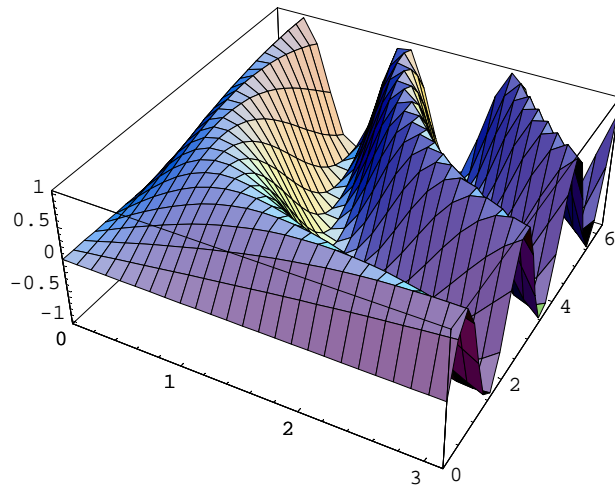
Oder auf einmal:

■ Ou à la fois:

```
In[74]:= Plot3D[Sin[x y],{x, 0, Pi}, {y, 0, 2 Pi},  
Lighting -> False];
```



```
In[75]:= Plot3D[Sin[x y],{x, 0, Pi}, {y, 0, 2 Pi},
  Lighting -> True];
```



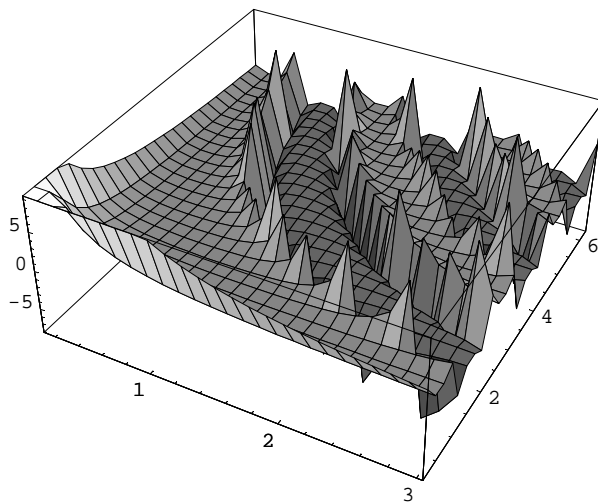
Probiere die Options ein wenig aus!

■ Essaie un peu les options!

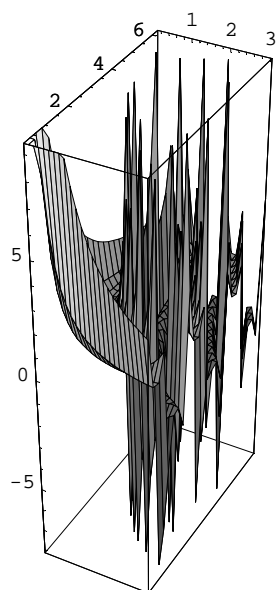
```
In[76]:= ?ClipFill
```

ClipFill is an option for Plot3D and related functions  
that specifies how clipped parts of surfaces are to be drawn. Mehr...

```
In[77]:= Plot3D[1/Sin[x y],{x, 0.1, Pi-0.1}, {y, 0.1, 2 Pi},
  ClipFill -> None];
```



```
In[78]:= Plot3D[1/Sin[x y],{x, 0.1, Pi-0.1}, {y, 0.1, 2 Pi},
  BoxRatios -> {1,2,4}];
```



## 4.8. Arbeiten mit Daten

### ■ Travailler avec des données

#### 4.8.1. Allgemeines

##### ■ Notions générales

Z.B. mit "ListPlot" und "ListPlot3D"

■ P.ex. avec "ListPlot" et "ListPlot3D"

```
In[79]:= ??ListPlot
```

ListPlot[{y1, y2, ...}] plots points corresponding to a list of values. The x coordinates are by default taken to be 1, 2, ... . ListPlot[{x1, y1}, {x2, y2}, ...] plots a list of points with specified x and y coordinates. Mehr...

```
Attributes[ListPlot] = {Protected}
```

```
Options[ListPlot] = {AspectRatio ->  $\frac{1}{\text{GoldenRatio}}$ , Axes -> Automatic, AxesLabel -> None,
  AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic, ColorOutput -> Automatic,
  DefaultColor -> Automatic, DefaultFont -> $DefaultFont, DisplayFunction -> $DisplayFunction,
  Epilog -> {}, FormatType -> $FormatType, Frame -> False, FrameLabel -> None, FrameStyle -> Automatic,
  FrameTicks -> Automatic, GridLines -> None, ImageSize -> Automatic, PlotJoined -> False,
  PlotLabel -> None, PlotRange -> Automatic, PlotRegion -> Automatic, PlotStyle -> Automatic,
  Prolog -> {}, RotateLabel -> True, TextStyle -> $TextStyle, Ticks -> Automatic}
```

```
In[80]:= ??ListPlot3D
```

ListPlot3D[array] generates a three-dimensional plot of a surface representing an array of height values. ListPlot3D[array, shades] generates a plot with each element of the surface shaded according to the specification in shades. Mehr...

```
Attributes[ListPlot3D] = {Protected}
```

```
Options[ListPlot3D] =
```

```
{AmbientLight → GrayLevel[0], AspectRatio → Automatic, Axes → True, AxesEdge → Automatic,
AxesLabel → None, AxesStyle → Automatic, Background → Automatic, Boxed → True,
BoxRatios → {1, 1, 0.4}, BoxStyle → Automatic, ClipFill → Automatic, ColorFunction → Automatic,
ColorFunctionScaling → True, ColorOutput → Automatic, DefaultColor → Automatic,
DefaultFont → $DefaultFont, DisplayFunction → $DisplayFunction, Epilog → {}, FaceGrids → None,
FormatType → $FormatType, HiddenSurface → True, ImageSize → Automatic, Lighting → True,
LightSources → {{1., 0., 1.}, RGBColor[1, 0, 0]}, {{1., 1., 1.}, RGBColor[0, 1, 0]},
{{0., 1., 1.}, RGBColor[0, 0, 1]}}, Mesh → True, MeshRange → Automatic, MeshStyle → Automatic,
Plot3Matrix → Automatic, PlotLabel → None, PlotRange → Automatic, PlotRegion → Automatic,
Prolog → {}, Shading → True, SphericalRegion → False, TextStyle → $TextStyle, Ticks → Automatic,
ViewCenter → Automatic, ViewPoint → {1.3, -2.4, 2.}, ViewVertical → {0., 0., 1.}}
```

Erst Daten generieren:

■ D'abord générer des données:

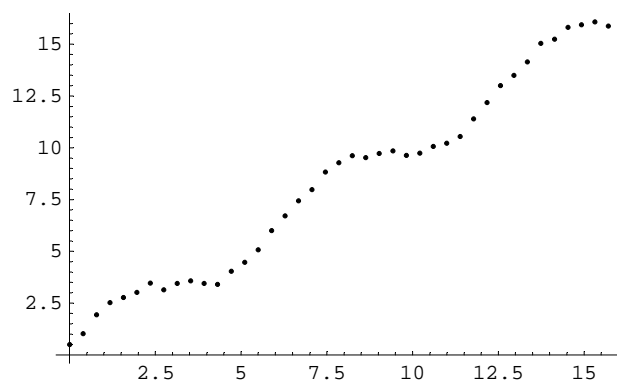
```
In[81]:= data = N[Table[{x, Sin[x] + x + 0.5 Random[]},
{x, 0, 5 Pi, Pi/8}]]
```

```
Out[81]= {{0., 0.495083}, {0.392699, 1.02267}, {0.785398, 1.93979},
{1.1781, 2.52274}, {1.5708, 2.77231}, {1.9635, 3.01449}, {2.35619, 3.47185},
{2.74889, 3.13403}, {3.14159, 3.44368}, {3.53429, 3.5747}, {3.92699, 3.45122},
{4.31969, 3.40938}, {4.71239, 4.04212}, {5.10509, 4.47738},
{5.49779, 5.07444}, {5.89049, 6.00353}, {6.28319, 6.71473},
{6.67588, 7.43788}, {7.06858, 7.98662}, {7.46128, 8.82733}, {7.85398, 9.27882},
{8.24668, 9.62249}, {8.63938, 9.5258}, {9.03208, 9.72418}, {9.42478, 9.85453},
{9.81748, 9.63944}, {10.2102, 9.7351}, {10.6029, 10.0676}, {10.9956, 10.2238},
{11.3883, 10.5419}, {11.781, 11.3973}, {12.1737, 12.1772}, {12.5664, 12.9925},
{12.9591, 13.4962}, {13.3518, 14.151}, {13.7445, 15.041}, {14.1372, 15.2336},
{14.5299, 15.812}, {14.9226, 15.9381}, {15.3153, 16.0749}, {15.708, 15.8728}}
```

Dann Daten ploten:

■ Ensuite "ploter" les données:

```
In[82]:= dataplot = ListPlot[data];
```



Polynom durch diese Punkte als Funktion von x:

Polynôme à travers ces points comme une fonction de x:

`In[83]:= ??InterpolatingPolynomial`

`InterpolatingPolynomial[data, var]` gives a polynomial in the variable `var` which provides an exact fit to a list of data. The data can have the forms `{x1, f1}`, `{x2, f2}`, ... or `{f1, f2, ...}`, where in the second case, the `xi` are taken to have values 1, 2, ... . The `fi` can be replaced by `{fi, dfi, ddfi, ...}`, specifying derivatives at the points `xi`. Mehr...

`Attributes[InterpolatingPolynomial] = {Protected}`

`Options[InterpolatingPolynomial] = {Modulus -> 0}`

`In[84]:= InterpolatingPolynomial[data, x]`

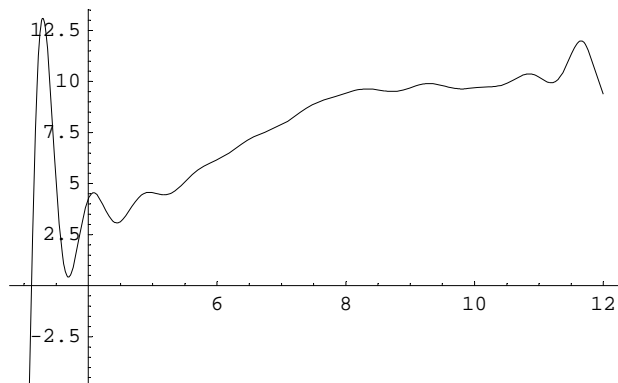
`Out[84]= 0.495083 +`  

$$\begin{aligned} & (1.3435 + (1.26296 + (-1.9917 + (1.26931 + (-0.356241 + (-0.0111603 + (-0.0924798 + \\ & (0.271334 + (-0.309284 + (0.230771 + \\ & (-0.130027 + (0.0591378 + (-0.0226376 + \\ & (0.00753882 + (-0.00225315 + (0.00062332 + \\ & (-0.000164065 + (0.0000417999 + \\ & (-0.0000103277 + (2.44697 \times 10^{-6} + \\ & (-5.46525 \times 10^{-7} + (1.13052 \times 10^{-7} + \\ & (-2.1273 \times 10^{-8} + (3.55527 \times 10^{-9} + \\ & (-5.03498 \times 10^{-10} + (5.24806 \times 10^{-11} + \\ & (-1.05591 \times 10^{-12} + (-1.37654 \times 10^{-12} + \\ & (5.01287 \times 10^{-13} + (-1.25382 \times 10^{-13} + \\ & (2.64785 \times 10^{-14} + (-5.02029 \times 10^{-15} + \\ & (8.77875 \times 10^{-16} + (-1.43593 \times 10^{-16} + \\ & (2.21492 \times 10^{-17} + (-3.23784 \times 10^{-18} + \\ & (4.49951 \times 10^{-19} + (-5.956 \times 10^{-20} + \\ & (7.5195 \times 10^{-21} - 9.06216 \times 10^{-22} (-15.3153 + \\ & x)) (-14.9226 + x)) (-14.5299 + x)) \\ & (-14.1372 + x)) (-13.7445 + x)) \\ & (-13.3518 + x)) (-12.9591 + x)) \\ & (-12.5664 + x)) (-12.1737 + x)) (-11.781 + \\ & x)) (-11.3883 + x)) (-10.9956 + x)) \\ & (-10.6029 + x)) (-10.2102 + x)) \\ & (-9.81748 + x)) (-9.42478 + x)) \\ & (-9.03208 + x)) (-8.63938 + x)) \\ & (-8.24668 + x)) (-7.85398 + x)) \\ & (-7.46128 + x)) (-7.06858 + x)) \\ & (-6.67588 + x)) (-6.28319 + x)) \\ & (-5.89049 + x)) (-5.49779 + x)) \\ & (-5.10509 + x)) (-4.71239 + x)) \\ & (-4.31969 + x)) (-3.92699 + x)) \\ & (-3.53429 + x)) (-3.14159 + x)) (-2.74889 + x)) \\ & (-2.35619 + x)) (-1.9635 + x)) (-1.5708 + x)) \\ & (-1.1781 + x)) (-0.785398 + x)) (-0.392699 + x)) (0. + x) \end{aligned}$$

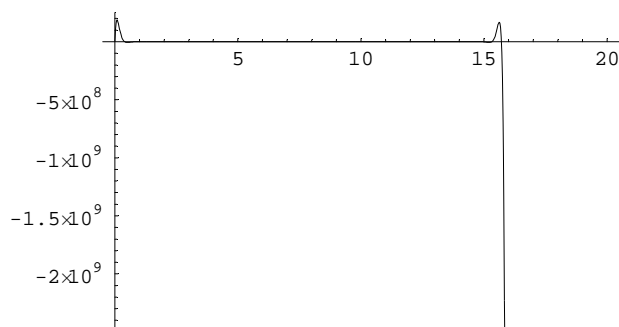
```
In[85]:= Chop[InterpolatingPolynomial[data,x]]
```

```
Out[85]= 0.495083 +
(1.3435 + (1.26296 + (-1.9917 + (1.26931 + (-0.356241 + (-0.0111603 + (-0.0924798 +
(0.271334 + (-0.309284 + (0.230771 +
(-0.130027 + (0.0591378 + (-0.0226376 + (0.00753882 +
(-0.00225315 + (0.00062332 + (-0.000164065 +
(0.0000417999 + (-0.0000103277 +
(2.44697 × 10-6 + (-5.46525 × 10-7 +
(1.13052 × 10-7 + (-2.1273 × 10-8 +
(3.55527 × 10-9 - 5.03498 × 10-10 (-9.42478 +
x) (-9.03208 + x) (-8.63938 + x)
(-8.24668 + x) (-7.85398 + x)
(-7.46128 + x) (-7.06858 + x) (-6.67588 +
x) (-6.28319 + x) (-5.89049 + x)
(-5.49779 + x) (-5.10509 + x)
(-4.71239 + x) (-4.31969 + x) (-3.92699 + x)
(-3.53429 + x) (-3.14159 + x) (-2.74889 + x)
(-2.35619 + x) (-1.9635 + x) (-1.5708 + x)
(-1.1781 + x) (-0.785398 + x) (-0.392699 + x) x
```

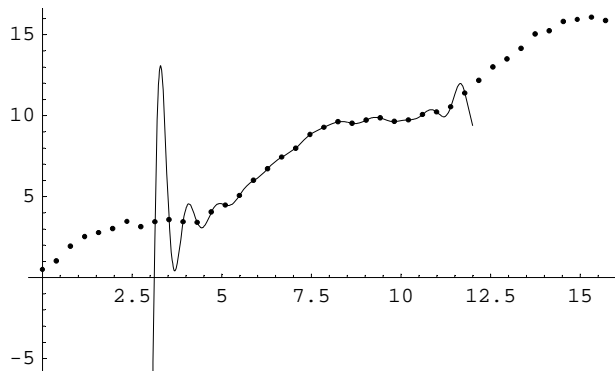
```
In[86]:= interplot = Plot[Evaluate[
InterpolatingPolynomial[data,x]],{x,3,12}];
```



```
In[87]:= Plot[Evaluate[
InterpolatingPolynomial[data,x]],{x,0,20}];
```



```
In[88]:= Show[dataplot, interplot];
```



## 4.8.2. "Regressionskurven" ■ "Courbes de régression"

Arbeiten mit "Fit":

■ Travailler avec "Fit":

```
In[89]:= ??Fit
```

Fit[data, funs, vars] finds a least-squares fit to a list of data as a linear combination of the functions funs of variables vars. The data can have the form  $\{\{x_1, y_1, \dots, f_1\}, \{x_2, y_2, \dots, f_2\}, \dots\}$ , where the number of coordinates  $x, y, \dots$  is equal to the number of variables in the list vars. The data can also be of the form  $\{f_1, f_2, \dots\}$ , with a single coordinate assumed to take values 1, 2,  $\dots$ . The argument funs can be any list of functions that depend only on the objects vars. Mehr...

```
Attributes[Fit] = {Protected}
```

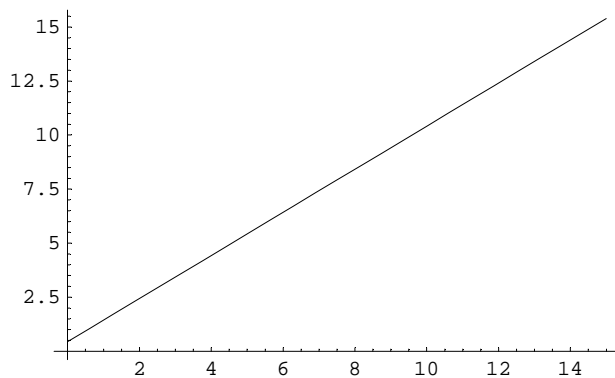
Gerade durch die Punktwolke:

■ En ligne droite à travers le nuage de points:

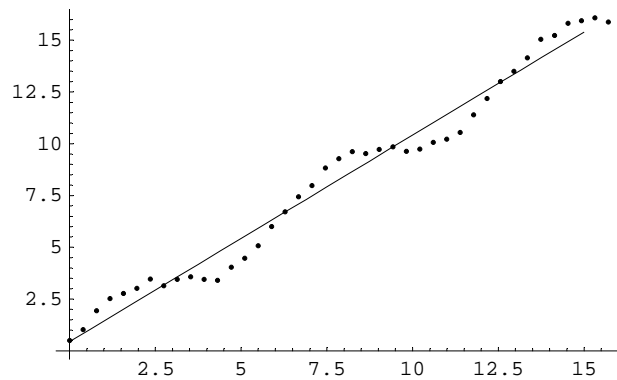
```
In[90]:= trent = Fit[data, {1, x}, x]
```

```
Out[90]= 0.446685 + 0.99655 x
```

```
In[91]:= trentplot1 = Plot[trent, {x, 0, 15}];
```



```
In[92]:= Show[trentplot1, dataplot];
```



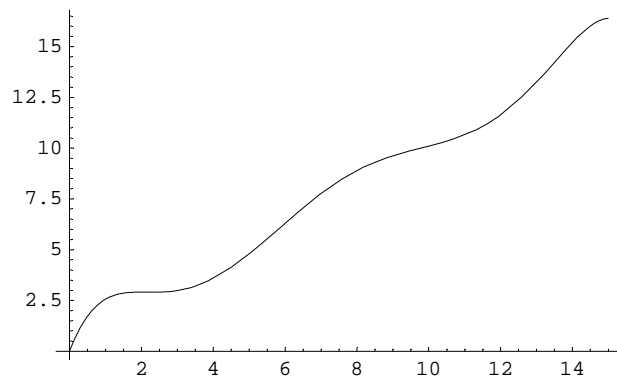
Polynomkurve 6. Grades durch die Punktwolke:

■ Courbe polynomiale de 6e degré à travers le nuage de points:

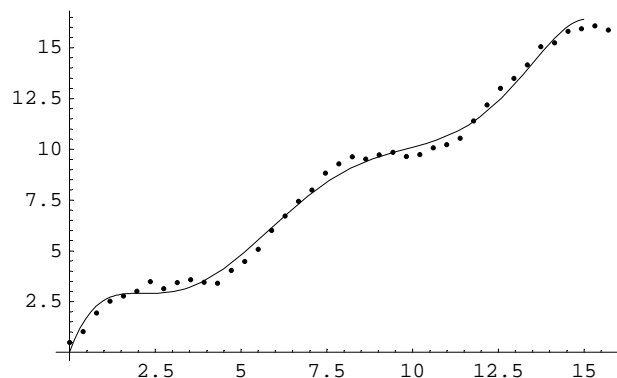
```
In[93]:= trent6 = Fit[data, {1, x, x^2, x^3, x^4,
                             x^5, x^6}, x]
```

```
Out[93]= 0.00723834 + 4.68972 x - 2.80482 x^2 +
          0.758888 x^3 - 0.0930131 x^4 + 0.00527162 x^5 - 0.000112239 x^6
```

```
In[94]:= trentplot6 = Plot[trent6, {x, 0, 15}];
```



```
In[95]:= Show[trentplot6, dataplot];
```



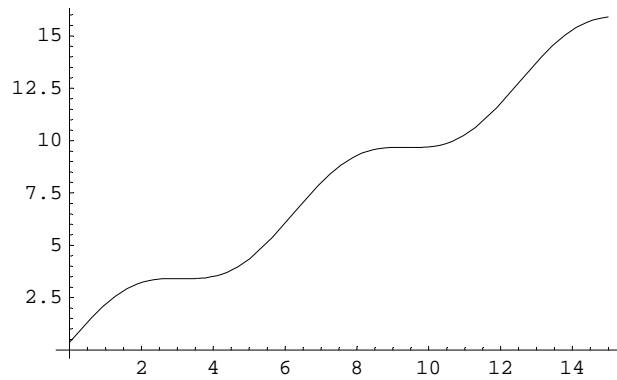
Kurve als Ueberlagerung von Gerade und Cosinus- sowie Sinusfunktion durch die Punktwolke:

■ Courbe comme superposition d'une droite et d'une fonction cosinus ainsi que sinus à travers le nuage de points:

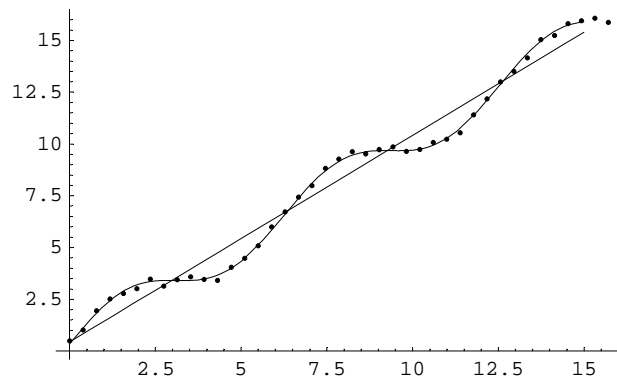
```
In[96]:= trentcc = Fit[data, {1, x, Cos[x], Sin[x]}, x]
```

```
Out[96]= 0.317202 + 0.997138 x + 0.0400338 Cos[x] + 1.01829 Sin[x]
```

```
In[97]:= trentplotcc = Plot[trentcc, {x, 0, 15}];
```

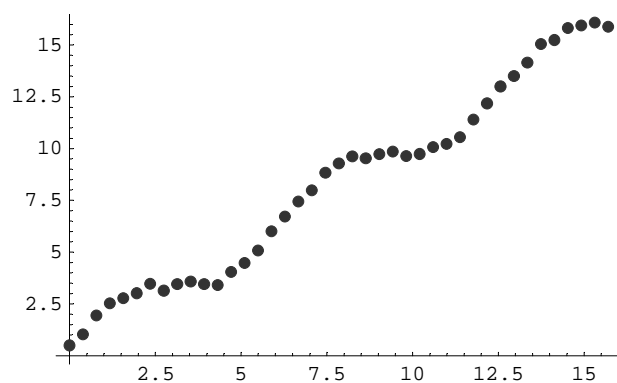


```
In[98]:= Show[trentplotcc, trentplot1, dataplot];
```



Optionen: ■ Options:

```
In[99]:= dataplot1 = ListPlot[data, PlotStyle->{PointSize[0.02],
GrayLevel[0.2]}];
```

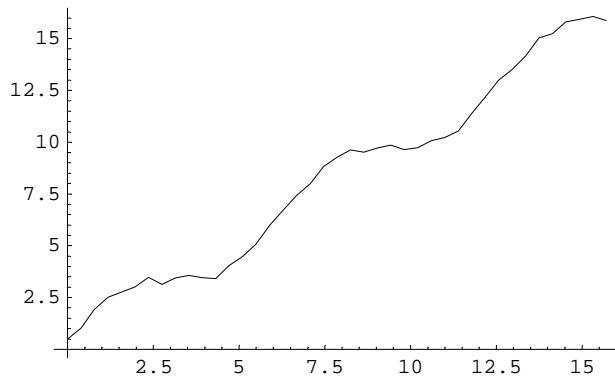


Punkte verbinden:

■ Relier les points:

In[100]:=

```
dataplot2 = ListPlot[data, PlotJoined->True];
```

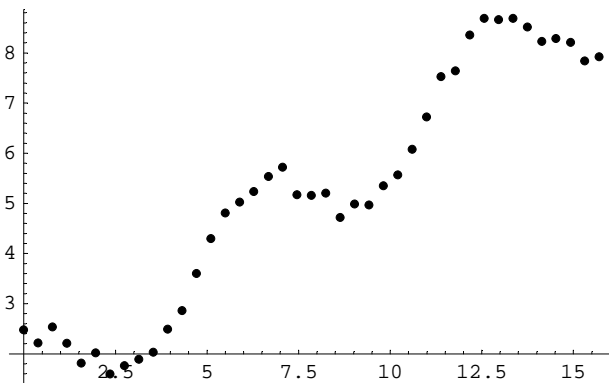


ListPlot kann auf einmal nur eine Datenmenge plotten, jedoch kann man solche Plots überlagern:

■ ListPlot ne peut "ploter" qu'un ensemble de données à la fois, on peut cependant superposer de tels plots:

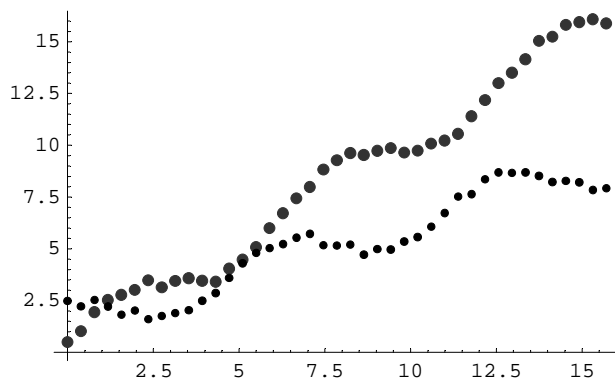
In[101]:=

```
data1 = N[Table[{x, 1 + Cos[x] + 0.5 x +
  0.5 Random[]}, {x, 0, 5 Pi, Pi/8}]];
dataplot3 = ListPlot[data1,
  PlotStyle->{PointSize[0.014]}];
```



In[103]:=

```
Show[dataplot1,dataplot3];
```



### 4.8.3. Unterdrückung des Graphik-Outputs (Zeit sparen!) ■ Supprimer la sortie des graphes (économie de temps!)

Studiere: Zuerst Daten generieren (Tabelle mit 3 Listen zu je 5 Elementen, "DisplayFunction"):

■ Etudie: D'abord générer les données (tableau à 3 listes à 5 éléments, "DisplayFunction"):

In[104]:=

```
myMatrix = Table[Random[], {3}, {5}]
```

Out[104]=

```
{0.254526, 0.951846, 0.093696, 0.672727, 0.330327},
{0.749974, 0.304512, 0.166322, 0.601042, 0.417562},
{0.995414, 0.713963, 0.708045, 0.335777, 0.306895}}
```

In[105]:=

```
??DisplayFunction
```

DisplayFunction is an option for graphics and sound functions that specifies a function to apply to graphics and sound objects in order to display them. Mehr...

```
Attributes[DisplayFunction] = {Protected}
```

Graphiken rechnen, jedoch nicht einzeln ausgeben:

■ Calculer des graphes, cependant ne pas les sortir:

In[106]:=

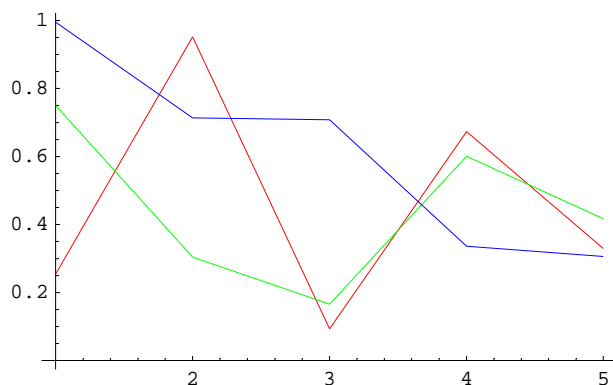
```
plot1 = ListPlot[myMatrix[[1]],
  PlotStyle->RGBColor[1,0,0], PlotJoined->True,
  DisplayFunction->Identity];
plot2 = ListPlot[myMatrix[[2]],
  PlotStyle->RGBColor[0,1,0], PlotJoined->True,
  DisplayFunction->Identity];
plot3 = ListPlot[myMatrix[[3]],
  PlotStyle->RGBColor[0,0,1], PlotJoined->True,
  DisplayFunction->Identity];
```

Graphiken ausgeben, alle zusammen in einem Bild:

■ Sortir les graphes, tous ensemble en une image:

In[109]:=

```
Show[plot1, plot2, plot3,
  DisplayFunction :> $DisplayFunction];
```



Erklärungen: ■ Explications:

In[110]:=

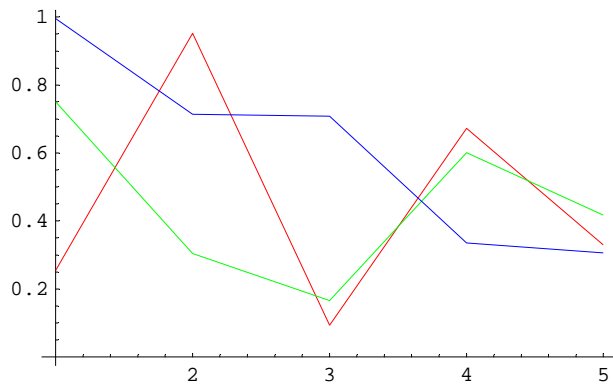
**??\$DisplayFunction**

\$DisplayFunction gives the default setting  
for the option DisplayFunction in graphics functions. Mehr...

\$DisplayFunction = Display[\$Display, #1] &

In[111]:=

**Show[plot1, plot2, plot3,  
DisplayFunction -> \$DisplayFunction];**



In[112]:=

**??->**

lhs -> rhs represents a rule that transforms lhs to rhs. Mehr...

Attributes[Rule] = {Protected, SequenceHold}

In[113]:=

**??:>**

lhs :> rhs represents a rule that transforms  
lhs to rhs, evaluating rhs only when the rule is used. Mehr...

Attributes[RuleDelayed] = {HoldRest, Protected, SequenceHold}

#### 4.8.4. Plot von 3-dimensionalen Daten ■ Plot de données à 3 dimensions

Erklärungen: ■ Explications:

```
In[114]:=
```

```
??ListContourPlot
```

```
ListContourPlot[array] generates a contour plot from an array of height values. Mehr...
```

```
Attributes[ListContourPlot] = {Protected}
```

```
Options[ListContourPlot] =
```

```
{AspectRatio → 1, Axes → False, AxesLabel → None, AxesOrigin → Automatic,  
AxesStyle → Automatic, Background → Automatic, ColorFunction → Automatic,  
ColorFunctionScaling → True, ColorOutput → Automatic, ContourLines → True,  
Contours → 10, ContourShading → True, ContourSmoothing → True, ContourStyle → Automatic,  
DefaultColor → Automatic, DefaultFont → $DefaultFont, DisplayFunction → $DisplayFunction,  
Epilog → {}, FormatType → $FormatType, Frame → True, FrameLabel → None,  
FrameStyle → Automatic, FrameTicks → Automatic, ImageSize → Automatic,  
MeshRange → Automatic, PlotLabel → None, PlotRange → Automatic, PlotRegion → Automatic,  
Prolog → {}, RotateLabel → True, TextStyle → $TextStyle, Ticks → Automatic}
```

```
In[115]:=
```

```
??ListDensityPlot
```

```
ListDensityPlot[array] generates a smooth density plot from an array of values. Mehr...
```

```
Attributes[ListDensityPlot] = {Protected}
```

```
Options[ListDensityPlot] =
```

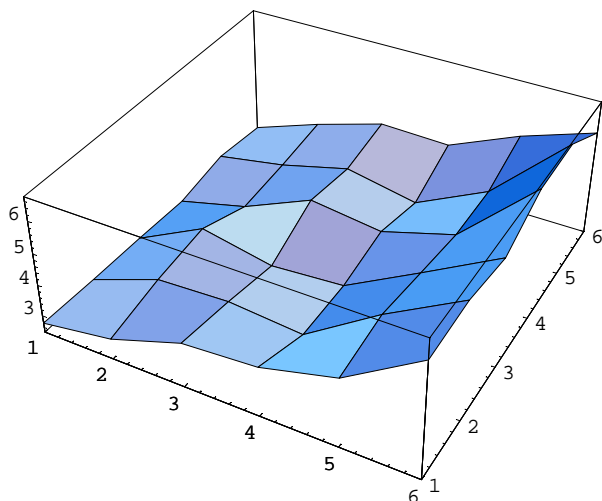
```
{AspectRatio → 1, Axes → False, AxesLabel → None, AxesOrigin → Automatic,  
AxesStyle → Automatic, Background → Automatic, ColorFunction → Automatic,  
ColorFunctionScaling → True, ColorOutput → Automatic, DefaultColor → Automatic,  
DefaultFont → $DefaultFont, DisplayFunction → $DisplayFunction, Epilog → {},  
FormatType → $FormatType, Frame → True, FrameLabel → None, FrameStyle → Automatic,  
FrameTicks → Automatic, ImageSize → Automatic, Mesh → True, MeshRange → Automatic,  
MeshStyle → Automatic, PlotLabel → None, PlotRange → Automatic, PlotRegion → Automatic,  
Prolog → {}, RotateLabel → True, TextStyle → $TextStyle, Ticks → Automatic}
```

Erst Daten generieren:

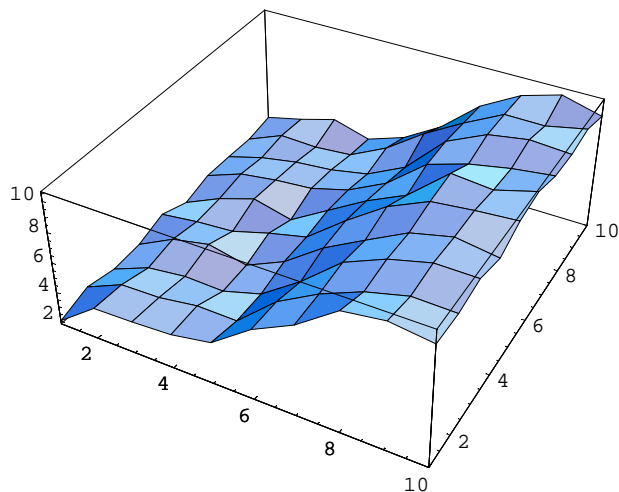
■ D'abord générer des données:

```
In[116]:=
array1 = Table[x + Sin[x] + Random[], {6},
  {x, 1, 6}]; Print[array1];
ListPlot3D[array1];

{{2.74784, 3.09658, 3.86844, 4.02567, 4.62956, 6.11237},
 {2.66127, 3.47913, 3.59256, 3.38045, 4.90903, 6.19672},
 {2.62019, 3.71623, 3.2591, 3.41482, 4.65347, 5.92647},
 {2.54189, 3.08551, 4.03955, 3.74104, 4.40572, 6.5899},
 {2.83353, 3.21986, 3.77844, 3.33004, 4.44466, 6.63936},
 {2.65899, 3.42631, 4.09326, 4.02472, 4.99064, 5.76146}}
```



```
In[118]:=
array2 = Table[x + Sin[x] + Random[], {10},
  {x, 1, 10}]; ListPlot3D[array2];
```



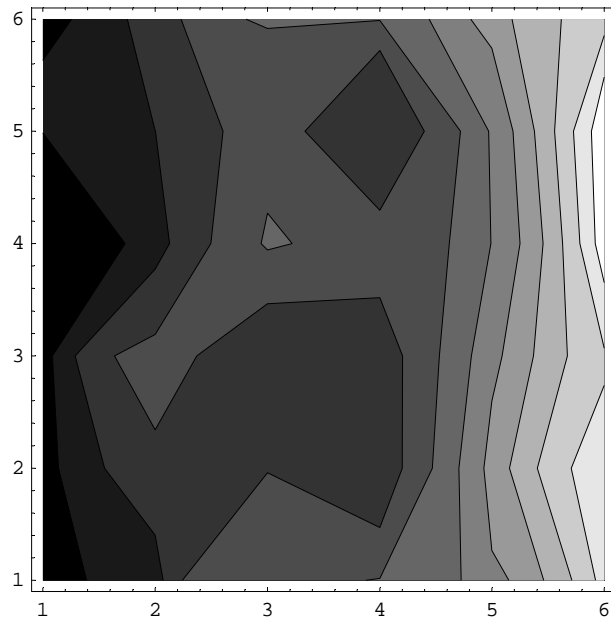
Zur besseren Sichtbarmachung zeichnet *Mathematica* eine Fläche und nicht eine Punktwolke im 3-dimensionalen Raum. `ListContourPlot` zeichnet davon eine

Höhenlinienkarte:

■ Pour une meilleure perceptibilité, *Mathematica* dessine une surface et non pas un nuage de points dans l'espace à trois dimensions. `ListContourPlot` dessine une carte indiquant les courbes à niveau:

```
In[119]:=
```

```
ListContourPlot[array1];
```

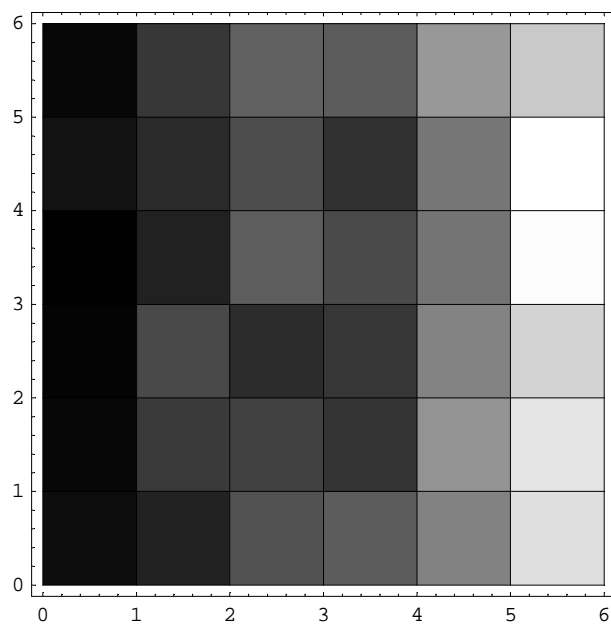


Eine andere Möglichkeit zur Veranschaulichung bietet ListDensityPlot:

ListDensityPlot offre une autre possibilité d'illustration:

```
In[120]:=
```

```
ListDensityPlot[array1];
```



---

## 4.9. Eingebaute Graphikelemente oder Blöcke

### ■ Eléments ou blocs graphiques incorporés

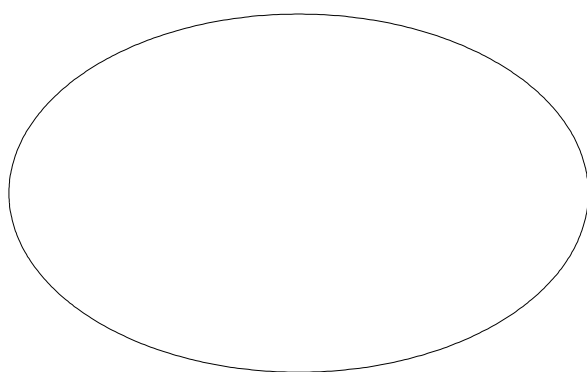
#### 4.9.1. Allgemeines

##### ■ Notions générales

*Mathematica* bietet die Möglichkeit, Linien, Punkte, Kreise, Scheiben und Polygone direkt zu zeichnen. Beispiele:

■ *Mathematica* offre la possibilité de dessigner directement des lignes, points, cercles, disques, polygones. Exemples:

```
In[121]:= Show[Graphics[{Circle[{0,0},.1]}]];
```



Wieso ist aus dem Kreis eine Ellipse geworden?

■ Pourquoi le cercle est-il devenu une ellipse?

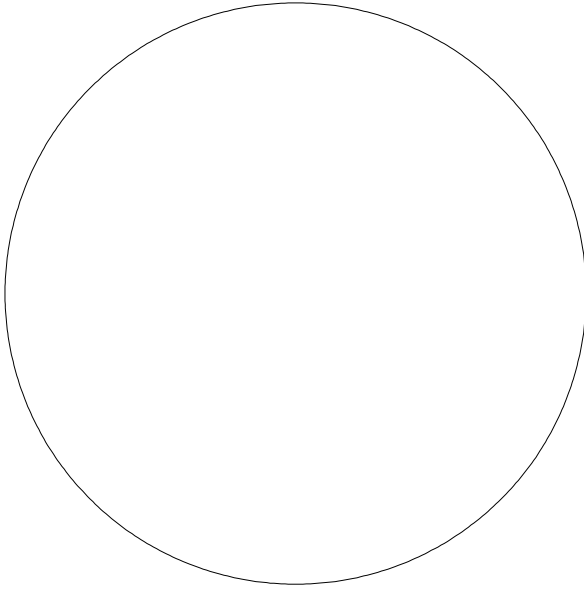
```
In[122]:= Options[Graphics][[1]]
```

```
Out[122]= AspectRatio ->  $\frac{1}{\text{GoldenRatio}}$ 
```

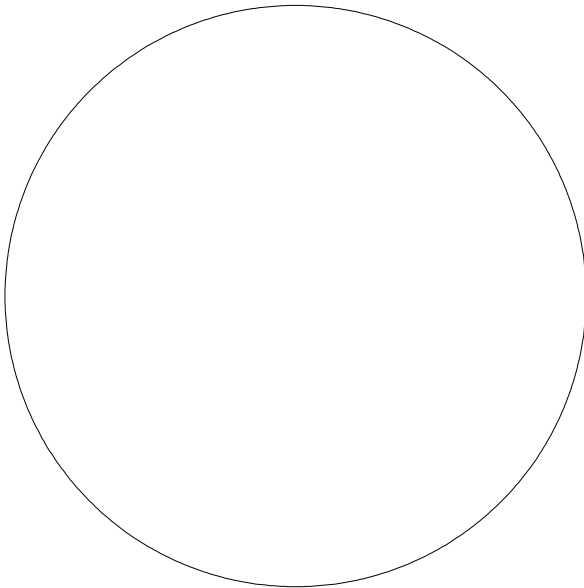
Man muss also das Achsenverhältnis anpassen!

■ Il faut donc adapter le rapport des axes:

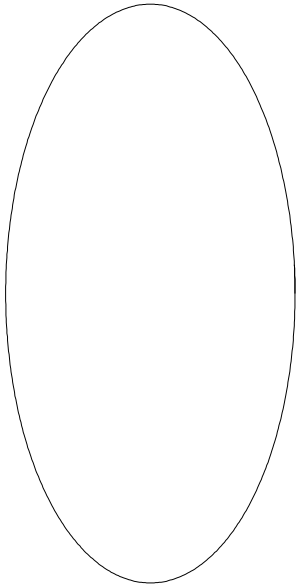
```
In[123]:= Show[Graphics[{Circle[{0,0},.1]}],  
  AspectRatio->1];
```



```
In[124]:= Show[Graphics[{Circle[{0,0},.1]}],  
  AspectRatio->Automatic];
```



```
In[125]:= Show[Graphics[{Circle[{0,0},.1]},  
  AspectRatio->2]];
```



Oder: ■ Ou:

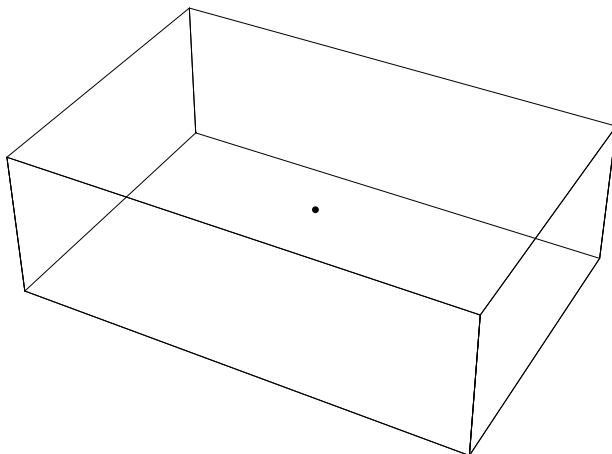
```
In[126]:= Show[Graphics[{  
  PointSize[0.05], Point[{0, 0}], Point[{1, 0}],  
  Line[{{0, 0},{1, 0}}],  
  Circle[{0,0},.1], Circle[{1,0},.1]},  
  AspectRatio->Automatic]];
```



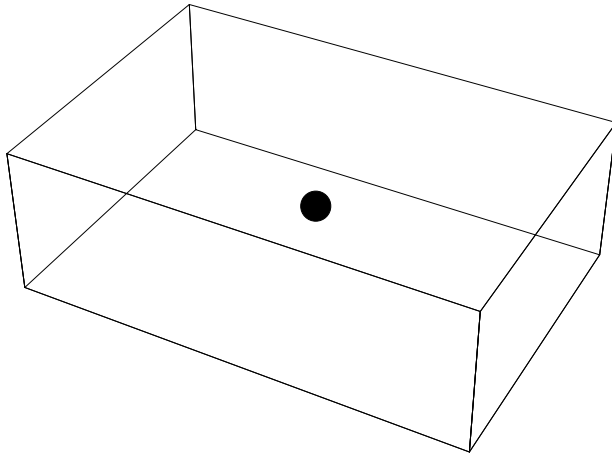
Oder ein Punkt im 3-dimensionalen Raum:

■ Ou un point dans l'espace à trois dimensions:

```
In[127]:= Show[Graphics3D[Point[{3,2,1}]]];
```



```
In[128]:=
  Show[Graphics3D[{PointSize[0.05],
    Point[{3,2,1}]}]];
```



Oder erst Liste generieren:

■ Ou générer d'abord une liste:

```
In[129]:=
  pts = Table[Random[], {4}, {3}]
```

```
Out[129]=
  {{0.65511, 0.860532, 0.213686}, {0.705759, 0.0158655, 0.25467},
  {0.938153, 0.333522, 0.155321}, {0.299344, 0.426078, 0.843024}}
```

Dann Liste manipulieren:

■ Ensuite manipuler la liste:

```
In[130]:=
  ??Map

Map[f, expr] or f /@ expr applies f to each element on the first level in expr. Map[
  f, expr, levelspec] applies f to parts of expr specified by levelspec. Mehr...

Attributes[Map] = {Protected}

Options[Map] = {Heads -> False}
```

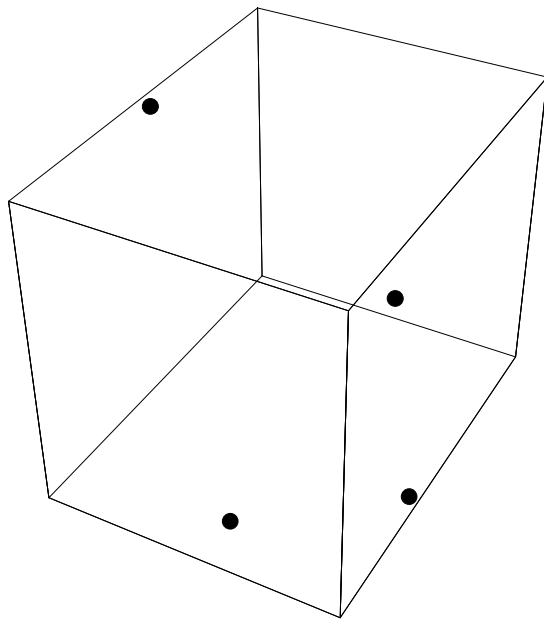
```
In[131]:=
  graphicPts = Map[Point, pts]
```

```
Out[131]=
  {Point[{0.65511, 0.860532, 0.213686}], Point[{0.705759, 0.0158655, 0.25467}],
  Point[{0.938153, 0.333522, 0.155321}], Point[{0.299344, 0.426078, 0.843024}]}
```

Dann davon Plot machen:

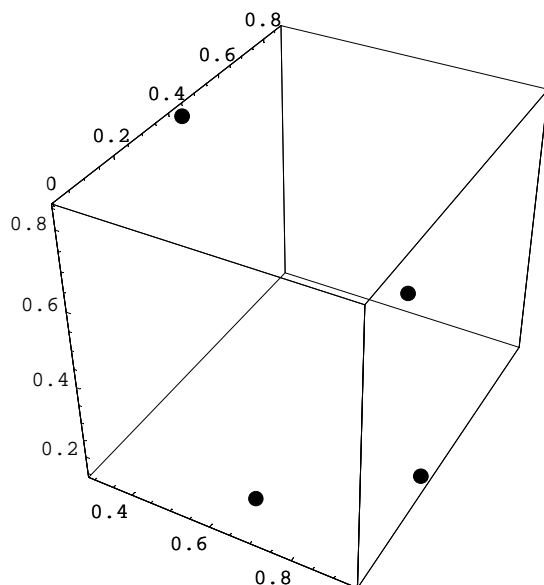
■ Ensuite faire un plot:

```
In[132]:= Show[Graphics3D[{PointSize[0.03], graphicPts}]];
```



Mit Skala: ■ Avec échelle:

```
In[133]:= Show[Graphics3D[{PointSize[0.03], graphicPts}],  
  Axes->Automatic];
```



#### 4.9.2. Umwandlung von graphischen Objekten:

##### ■ Transformation d'objets graphiques:

Mit Hilfe welcher Graphic-Objekte wird ein Plot ausgeführt?

■ A l'aide de quels objets graphiques exécute-t-on un plot?

```
In[134]:=
```

```
??InputForm
```

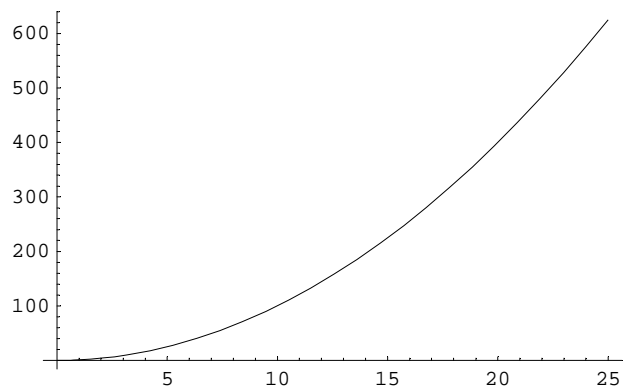
```
InputForm[expr] prints as a version of expr suitable for input to Mathematica. Mehr...
```

```
Attributes[InputForm] = {Protected}
```

```
Options[InputForm] = {NumberMarks->$NumberMarks}
```

```
In[135]:=
```

```
basicPlot = Plot[x^2, {x, 0, 25}];
```



```
In[136]:=
```

```
InputForm[basicPlot]
```

```
Out[136]//InputForm=
```

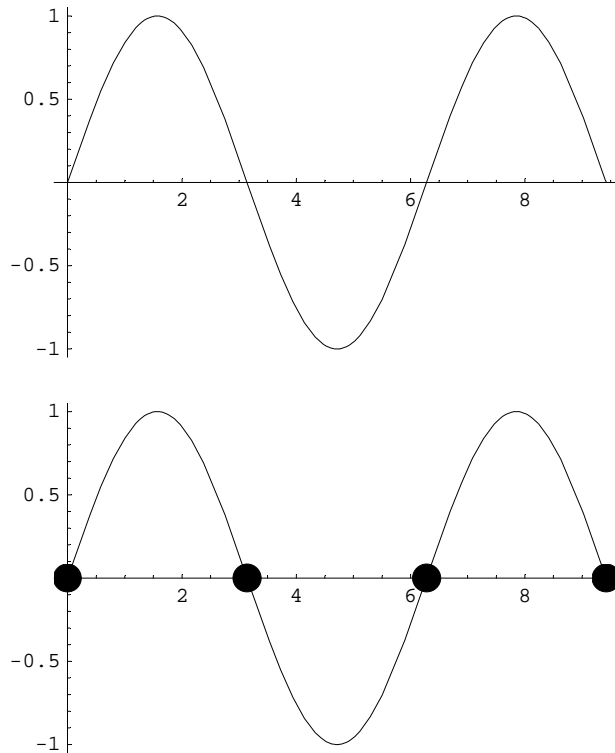
```
Graphics[
  {{Line[{{1.041666666666667*^-6,
    1.0850694444444446*^-12}, {0.030860679008700317,
    0.0009523815088780364}, {0.05893770458747362,
    0.003473653022040309}, {0.0913469976846706,
    0.008344273986003216}, {0.1219837330957231,
    0.01488003113996861}, {0.17647472934441902,
    0.031143330097185947}, {0.23562300449197435,
    0.05551820024582496}, {0.30076105590546476,
    0.0904572127493701}, {0.36981477027355836,
    0.13676296431248475}, {0.4937042690755045,
    0.24374390530337817}, {0.6215800864064731,
    0.3863618038170786}, {0.7413759154314858,
    0.5496382479818737}, {1.014174789322895,
    1.0285505032981384}, {1.2924370222301953,
    1.6703934564312544}, {1.5470261652831419,
    2.393289956070663}, {2.1202199964843422,
    4.495332833492064}, {2.618584361103532,
    6.856984056215994}, {3.1625718515622814,
    10.001860716294077}, {4.229514120963115,
    17.888789699426393}, {5.256668264192063,
    27.632561239763998}, {6.3482009479157915,
    40.29965527511895}, {7.399945505467634,
    54.759193483890634}, {8.41190193684759,
    70.76009419514024}, {9.488236908722328,
    90.02663963604064}, {10.52478375442518,
    110.77107307741218}, {11.521542473956146,
    132.7459409791755}, {12.582679733981891,
    158.3238292879586}, {13.604028867835751,
    185.06960143690847}, {14.689756542184393,
    215.78894726864917}, {15.735696090361147,
    247.6121314482071}, {16.741847512366014,
    280.2894581273161}, {17.812377474865663,
    317.28079130710165}, {18.843119311193426,
    355.06314537587065}, {19.834073021349305,
    393.39045261621635}, {20.889405271999962,
    436.36725261785983}, {21.904949396478735,
    479.82680806229405}, {22.984872061452286,
    528.3043436813299}, {24.025006600253953,
    577.2009421422459}, {24.999998958333332,
    624.9999479166677}}]}], {PlotRange -> Automatic,
  AspectRatio -> GoldenRatio^(-1),
  DisplayFunction -> $DisplayFunction,
  ColorOutput -> Automatic, Axes -> Automatic,
  AxesOrigin -> Automatic, PlotLabel -> None,
  AxesLabel -> None, Ticks -> Automatic,
  GridLines -> None, Prolog -> {}, Epilog -> {},
  AxesStyle -> Automatic, Background -> Automatic,
  DefaultColor -> Automatic, DefaultFont ->
  $DefaultFont, RotateLabel -> True, Frame -> False,
  FrameStyle -> Automatic, FrameTicks -> Automatic,
  FrameLabel -> None, PlotRegion -> Automatic,
  ImageSize -> Automatic, TextStyle -> $TextStyle,
  FormatType -> $FormatType}]
```

Trick: Ausgabe mehrerer Plots auf einmal:

■ Truc: Emission de plusieurs plots à la fois:

In[137]:=

```
Show[
  Plot[Sin[x], {x, 0, 3 Pi}],
  Graphics[{PointSize[0.05],
    Table[Point[{n Pi, 0}], {n, 0, 3}]
  }],
  ];
```

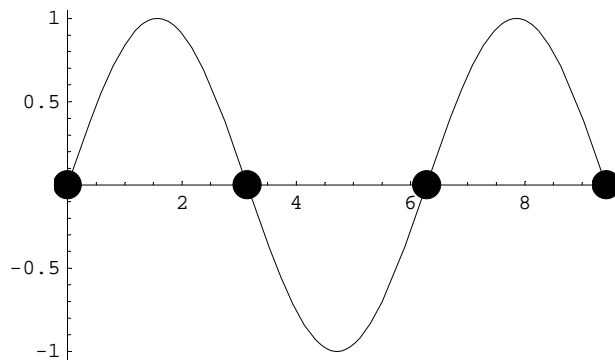


Hier sind 2 Plots gekommen. Damit nur der letzte kommt, muss man mit "DisplayFunction" arbeiten:

■ Voici 2 plots sortis. Pour faire sortir seulement le dernier plot, il faut travailler avec "DisplayFunction":

In[138]:=

```
Show[
  Plot[Sin[x], {x, 0, 3 Pi},
    DisplayFunction->Identity],
  Graphics[{PointSize[0.05],
    Table[Point[{n Pi, 0}], {n, 0, 3}]
  }], DisplayFunction->${DisplayFunction}
  ];
```



## 4.10. Beschriftung von Graphiken (Labels)

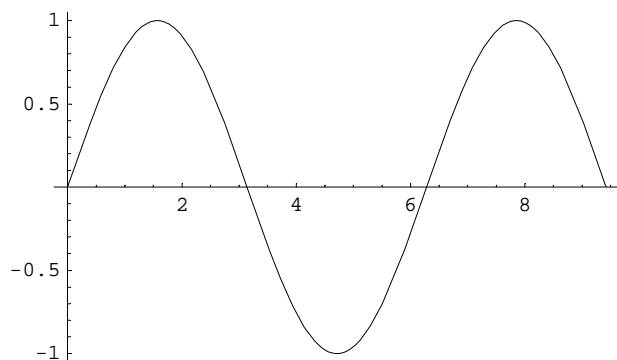
### ■ Doter des graphiques d'inscriptions (Labels)

Beschriftung der x-Achse. Zuerst ohne:

■ Faire des inscriptions sur l'axe x. D'abord sans inscriptions:

`In[139]:=`

```
Plot[Sin[x], {x, 0, 3 Pi}];
```



Dann mit: ■ Ensuite avec inscription:

`In[140]:=`

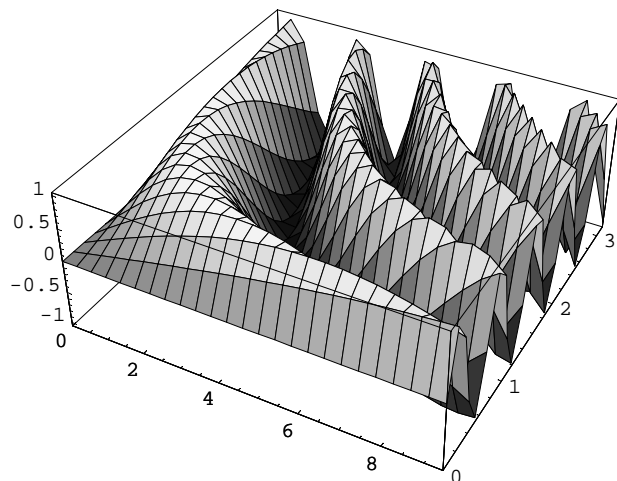
```
??Ticks
```

Ticks is an option for graphics functions that specifies tick marks for axes. Mehr...

```
Attributes[Ticks] = {Protected}
```

`In[141]:=`

```
Plot3D[Sin[x y], {x, 0, 3 Pi}, {y, 0, Pi}];
```



Labels: ■ Labels:

In[142]:=

**??PlotLabel**

PlotLabel is an option for graphics functions that specifies an overall label for a plot.  
Mehr...

Attributes[PlotLabel] = {Protected}

In[143]:=

**??FontForm**

FontForm is an obsolete function, superseded by StyleForm. FontForm[expr, {"font", size}] specifies that expr should be printed in the specified font and size. Mehr...

Attributes[FontForm] = {Protected}

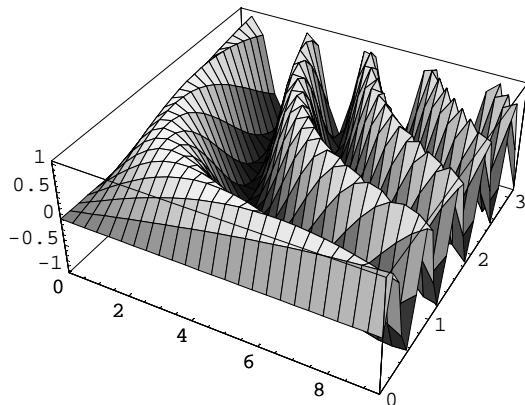
\n: Neue Linie

■ \n: Nouvelle ligne:

In[144]:=

```
Plot3D[Sin[x y], {x, 0, 3 Pi}, {y, 0, Pi},
  PlotLabel -> FontForm["Hallo friend!
  Plot3D[Sin[x y], \n
  {x, 0, 3 Pi}, {y, 0, Pi}]",
  {"Courier-Bold", 15}];
```

```
Hallo friend! Plot3D[Sin[x y],
  {x, 0, 3 Pi}, {y, 0, Pi}]
```



Oder: ■ Ou:

In[145]:=

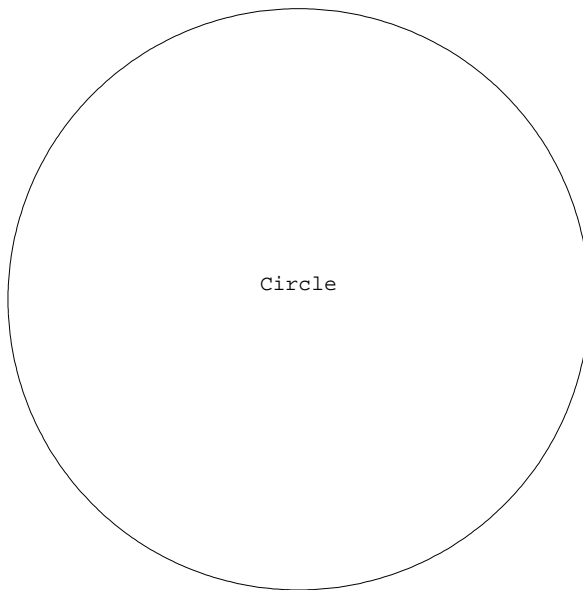
**??Text**

Text[expr, coords] is a graphics primitive that represents text corresponding to the printed form of expr, centered at the point specified by coords. Mehr...

Attributes[Text] = {Protected}

Options[Text] = {Background -> None, FormatType -> Automatic, TextStyle -> Automatic}

```
In[146]:= Show[Graphics[{Circle[{0,0},1],
  Text["Circle",{0,0.05}]}],
  AspectRatio -> Automatic];
```



```
In[147]:=
```

---

## 4.11. Graphik-Pakete

### ■ Paquets de graphiques

Zum Beispiel Rotationskörper:

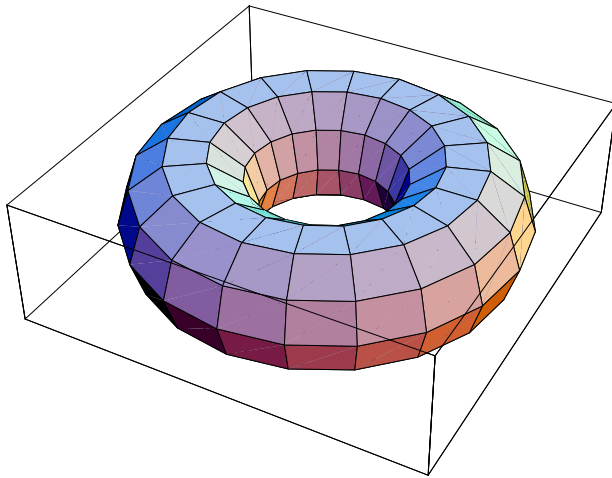
■ Par exemple corps de révolution:

Package laden: ■ Charger Package:

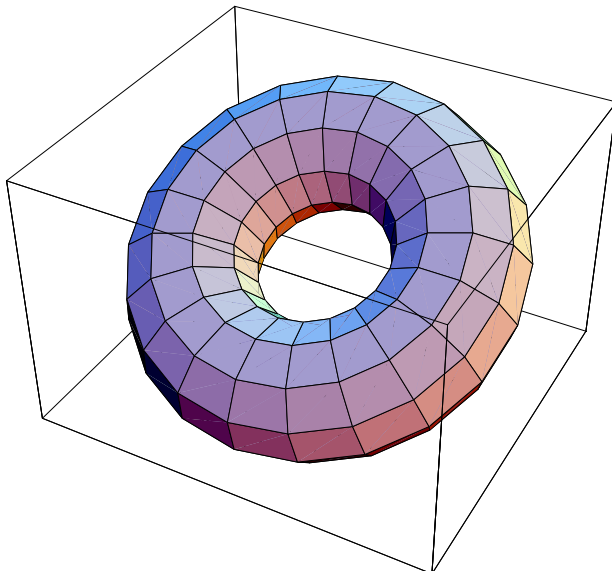
```
In[147]:= << Graphics/Shapes.m
```

Beispiele: ■ Exemples:

```
In[148]:= Show[Graphics3D[Torus[]];
```



```
In[149]:= Show[RotateShape[Graphics3D[Torus[]],  
0, 5 Pi/6, 0];
```



---

## 4.12. Animationen

### ■ Animations

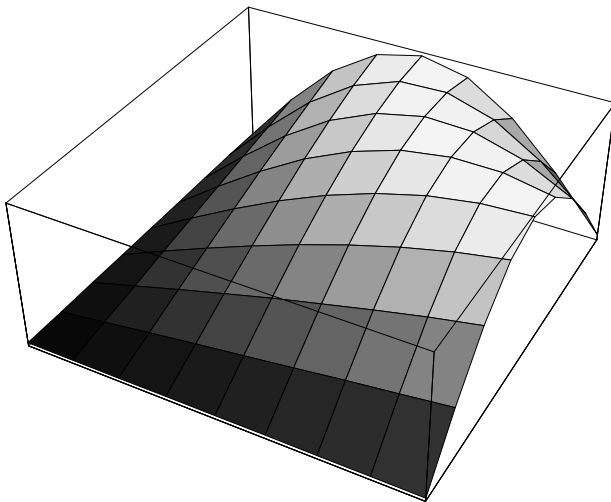
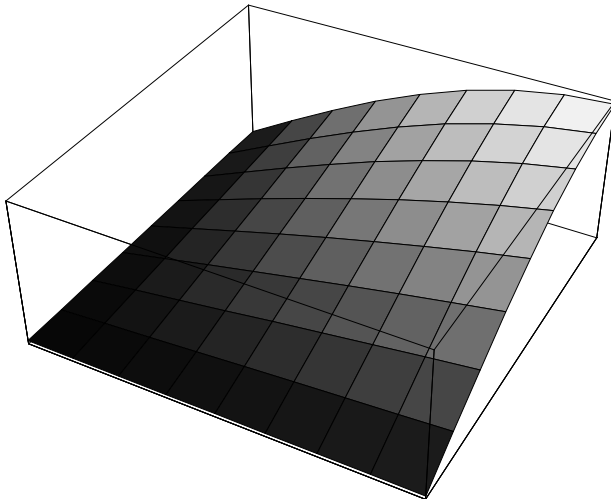
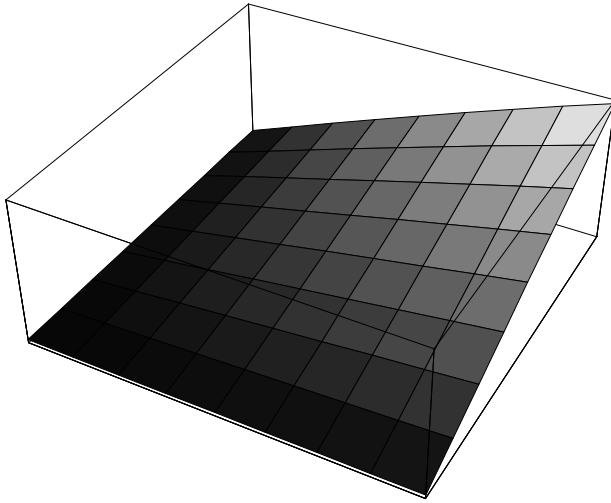
**Auf dem Buchdeckel von Nancy Blachmans Buch:**

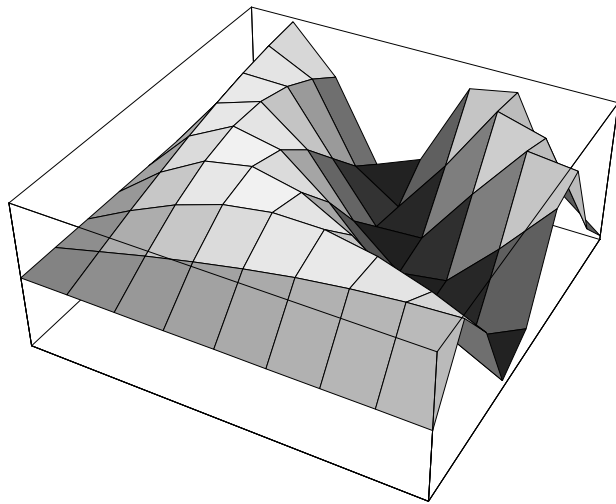
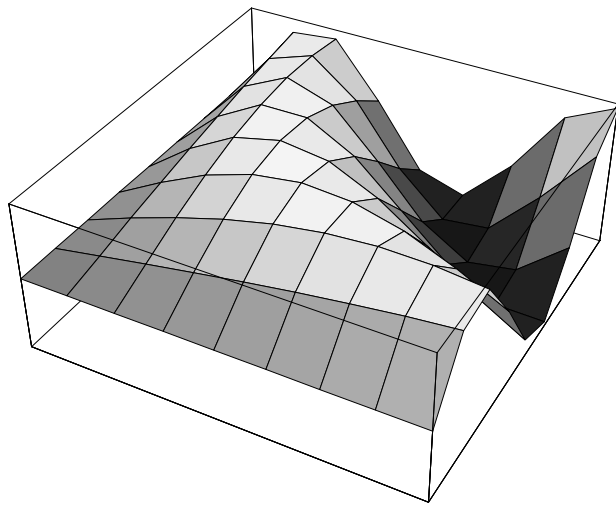
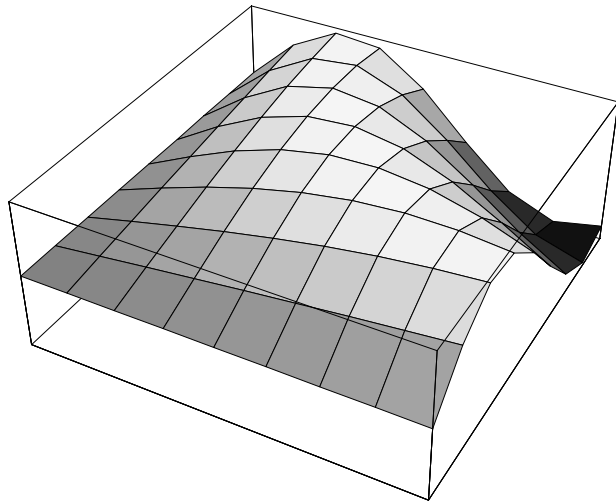
■ **De la couverture du livre de Nancy Blachman:**

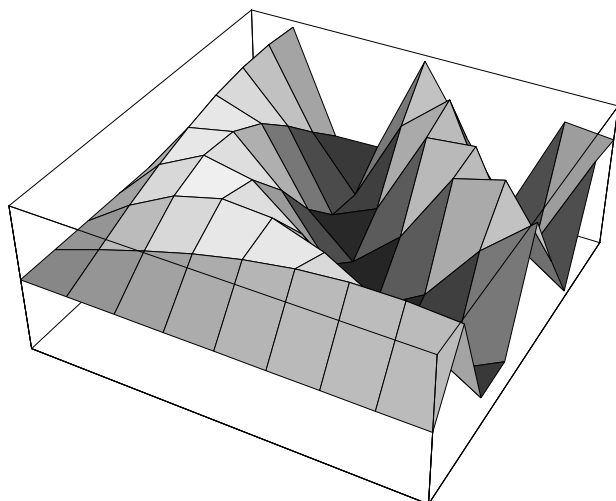
Zuerst Graphiken rechnen (hier nur wenige Bilder zur schnellen und kurzen Demonstration, Bilderzahl bitte im Praktikum selbst erhöhen):

■ D'abord calculer des graphiques (ici quelques images pour une démonstration rapide et brève, augmenter soi-même le nombre d'images lors du travail pratique):

```
In[150]:=  
cover = Table[Plot3D[Sin[x y], {x, 0, t/2},  
  {y, 0, t}, PlotPoints -> 9,  
  Ticks -> None],  
  {t, 1, 6, 3/4}]
```







Out[150]=

```
{ - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -,
  - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics -, - SurfaceGraphics - }
```

### Vorgehen zum Animieren:

#### ■ Procédé d'animation:

New: ==>Go to Cell,Animate Selectet Graphics (Ctrl+Y)

Old: ==>

Jede der eben ausgegebenen Graphiken ist in einer eigenen Zelle untergebracht. Alle diese Graphikzellen wiederum bilden eine gemeinsame Zelle (vgl. von links aus 2. senkrechte Linie am rechten Rand). Diese Zelle ist mit der Maus anzuklicken. (Sie wird dann dunkel markiert.) Nun wählst Du mit der Maus im Menu ("Graph") neu "Cell" und dann "Animate Selected Graphics ....y". Du wirst sehen, was geschieht! Um die Sache anzuhalten, brauchst Du nur mit der Maus irgendwohin ausserhalb der eben gewählten Menuzeile zu "klicken".

■ Chacun des graphiques qui viennent de sortir est placé dans une cellule à lui. Toutes ces cellules de graphiques constituent de nouveau une cellule en commun (compare de gauche la 2e ligne verticale sur le bord de droite). Il faut toucher cette ligne avec la souris. (Elle deviendra foncée.) Maintenant tu choisiss avec la souris dans le menu ("Graph") nouveau "Cell" et ensuite "Animate Selected Graphics ....y". Regarde ce qui se passe! Pour arrêter le processus, il te suffit d'activer la souris n'importe où en dehors du menu que tu viens de choisir.

### Eine verkleinerte Darstellung:

#### ■ Une représentation rapetissée:

In[151]:=

```
??GraphicsArray
```

```
GraphicsArray[{g1, g2, ...}] represents a row of graphics objects. GraphicsArray[{{g11,
  g12, ... }, ...}] represents a two-dimensional array of graphics objects. Mehr...
```

```
Attributes[GraphicsArray] = {Protected, ReadProtected}
```

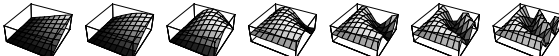
```
Options[GraphicsArray] = {AspectRatio -> Automatic, Axes -> False, AxesLabel -> None,
  AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic,
  ColorOutput -> Automatic, DefaultColor -> Automatic, DefaultFont -> $DefaultFont,
  DisplayFunction -> $DisplayFunction, Epilog -> {}, FormatType -> $FormatType, Frame -> False,
  FrameLabel -> None, FrameStyle -> Automatic, FrameTicks -> None, GraphicsSpacing -> 0.1,
  GridLines -> None, ImageSize -> Automatic, PlotLabel -> None, PlotRange -> Automatic,
  PlotRegion -> Automatic, Prolog -> {}, RotateLabel -> True, TextStyle -> $TextStyle, Ticks -> None}
```

```
In[152]:=
??GraphicsSpacing

GraphicsSpacing is an option for GraphicsArray
  which specifies the spacing between elements in the array. Mehr...

Attributes[GraphicsSpacing] = {Protected}
```

```
In[153]:=
Show[GraphicsArray[cover],
GraphicsSpacing -> 0.2];
```



## 4. 13. PostScript

### ■ PostScript

#### Erzeugen, ablegen, einlesen und betrachten von PostScript-Code

#### ■ Générer, déposer, lire et observer le code PostScript

Einige Erklärungen zu Befehlen:

■ Quelques explications quant aux ordres:

```
In[154]:=
??>>

expr >> filename writes expr to a file. Put[expr1,
  expr2, ... , "filename"] writes a sequence of expressions expr1 to a file. Mehr...

Attributes[Put] = {Protected}
```

```
In[155]:=
??Display

Display[channel, graphics] writes graphics or sound to the specified output channel
  in Mathematica PostScript format. Display[channel, graphics, "format"] writes
  graphics or sound in the specified format. Display[channel, expr, "format"]
  writes boxes, cells or notebook expressions in the specified format. Mehr...

Attributes[Display] = {Protected, ReadProtected}

Options[Display] = {CharacterEncoding->{}, ImageOffset->Automatic,
  ImageResolution->Automatic, ImageRotated->False, ImageSize->Automatic}
```

```
In[156]:=
??<<

<<name reads in a file, evaluating each expression in it, and returning the last one. Mehr...

Attributes[Get] = {Protected}

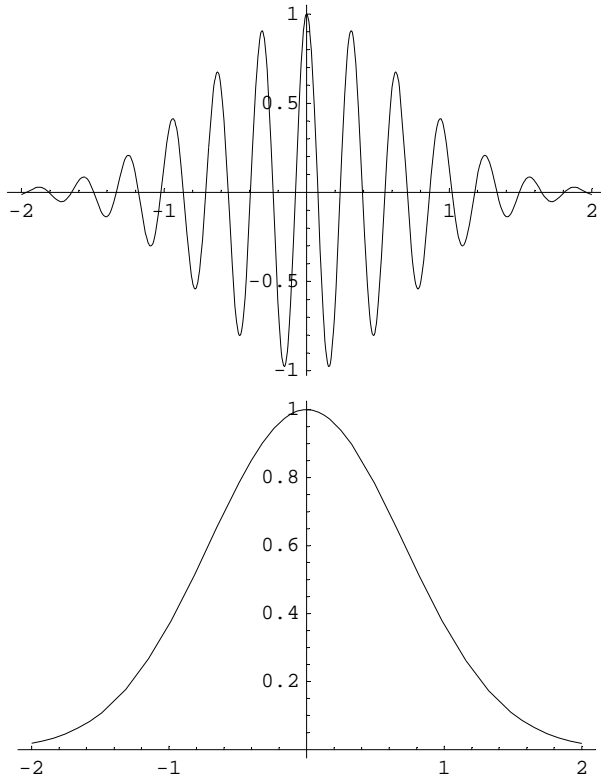
Options[Get] = {CharacterEncoding-> $CharacterEncoding, Path-> $Path}
```

Wir erzeugen eine Graphik:

■ Nous générons un graphique:

```
In[157]:=
```

```
expCos = Plot[Exp[-x^2] Cos[20 x], {x, -2, 2}];  
exp1    = Plot[Exp[-x^2], {x, -2, 2}];
```



Nun schauen wir uns den Inhalt des Files "postscript\_test1.ps" auf dem Schirm an. Du siehst sofort, dass hier offenbar Koordinaten von gerechneten Punkten gespeichert worden sind. Arbeite dazu die folgenden Befehle ab:

■ Maintenant nous regardons le contenu des fichiers (files) "postscript\_test1.ps" sur l'écran. Tu vois tout de suite qu'ici on a évidemment mémorisé les coordonnées des points calculés. Exécute les ordres suivants:

In[159]:=

?expl

Global`expl

expl =

```
Graphics[{{Line[{{-2., 0.0183157}, {-1.92101, 0.0249653}, {-1.83773, 0.0341409}, {-1.75327,
0.0462373}, {-1.66076, 0.0634096}, {-1.57856, 0.082756}, {-1.49063, 0.108395},
{-1.31656, 0.176696}, {-1.14885, 0.267173}, {-0.987513, 0.377124},
{-0.815873, 0.51394}, {-0.6506, 0.654895}, {-0.491692, 0.785244}, {-0.40333, 0.849867},
{-0.322484, 0.901229}, {-0.23804, 0.944912}, {-0.159643, 0.974836},
{-0.121048, 0.985454}, {-0.101284, 0.989794}, {-0.0799043, 0.993636},
{-0.0676623, 0.995432}, {-0.056399, 0.996824}, {-0.0458428, 0.997901},
{-0.0342557, 0.998827}, {-0.0292321, 0.999146}, {-0.0238468, 0.999431},
{-0.0191923, 0.999632}, {-0.0141292, 0.9998}, {-0.0111625, 0.999875},
{-0.00841775, 0.999929}, {-0.00300714, 0.999991}, {0.00190992, 0.999996},
{0.00713368, 0.999949}, {0.01191, 0.999858}, {0.0171347, 0.999706}, {0.0225892, 0.99949},
{0.027664, 0.999235}, {0.0367651, 0.998649}, {0.0464879, 0.997841},
{0.0687153, 0.995289}, {0.0889119, 0.992126}, {0.128404, 0.983648}, {0.165686, 0.972921},
{0.249743, 0.939534}, {0.327245, 0.898446}, {0.492737, 0.784436}, {0.651864, 0.653818},
{0.821291, 0.509401}, {0.984352, 0.379482}, {1.15771, 0.261767}, {1.32471, 0.172932},
{1.48534, 0.110115}, {1.56813, 0.0855182}, {1.65627, 0.0643623}, {1.74637, 0.0473684},
{1.83113, 0.0349775}, {1.91563, 0.0254858}, {1.99327, 0.0188149}, {2., 0.0183157}}]},
{PlotRange -> Automatic, AspectRatio ->  $\frac{1}{\text{GoldenRatio}}$ , DisplayFunction -> $DisplayFunction,
ColorOutput -> Automatic,
Axes -> Automatic,
AxesOrigin -> Automatic, PlotLabel -> None,
AxesLabel -> None, Ticks -> Automatic,
GridLines -> None, Prolog -> {},
Epilog -> {}, AxesStyle -> Automatic,
Background -> Automatic, DefaultColor -> Automatic,
DefaultFont -> $DefaultFont, RotateLabel -> True,
Frame -> False, FrameStyle -> Automatic,
FrameTicks -> Automatic, FrameLabel -> None,
PlotRegion -> Automatic, ImageSize -> Automatic,
TextStyle -> $TextStyle, FormatType -> $FormatType}]
```

Nun schreiben wir mit ">>" den PostScript-Code in ein File namens "postscript\_test.ps" u.s.w. ins eigene Unterverzeichnis: (Hinweis: Der Name des Unterverzeichnisses ist hier "c:\work".)

■ Maintenant nous écrivons avec ">>" le code "PostScript" dans un fichier du nom de "postscript\_test.ps" etc. dans notre propre sous-dossier: (Indication: Le nom du dossier qu'on a ici est "c:\work".)

In[160]:=

```
expCos >> C:\work\MathematicaData\postscri.ps;
expl >> C:\work\MathematicaData\postsc_1.ps;
```

Führe den Befehl aus und kontrolliere rasch im eigenen Verzeichnis, ob das File jetzt da steht! (Vielleicht vergeht etwas Zeit, bis es kommt.) "Display" tut dasselbe. Statt in ein PostScript-File speichern wir nachher den Code mit "Save" in ein *Mathematica*-File, das wir später wieder einlesen und den *Mathematica*-Code verwenden können. (Ausführen und kontrollieren! Probiere im einzulesenden File erst den PostScript-Code ein wenig zu verändern. Hüte Dich dabei vor Werten, die nicht verarbeitet werden können!):

■ Execute l'ordre et contrôle vite dans le propre dossier, si le fichier y est! (Peut-être tu dois attendre un peu qu'il apparaisse.) "Display" fait la même chose. Au lieu de mémoriser le code dans un fichier PostScript, nous le mémorisons après avec "Save" dans un fichier de *Mathematica*, que nous lisons plus tard en utilisant le code de *Mathematica*. (Faire et contrôler! Essaie de changer d'abord un peu le code PostScript dans le fichier à lire. Evite les valeurs qui ne peuvent pas être élaborées!)

```
In[162]:=
  Display["C:\\work\\MathematicaData\\postsc_2.ps",expCos];
  Display["C:\\work\\MathematicaData\\postsc_3.ps",expl];
```

```
In[164]:=
  Save[
    "C:\\work\\MathematicaData\\postscri.ma",expCos];
  Save[
    "C:\\work\\MathematicaData\\postsc_1.ma",expl]

Syntax::stresc : Unknown string escape \M. Mehr...
```

```
In[166]:=
  Save[
    "C:\\work\\MathematicaData\\postscri.nb",expCos];
  Save[
    "C:\\work\\MathematicaData\\postsc_1.nb",expl]
```

Nun löschen wir den PostScript-Code "expCos" u.s.w.:

■ Maintenant nous effaçons le code PostScript "expCos" etc.:

```
In[168]:=
  ??expl

Global`expl

expl =
Graphics[{{Line[{{-2., 0.0183157}, {-1.92101, 0.0249653}, {-1.83773, 0.0341409}, {-1.75327,
  0.0462373}, {-1.66076, 0.0634096}, {-1.57856, 0.082756}, {-1.49063, 0.108395},
  {-1.31656, 0.176696}, {-1.14885, 0.267173}, {-0.987513, 0.377124},
  {-0.815873, 0.51394}, {-0.6506, 0.654895}, {-0.491692, 0.785244}, {-0.40333, 0.849867},
  {-0.322484, 0.901229}, {-0.23804, 0.944912}, {-0.159643, 0.974836},
  {-0.121048, 0.985454}, {-0.101284, 0.989794}, {-0.0799043, 0.993636},
  {-0.0676623, 0.995432}, {-0.056399, 0.996824}, {-0.0458428, 0.997901},
  {-0.0342557, 0.998827}, {-0.0292321, 0.999146}, {-0.0238468, 0.999431},
  {-0.0191923, 0.999632}, {-0.0141292, 0.9998}, {-0.0111625, 0.999875},
  {-0.00841775, 0.999929}, {-0.00300714, 0.999991}, {0.00190992, 0.999996},
  {0.00713368, 0.999949}, {0.01191, 0.999858}, {0.0171347, 0.999706}, {0.0225892, 0.99949},
  {0.027664, 0.999235}, {0.0367651, 0.998649}, {0.0464879, 0.997841},
  {0.0687153, 0.995289}, {0.0889119, 0.992126}, {0.128404, 0.983648}, {0.165686, 0.972921},
  {0.249743, 0.939534}, {0.327245, 0.898446}, {0.492737, 0.784436}, {0.651864, 0.653818},
  {0.821291, 0.509401}, {0.984352, 0.379482}, {1.15771, 0.261767}, {1.32471, 0.172932},
  {1.48534, 0.110115}, {1.56813, 0.0855182}, {1.65627, 0.0643623}, {1.74637, 0.0473684},
  {1.83113, 0.0349775}, {1.91563, 0.0254858}, {1.99327, 0.0188149}, {2., 0.0183157}}]}],
{PlotRange → Automatic, AspectRatio →  $\frac{1}{\text{GoldenRatio}}$ , DisplayFunction → $DisplayFunction,
ColorOutput → Automatic,
Axes → Automatic,
AxesOrigin → Automatic, PlotLabel → None,
AxesLabel → None, Ticks → Automatic,
GridLines → None, Prolog → {}},
Epilog → {}, AxesStyle → Automatic,
Background → Automatic, DefaultColor → Automatic,
DefaultFont → $DefaultFont, RotateLabel → True,
Frame → False, FrameStyle → Automatic,
FrameTicks → Automatic, FrameLabel → None,
PlotRegion → Automatic, ImageSize → Automatic,
TextStyle → $TextStyle, FormatType → $FormatType}]
```

```
In[169]:=
  Remove[expCos, expl]
```

```
In[170]:=
??exp1

Information::notfound : Symbol exp1 not found. Mehr...
```

```
In[171]:=
??expCos

Information::notfound : Symbol expCos not found. Mehr...
```

Das File kann wieder eingelesen werden (Achtung Front-End!):

■ On peut relire le fichier (Attention Front-End!):

```
In[172]:=
??Get

<<name reads in a file, evaluating each expression in it, and returning the last one. Mehr...

Attributes[Get] = {Protected}

Options[Get] = {CharacterEncoding->$CharacterEncoding, Path->$Path}
```

```
In[173]:=
?? !!

n!! gives the double factorial of n. Mehr...

Attributes[Factorial2] = {Listable, NumericFunction, Protected}
```

```
In[174]:=
?? <<

<<name reads in a file, evaluating each expression in it, and returning the last one. Mehr...

Attributes[Get] = {Protected}

Options[Get] = {CharacterEncoding->$CharacterEncoding, Path->$Path}
```

```
In[175]:=
Get["C:\\work\\MathematicaData\\postsc_1.nb",exp1]

Get::notencode : Warning: the file C:\work\MathematicaData\postsc_1.nb is not encoded. Mehr...
```

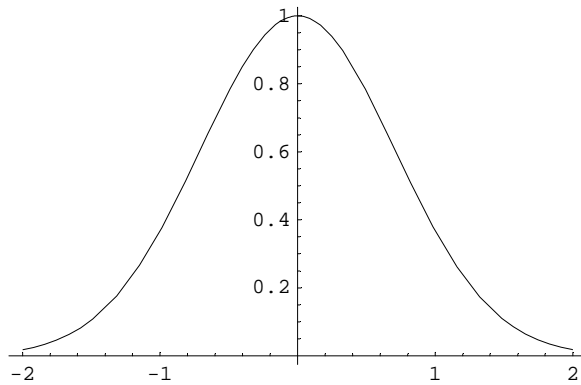
```
Out[175]=
StandardForm
```

```
In[176]:=
Get["C:\\work\\MathematicaData\\postsc_1.nb"]; exp1
```

```
Out[176]=
- Graphics -
```

In[177]:=

```
Get["C:\work\MathematicaData\postsc_1.nb"]; Show[expl]
```



Out[177]=

```
- Graphics -
```

In[178]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.nb", expl]
```

```
Get::notencode : Warning: the file C:\work\MathematicaData\postsc_1.nb is not encoded. Mehr...
```

Out[178]=

```
StandardForm
```

In[179]:=

```
?expl
```

```
Global`expl
```

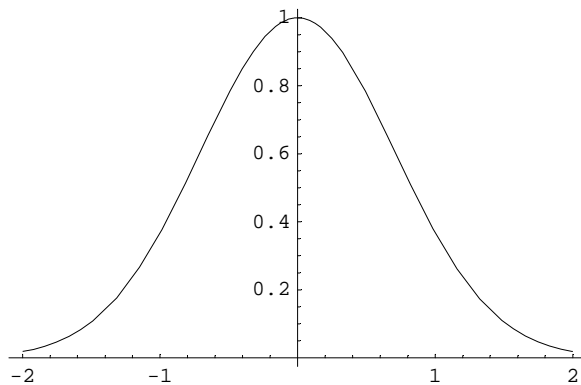
```
expl =
```

```
Graphics[{{Line[{{-2., 0.0183157}, {-1.92101, 0.0249653}, {-1.83773, 0.0341409}, {-1.75327,
0.0462373}, {-1.66076, 0.0634096}, {-1.57856, 0.082756}, {-1.49063, 0.108395},
{-1.31656, 0.176696}, {-1.14885, 0.267173}, {-0.987513, 0.377124},
{-0.815873, 0.51394}, {-0.6506, 0.654895}, {-0.491692, 0.785244}, {-0.40333, 0.849867},
{-0.322484, 0.901229}, {-0.23804, 0.944912}, {-0.159643, 0.974836},
{-0.121048, 0.985454}, {-0.101284, 0.989794}, {-0.0799043, 0.993636},
{-0.0676623, 0.995432}, {-0.056399, 0.996824}, {-0.0458428, 0.997901},
{-0.0342557, 0.998827}, {-0.0292321, 0.999146}, {-0.0238468, 0.999431},
{-0.0191923, 0.999632}, {-0.0141292, 0.9998}, {-0.0111625, 0.999875},
{-0.00841775, 0.999929}, {-0.00300714, 0.999991}, {0.00190992, 0.999996},
{0.00713368, 0.999949}, {0.01191, 0.999858}, {0.0171347, 0.999706}, {0.0225892, 0.99949},
{0.027664, 0.999235}, {0.0367651, 0.998649}, {0.0464879, 0.997841},
{0.0687153, 0.995289}, {0.0889119, 0.992126}, {0.128404, 0.983648}, {0.165686, 0.972921},
{0.249743, 0.939534}, {0.327245, 0.898446}, {0.492737, 0.784436}, {0.651864, 0.653818},
{0.821291, 0.509401}, {0.984352, 0.379482}, {1.15771, 0.261767}, {1.32471, 0.172932},
{1.48534, 0.110115}, {1.56813, 0.0855182}, {1.65627, 0.0643623}, {1.74637, 0.0473684},
{1.83113, 0.0349775}, {1.91563, 0.0254858}, {1.99327, 0.0188149}, {2., 0.0183157}}]}],
{PlotRange -> Automatic, AspectRatio ->  $\frac{1}{\text{GoldenRatio}}$ , DisplayFunction -> $DisplayFunction,
ColorOutput -> Automatic,
Axes -> Automatic,
AxesOrigin -> Automatic, PlotLabel -> None,
AxesLabel -> None, Ticks -> Automatic,
GridLines -> None, Prolog -> {},
Epilog -> {}, AxesStyle -> Automatic,
Background -> Automatic, DefaultColor -> Automatic,
DefaultFont -> $DefaultFont, RotateLabel -> True,
Frame -> False, FrameStyle -> Automatic,
FrameTicks -> Automatic, FrameLabel -> None,
PlotRegion -> Automatic, ImageSize -> Automatic,
TextStyle -> $TextStyle, FormatType -> $FormatType}]
```

In[180]:=

```
Show[Graphics[expCos]];
Show[Graphics[exp1]];
```

Graphics::gprim : expCos was encountered where a Graphics primitive or directive was expected. Mehr...



Der Befehl ist jetzt wieder da! Nun schauen wir uns den Inhalt des Files "postscri.ps" auf dem Schirm an.

■ Voici de nouveau l'ordre! Maintenant nous regardons le contenu du fichier "postscri.ps" sur l'écran.

In[182]:=

```
!!"C:\\work\\MathematicaData\\postscri.ps"
```

General::noopen : Cannot open "C:\\work\\MathematicaData\\postscri.ps". Mehr...

In[183]:=

```
!!C:\\work\\MathematicaData\\postscri.ps
```

```
Graphics[{{Line[{{-1.999998333333333, -0.0122153593511319},
  {-1.9623003192812842, 0.0005082587300581328}, {-1.9210073169479194,
  0.018752002927052425}, {-1.910299579465502, 0.022740269212148626},
  {-1.9001774790199295, 0.025790484842825256}, {-1.8945829997783903,
  0.027103353077953987}, {-1.8916352820184055, 0.02767476304778044},
  {-1.8885103672286825, 0.028184000386560142}, {-1.8829909524966886,
  0.028827453836099467}, {-1.8778952782513714, 0.029115238245558964},
  {-1.8731413317543735, 0.029105227315202174}, {-1.8680274614917154,
  0.028783053185285283}, {-1.863199958792915, 0.028174933726430717},
  {-1.8587831546785645, 0.027355730435410536}, {-1.8536232573178666,
  0.026078553418176047}, {-1.8479909934251308, 0.024292164406973862},
  {-1.8377320337083367, 0.020012423042739088}, {-1.8257366220651903,
  0.013444784972129038}, {-1.8144362944250474, 0.005937356649683411},
  {-1.7933141736895866, -0.010391844332965274}, {-1.768652621938747,
  -0.030023715053337733}, {-1.7578358910677812, -0.03757544820569005},
  {-1.745962817885428, -0.04436555940559883}, {-1.7409954301270045,
  -0.04661226368848812}, {-1.7356703543983865, -0.04856988304671834},
  {-1.7306441655063993, -0.04995350510306155}, {-1.726061421963292,
  -0.05079554189295374}, {-1.7208899413754872, -0.05123896481969352},
  {-1.7179679228759093, -0.05124293728587858}, {-1.7152644782688042,
  -0.05108393336125241}, {-1.7103050520507237, -0.0503777772427448},
```

{-1.705036285194253, -0.04903000884182273}, {-1.6998130954536856, -0.04707998343149028}, {-1.6941824661510267, -0.04429399736242225}, {-1.6839927695262276, -0.03748701340302931}, {-1.6718751692383962, -0.026621373541510396}, {-1.660764800562505, -0.01436785949148811}, {-1.6223373700539767, 0.03698336903382436}, {-1.6122046687997946, 0.05026582756968046}, {-1.6012204218638049, 0.0631825070687631}, {-1.5915368555668616, 0.07268572843936418}, {-1.58624016644988, 0.07694697859595899}, {-1.5812846856736067, 0.08024862424865138}, {-1.5754383042802425, 0.0832155003092454}, {-1.572337956846547, 0.08435492782676386}, {-1.5690072708401817, 0.08522809918894149}, {-1.5659728443536363, 0.08569752092152322}, {-1.5630921326197909, 0.08584815274454088}, {-1.5576873525951036, 0.08533590009119145}, {-1.5523642925764947, 0.08379516017296415}, {-1.5474289001941306, 0.08143362486126446}, {-1.5420125470492216, 0.07780448278522988}, {-1.536246079580289, 0.07275700026813073}, {-1.5249155505844803, 0.059402686355185516}, {-1.5142556242299647, 0.04300384325934713}, {-1.4943432743542384, 0.004472710594773914}, {-1.4532279345574657, -0.08515390290044901}, {-1.4432335331173123, -0.10348325385511208}, {-1.432533564692643, -0.11946821817256431}, {-1.42638894081693, -0.12655827281836432}, {-1.4207938199321637, -0.13150559927652575}, {-1.41809797650952, -0.13334094680801603}, {-1.4151693384029775, -0.13491283836164045}, {-1.4098481479164913, -0.13660415416094238}, {-1.4050416145772238, -0.13679532253598312}, {-1.4005351214442552, -0.1357881187725129}, {-1.3955913202634447, -0.13333739136336295}, {-1.3902688897887092, -0.12910990712973433}, {-1.3849829640264526, -0.1232818098314748}, {-1.3794251016434265, -0.11542481340098515}, {-1.36949576036685, -0.09713667609456161}, {-1.3516769177229944, -0.05213560711740385}, {-1.3322109850418382, 0.010049649308713196}, {-1.311814606942803, 0.0805873886472376}, {-1.301081564243093, 0.11597117145006212}, {-1.2897332454943118, 0.14947293224243818}, {-1.2797977572941663, 0.17390652324509817}, {-1.2745323012358132, 0.1845386245184353}, {-1.2688830972260063, 0.19390980702104588}, {-1.2636935060450525, 0.20050558155842912}, {-1.2589579198010443, 0.20473215653759175}, {-1.2541975830843266, 0.20717275659823553}, {-1.249684335929672, 0.20775197540724452}, {-1.247041828138711, 0.20728903288692044}, {-1.2441226325567252, 0.20607895510875865}, {-1.2410738117547113, 0.20402551530318114}, {-1.2382092114607313, 0.20135719496762308}, {-1.2330981005800692, 0.19481588054246313}, {-1.2275838049165742, 0.18521572373110567}, {-1.2214743617585684, 0.17155776588451163}, {-1.2150369184940277, 0.1538563691722545}, {-1.203523571823205, 0.11438509646840048}, {-1.1616357322503013, -0.0838667012692308}, {-1.1410574771688768, -0.18355519085192498}, {-1.1309454699394663, -0.2252446687030288}, {-1.1216144865495554, -0.2570070238434482}, {-1.116498155471344, -0.2711457519497164}, {-1.1108655908628202, -0.28370556740513797}, {-1.1081370509617479, -0.2885858899753463}, {-1.1052034083602063, -0.29291770319564026}, {-1.1021050224055187, -0.2964329900999418}, {-1.0992038714781798, -0.29871236701229176}, {-1.0937451922278991, -0.3002768983062175}, {-1.0885588150162093, -0.29839061424267377}, {-1.0839033652069439, -0.2938521069914662}, {-1.0788242866321036, -0.28581302032434724}, {-1.0735924733113682, -0.27417965132220024}, {-1.068666299969677, -0.26016454226758307}, {-1.0594607465548975, -0.22631156310060757}, {-1.0388287578280821, -0.11852835113159345}, {-1.0017158165421325, 0.1380386876955374}, {-0.9804818490676098, 0.27714025981739737}, {-0.9701152472705907, 0.3320839289435045}, {-0.9606680178167547, 0.37136370846023714}, {-0.9555429871463601, 0.3876711007347637}, {-0.9506903685808147, 0.39957150651606144}, {-0.94556841761877, 0.4081946077693798}, {-0.9426751414885124, 0.411212892719486}, {-0.939996360794677, 0.41278513550084933}, {-0.934770753521844, 0.4124157216304582}, {-0.9317946152243468, 0.41014558532629825}, {-0.929001985645631, 0.4066458604324708}, {-0.9260688269140263, 0.40153938048657495}, {-0.9233808299909719, 0.39557434627057825}, {-0.9173242395579699, 0.3776634222547061}, {-0.9127976387820719, 0.3602999772862939}, {-0.9079642254182297, 0.3381144016387083}, {-0.8992602803223766, 0.28915729392454115}, {-0.8795104906864795, 0.1413842470955593}, {-0.8397190375423287, -0.23005651682605105}, {-0.8187971422559015, -0.4015572594838865}, {-0.8082518687084622, -0.4669266707753546}, {-0.8022604864947778, -0.49579074972261117}, {-0.7966578833395991, -0.5167287009438404}, {-0.7935177520634596, -0.525757956681514}, {-0.7905381888494231, -0.5324618998910939}, {-0.7878635232886223, -0.5368991395999997}, {-0.7849493165407382, -0.5400002249881906}, {-0.7817880476133191, -0.541289292474234}, {-0.7784241259699886, -0.5402608150821243}, {-0.7752499569500357,

-0.5370024066246588}, {-0.7723771954811092, -0.5321307725565154},  
{-0.7668323840651169, -0.5175679461624914}, {-0.7637114688582315,  
-0.5064022525594425}, {-0.7608453028627378, -0.4942884130331306},  
{-0.750001386238209, -0.43286649730882254}, {-0.7398238797968588,  
-0.35436495342595553}, {-0.7303395497421928, -0.2654707763053889},  
{-0.7087968591039506, -0.023453258580960444}, {-0.6862069434280346,  
0.2506456308828193}, {-0.6653448287717248, 0.47409810939227504},  
{-0.6550941824327547, 0.5599189734032894}, {-0.6498581570937355,  
0.5956349265395152}, {-0.6442239577120495, 0.6271947855867068},  
{-0.6412828970965597, 0.6406690475025774}, {-0.6385912090240337,  
0.651124447712424}, {-0.633312731602339, 0.6662557045388081},  
{-0.6303669227819514, 0.6715264501165197}, {-0.6271504370779537,  
0.6746302040365503}, {-0.6241122714698554, 0.6749890352056996},  
{-0.6212907785054942, 0.673065392685534}, {-0.615848103126219,  
0.6631864529584339}, {-0.610719214050287, 0.6464572445824115},  
{-0.6061140961990733, 0.6253747705571909}, {-0.6011310493901006,  
0.5962432752291937}, {-0.58993621358246, 0.5080588617703872},  
{-0.5796332258514509, 0.40179883304703257}, {-0.5419911185092119,  
-0.11563789469737708}, {-0.5212657131179157, -0.41140376620744473},  
{-0.5016259570844893, -0.6383082637499763}, {-0.49139111028154175,  
-0.7225368945842895}, {-0.4856072258904732, -0.7575352067107914},  
{-0.4802543250749207, -0.78114955618428}, {-0.4757632963228615,  
-0.7941758653567549}, {-0.47080452614718676, -0.8011599707142121},  
{-0.46562566801066907, -0.800017069509857}, {-0.460762181661922,  
-0.7910321667400466}, {-0.45609740432417917, -0.7752294176464226},  
{-0.4511846581763733, -0.7510703522206259}, {-0.4469901738226814,  
-0.7244650488260634}, {-0.4424004989015632, -0.6892291015728184},  
{-0.43214488080401514, -0.5887125083849086}, {-0.4226003398105357,  
-0.4709297597147562}, {-0.37797369624595195, 0.25162584454478143},  
{-0.35605021441036094, 0.5894204860628905}, {-0.34642322744337106,  
0.7085825762585028}, {-0.34131725061385637, 0.7619387936024965},  
{-0.3358302221056187, 0.8107423480186627}, {-0.33042975244711636,  
0.8495136641902389}, {-0.3254936995516747, 0.8764609305371227},  
{-0.32053192780411477, 0.8950417017536064}, {-0.3158200401288461,  
0.9045714043821929}, {-0.3105605721446212, 0.9057062735594216},  
{-0.3056826236909325, 0.8977335870214513}, {-0.30032936887031186,  
0.8790188868261573}, {-0.2973379125422991, 0.8640692854201357},  
{-0.29462830641223864, 0.847791590281556}, {-0.2835658740663681,  
0.7553319633410993}, {-0.27733670569774405, 0.6860096772305007},  
{-0.2716113406541768, 0.6123778152682854}, {-0.24994991234572664,  
0.2655800740728669}, {-0.21225773815848598, -0.43057614557528495},  
{-0.2014045185385447, -0.607001214670649}, {-0.1911333494049853,  
-0.7490273389929445}, {-0.18191240706908424, -0.8505632497565038},  
{-0.1767416129962974, -0.8952652342589389}, {-0.1718920901267761,  
-0.9285918556554922}, {-0.16697957316058187, -0.9535023649777139},  
{-0.161562565091676, -0.9703221049620906}, {-0.15628418144238038,  
-0.9757476325207776}, {-0.15331730465130441, -0.9740039752842746},  
{-0.15059977644037958, -0.9693770667343177}, {-0.1460165797564408,  
-0.9550405828469034}, {-0.1410081756896768, -0.9301062215571163},  
{-0.13577962138178978, -0.893991278427419}, {-0.13091999659907888,  
-0.8515076827970871}, {-0.12123049018585531, -0.7427944269588428},  
{-0.11243516664911235, -0.6192856959607796}, {-0.09270736290683174,  
-0.2771819714654778}, {-0.07074128821999481, 0.15456352776450047},  
{-0.047508779695364454, 0.5802291446050808}, {-0.03794148475458846,  
0.7245979117086273}, {-0.027673730365563417, 0.8500519974179697},  
{-0.021783670915650262, 0.9061559525105312}, {-0.01640722304970641,  
0.9463871901825228}, {-0.01100269176883796, 0.9757675666255191},  
{-0.005877094856298342, 0.9930655997505854}, {-0.0012603543457706003,  
0.9996807302191512}, {0.003841377090691665, 0.9970355032139022},  
{0.00916971378005925, 0.9831476806414581}, {0.01417765010626921,  
0.9598745223762447}, {0.018835778805047276, 0.9295480348256523},  
{0.02311152799549632, 0.8945820542534499}, {0.03275314394312561,  
0.792159363745938}, {0.05323437432389867, 0.4834057057361746},  
{0.07482189650603684, 0.07387515592061891}, {0.09497769065648072,  
-0.3199678960821519}, {0.1165347071741706, -0.6795558291669739},  
{0.12290188181040415, -0.7637091472760054}, {0.1289196612394384,  
-0.8316141273358323}, {0.13477808256826967, -0.8859263593475684},  
{0.14018569726502234, -0.9251000595565696}, {0.14531416564759841,  
-0.9521232602335068}, {0.15074987886152633, -0.9697081023973886},  
{0.15620612109614104, -0.9757460076225583}, {0.1593087430762231,  
-0.9739713489533526}, {0.16221476370581453, -0.9688971227041098},  
{0.16745001383881453, -0.9515105805200534}, {0.1703767756527555,  
-0.9372399699444143}, {0.17302994423284257, -0.9215400634230125},

{0.18306876655623647, -0.8393213096217108}, {0.18869925168730767, -0.7783995770475729}, {0.194788749285674, -0.7016958861226725}, {0.2058274871176703, -0.5379247908600145}, {0.2472775023612705}, {0.21734939392563712}, {0.2669876956883489}, {0.5466263082463674}, {0.27775578978345566}, {0.6910376340383966}, {0.28780683415627745}, {0.7955882301276718}, {0.29221298913057425}, {0.8311225084438845}, {0.2963862394019538}, {0.858646541028629}, {0.30119657032466773}, {0.8827537327474217}, {0.30558165684475946}, {0.8974768328574726}, {0.3104357848654642}, {0.905610656162231}, {0.31492594941144675}, {0.9054747211033035}, {0.319854075345852}, {0.8969036163134283}, {0.3250885838720468}, {0.8783015513818669}, {0.32986816478766834}, {0.8529956085057786}, {0.33509542949091004}, {0.8165707134217922}, {0.34562891803050866}, {0.717360860055138}, {0.3644582901051175}, {0.4686814149708279}, {0.40689090862089167}, {-0.2373117913483952}, {0.426125604486471}, {-0.5169094904177725}, {0.43689074850830234}, {-0.6388263696340332}, {0.44197946198477883}, {-0.6856842565689083}, {0.44667041262385826}, {-0.7222151789885232}, {0.4510207771170432}, {-0.7501332725563638}, {0.4558052679116797}, {-0.7740071803631008}, {0.4606148703666571}, {-0.7906404333586745}, {0.4657059069711432}, {-0.8001009858284969}, {0.4701818412566341}, {-0.8014806461151873}, {0.47498938854669887}, {-0.7957813656725395}, {0.4795274484879301}, {-0.7836835448282652}, {0.48367316714241243}, {-0.7670632421985295}, {0.48910652724614345}, {-0.7375047588806098}, {0.49421732943212265}, {-0.7020199626620278}, {0.5037632278937437}, {-0.6174246056732089}, {0.5251478059134861}, {-0.35894759636214624}, {0.5635945404101352}, {0.19857175803531235}, {0.5742159503756941}, {0.33764078266505365}, {0.5854201872696259}, {0.4642184562076209}, {0.59498618095222}, {0.5515984138888202}, {0.6003741082859045}, {0.591255829543805}, {0.60547052133523}, {0.6219780313442358}, {0.6083075285152632}, {0.6361250407609035}, {0.6113889278134459}, {0.6490479456023531}, {0.6167770249727986}, {0.6654477396485784}, {0.6198296973349456}, {0.6712117051612357}, {0.623067888318758}, {0.6745310254743232}, {0.6258324575645496}, {0.6750943051423131}, {0.6288793992299165}, {0.6733081258630773}, {0.634303848090634}, {0.663966651997967}, {0.6373520249762497}, {0.6553223823800816}, {0.6401426151420501}, {0.6453200158057888}, {0.645553737508074}, {0.6204158430707607}, {0.650700713891915}, {0.5902905186685075}, {0.6939669926869411}, {0.15752673366231526}, {0.7152020721601373}, {-0.09959206992178356}, {0.7347515180442568}, {-0.308535470979283}, {0.7439550305288842}, {-0.388503320333675}, {0.7525526748937331}, {-0.44947412460545044}, {0.761520709343437}, {-0.4973020699793587}, {0.7662730077468173}, {-0.5157230754805726}, {0.7713706752869288}, {-0.5299915837104358}, {0.7764655824982817}, {-0.5385135508046426}, {0.7820111997625422}, {-0.5412697136677055}, {0.7872487221463473}, {-0.5377047719381315}, {0.7920974807320128}, {-0.5291826799478381}, {0.7964529521305389}, {-0.5173784076900075}, {0.8012635440655589}, {-0.49995684200916335}, {0.8110073898497123}, {-0.45154940718504377}, {0.8216866556673058}, {-0.38077890138613835}, {0.8314843412770514}, {-0.30276272782875757}, {0.8535723501576126}, {-0.09933208457858644}, {0.8742799127520443}, {0.09562488950229762}, {0.8935577024893068}, {0.25127274525933935}, {0.9034528690271358}, {0.3141373454867022}, {0.9140306207806459}, {0.3653613330128868}, {0.9200662975061342}, {0.38653503814918144}, {0.9230318897571291}, {0.39471010144585883}, {0.9257679017110405}, {0.40093262096077703}, {0.9308868297973646}, {0.40915363270998506}, {0.9364394027231373}, {0.41303250288556703}, {0.9418327858488762}, {0.4118351189538449}, {0.9468818702625253}, {0.4063788296314459}, {0.9524211948594483}, {0.3957343604592793}, {0.9555317848759625}, {0.3877026178486205}, {0.9583757626343072}, {0.37911768876508517}, {0.9702935393392168}, {0.33123670459888427}, {0.9816146322866746}, {0.2704990999329876}, {1.0219337574845162}, {-0.006447852680366173}, {1.0440580327404785}, {-0.1494976259339504}, {1.0549091834196738}, {-0.2060834353790531}, {1.0648858169822157}, {-0.2474436640448388}, {1.0743602626221473}, {-0.27609822989107596}, {1.0791303978563087}, {-0.2863886585377893}, {1.0843627332104027}, {-0.2944202724381302}, {1.0896214981734995}, {-0.29904808873811173}, {1.0925692255875215}, {-0.30013993168987985}, {1.0952550142087834}, {-0.30020482778189655}, {1.1004751463501758}, {-0.2978354880883429}, {1.1054168621295604}, {-0.2926351096973051}, {1.1098516796268472}, {-0.28561328923965135}, {1.1145598420490528}, {-0.2758356038124899}, {1.1230130242444196}, {-0.25271804093753775}, {1.142140568536079}, {-0.17872230408925346}, {1.1815509438027387}, {0.01708703617472436}, {1.2022395560575547}, {0.10941576698635769}, {1.2126386342258384}, {0.1464307316264005}, {1.218596390232883}, {0.16405381673229888}, {1.2240060265504031},

```

0.17759683511463492}, {1.2288279588669109, 0.18761035167194878},
{1.2340514887294487, 0.19620888071038206}, {1.2395697767980838,
0.20271393807973756}, {1.2447757706178375, 0.20641373354368245},
{1.2496593073706657, 0.20775038993425438}, {1.2548260055427813,
0.20695695951905246}, {1.2592371268498836, 0.20453200348774384},
{1.26408599477186, 0.20007719367288554}, {1.268870350843575,
0.19392842885484418}, {1.2732388716663274, 0.18687694203521504},
{1.2830649217771246, 0.1664640675381528}, {1.3008089017854554,
0.11683131393049553}, {1.3395619533221224, -0.014559309921745341},
{1.3497807250584448, -0.04656505959061242}, {1.360812454108068,
-0.07699189854131996}, {1.3705376522004298, -0.09930216737751288},
{1.3807913421137497, -0.11751844389690592}, {1.3866981275031691,
-0.1253500384725379}, {1.389824237087754, -0.12868205488181444},
{1.393168712185386, -0.13161783385082187}, {1.3962490792977973,
-0.1337452708365822}, {1.399052023099811, -0.1352016057928178},
{1.4045335760246098, -0.13673969740956485}, {1.4098285490083289,
-0.13660754156569366}, {1.4128172144948126, -0.13584859261362373},
{1.4155767598509525, -0.13472094801234416}, {1.4206443460942033,
-0.1316169197489221}, {1.4260171995048834, -0.12693301271192878},
{1.436311228356535, -0.1143182202119677}, {1.4459559617815967,
-0.09877639029056667}, {1.4679201985139687, -0.054223359972494584},
{1.4887579343823243, -0.00762072873950793}, {1.5115059793294574,
0.0382345442533714}, {1.5219052911983821, 0.05512766037480968},
{1.5329927229796834, 0.06937991404491133}, {1.5383799169472308,
0.07476578756256917}, {1.5434555481172565, 0.07887721208535046},
{1.5480128760885281, 0.08176009227545963}, {1.5529432631496285,
0.08401317903186631}, {1.5577999319927756, 0.08535730267782553},
{1.562334603946491, 0.08583929036751099}, {1.5673100084113765,
0.08552953552719611}, {1.5726443116134108, 0.08425618533669176},
{1.5753219956329856, 0.08326380557115205}, {1.5782350884201926,
0.08192639160190762}, {1.5835319886303225, 0.07883722949145197},
{1.5934602592130633, 0.07096490215862193}, {1.6128244376616974,
0.04948420103577662}, {1.6307422583466153, 0.02543404632466773},
{1.6491132192575328, 0.00029382516263982446}, {1.6684570224553763,
-0.023063944014359752}, {1.6785864950413831, -0.032992068553395676},
{1.6896682018240128, -0.04155430222370065}, {1.6945391991512138,
-0.044491418002653244}, {1.6997435873078401, -0.047049906538513514},
{1.7046552079069046, -0.04890844329652339}, {1.7091185902618236,
-0.05012830831614879}, {1.711901482632756, -0.05066418084712502},
{1.7148481318213324, -0.05104538406562282}, {1.7201314630241844,
-0.051257364803291655}, {1.7230685323472925, -0.051119227645411644},
{1.725816724432411, -0.05082882337742177}, {1.7288318026150509,
-0.050335628120028626}, {1.7320712690896647, -0.049608398055116146},
{1.7378798445858235, -0.047816864809224556}, {1.743394830049025,
-0.045575434520178475}, {1.7537426239565226, -0.04012069650849713},
{1.7737875026298968, -0.026116835698228463}, {1.7956411814708972,
-0.00850661872577543}, {1.8180209034331518, 0.008440018725300728},
{1.8289944674381577, 0.01538353194694639}, {1.8390785712365771,
0.020647516279422223}, {1.8482787300585155, 0.024393268036740877},
{1.8529669537056004, 0.02589139337639477}, {1.8579867443579592,
0.02718112978745319}, {1.8630295576621545, 0.028148004455478503},
{1.8677318350704086, 0.02875436483964524}, {1.8706049904237505,
0.028986380194225708}, {1.873323538248825, 0.0291106936040121},
{1.8784157905220895, 0.02909979399104083}, {1.8834819593167174,
0.028783911606270977}, {1.8863644484863291, 0.028473886915831758},
{1.889067323789397, 0.0281007099146851}, {1.8945097503439552,
0.027118590924287478}, {1.9003696902891154, 0.025740332252873793},
{1.9109593009628465, 0.022514858084424577}, {1.920657051873991,
0.018892401701557857}, {1.9665514341784773, -0.0012774939524654253},
{1.999998333333333, -0.0122153593511319}}}],
{PlotRange -> Automatic, AspectRatio -> GoldenRatio^(-1),
DisplayFunction -> $DisplayFunction, ColorOutput -> Automatic,
Axes -> Automatic, AxesOrigin -> Automatic, PlotLabel -> None,
AxesLabel -> None, Ticks -> Automatic, GridLines -> None, Prolog -> {},
Epilog -> {}, AxesStyle -> Automatic, Background -> Automatic,
DefaultColor -> Automatic, DefaultFont -> $DefaultFont,
RotateLabel -> True, Frame -> False, FrameStyle -> Automatic,
FrameTicks -> Automatic, FrameLabel -> None, PlotRegion -> Automatic,
ImageSize -> Automatic, TextStyle -> $TextStyle, FormatType -> $FormatType}]

```

```
In[184]:=
```

```
Get["C:\\work\\MathematicaData\\postscri.ps"]
```

```
Out[184]=
```

```
- Graphics -
```

```
In[185]:=
```

```
Get["C:\\work\\MathematicaData\\postscri.ps"] // FullForm
```

```
Out[185]//FullForm=
```

```
Graphics[List[List[Line[List[List[-1.9999998333333333`, -0.0122153593511319`],  
List[-1.9623003192812842`, 0.0005082587300581328`],  
List[-1.9210073169479194`, 0.018752002927052425`],  
List[-1.910299579465502`, 0.022740269212148626`], List[-1.9001774790199295`,  
0.025790484842825256`], List[-1.8945829997783903`, 0.027103353077953987`],  
List[-1.8916352820184055`, 0.02767476304778044`],  
List[-1.8885103672286825`, 0.028184000386560142`],  
List[-1.8829909524966886`, 0.028827453836099467`],  
List[-1.8778952782513714`, 0.029115238245558964`],  
List[-1.8731413317543735`, 0.029105227315202174`],  
List[-1.8680274614917154`, 0.028783053185285283`],  
List[-1.863199958792915`, 0.028174933726430717`],  
List[-1.8587831546785645`, 0.027355730435410536`],  
List[-1.8536232573178666`, 0.026078553418176047`],  
List[-1.8479909934251308`, 0.024292164406973862`],  
List[-1.8377320337083367`, 0.020012423042739088`],  
List[-1.8257366220651903`, 0.013444784972129038`],  
List[-1.8144362944250474`, 0.005937356649683411`],  
List[-1.7933141736895866`, -0.010391844332965274`],  
List[-1.768652621938747`, -0.030023715053337733`],  
List[-1.7578358910677812`, -0.03757544820569005`],  
List[-1.745962817885428`, -0.04436555940559883`],  
List[-1.7409954301270045`, -0.04661226368848812`],  
List[-1.7356703543983865`, -0.04856988304671834`],  
List[-1.7306441655063993`, -0.04995350510306155`],  
List[-1.726061421963292`, -0.05079554189295374`],  
List[-1.7208899413754872`, -0.05123896481969352`],  
List[-1.7179679228759093`, -0.05124293728587858`],  
List[-1.7152644782688042`, -0.05108393336125241`],  
List[-1.7103050520507237`, -0.0503777772427448`],  
List[-1.705036285194253`, -0.04903000884182273`],  
List[-1.6998130954536856`, -0.04707998343149028`],  
List[-1.6941824661510267`, -0.04429399736242225`],  
List[-1.6839927695262276`, -0.03748701340302931`],  
List[-1.6718751692383962`, -0.026621373541510396`],  
List[-1.660764800562505`, -0.01436785949148811`],  
List[-1.6223373700539767`, 0.03698336903382436`],  
List[-1.6122046687997946`, 0.05026582756968046`],  
List[-1.6012204218638049`, 0.0631825070687631`],  
List[-1.5915368555668616`, 0.07268572843936418`],  
List[-1.58624016644988`, 0.07694697859595899`],  
List[-1.5812846856736067`, 0.08024862424865138`],  
List[-1.5754383042802425`, 0.0832155003092454`],  
List[-1.572337956846547`, 0.08435492782676386`],  
List[-1.5690072708401817`, 0.08522809918894149`],  
List[-1.5659728443536363`, 0.08569752092152322`],  
List[-1.5630921326197909`, 0.08584815274454088`],  
List[-1.5576873525951036`, 0.08533590009119145`],  
List[-1.5523642925764947`, 0.08379516017296415`],  
List[-1.5474289001941306`, 0.08143362486126446`],  
List[-1.5420125470492216`, 0.07780448278522988`],
```

List[-1.536246079580289`, 0.07275700026813073`],  
List[-1.5249155505844803`, 0.059402686355185516`],  
List[-1.5142556242299647`, 0.04300384325934713`],  
List[-1.4943432743542384`, 0.004472710594773914`],  
List[-1.4532279345574657`, -0.08515390290044901`],  
List[-1.4432335331173123`, -0.10348325385511208`],  
List[-1.432533564692643`, -0.11946821817256431`],  
List[-1.42638894081693`, -0.12655827281836432`],  
List[-1.4207938199321637`, -0.13150559927652575`],  
List[-1.41809797650952`, -0.13334094680801603`],  
List[-1.4151693384029775`, -0.13491283836164045`],  
List[-1.4098481479164913`, -0.13660415416094238`],  
List[-1.4050416145772238`, -0.13679532253598312`],  
List[-1.4005351214442552`, -0.1357881187725129`],  
List[-1.3955913202634447`, -0.13333739136336295`],  
List[-1.3902688897887092`, -0.12910990712973433`],  
List[-1.3849829640264526`, -0.1232818098314748`],  
List[-1.3794251016434265`, -0.11542481340098515`],  
List[-1.36949576036685`, -0.09713667609456161`],  
List[-1.3516769177229944`, -0.05213560711740385`],  
List[-1.3322109850418382`, 0.010049649308713196`],  
List[-1.311814606942803`, 0.0805873886472376`],  
List[-1.301081564243093`, 0.11597117145006212`],  
List[-1.2897332454943118`, 0.14947293224243818`],  
List[-1.2797977572941663`, 0.17390652324509817`],  
List[-1.2745323012358132`, 0.1845386245184353`],  
List[-1.2688830972260063`, 0.19390980702104588`],  
List[-1.2636935060450525`, 0.20050558155842912`],  
List[-1.2589579198010443`, 0.20473215653759175`],  
List[-1.2541975830843266`, 0.20717275659823553`],  
List[-1.249684335929672`, 0.20775197540724452`],  
List[-1.247041828138711`, 0.20728903288692044`],  
List[-1.2441226325567252`, 0.20607895510875865`],  
List[-1.2410738117547113`, 0.20402551530318114`],  
List[-1.2382092114607313`, 0.20135719496762308`],  
List[-1.2330981005800692`, 0.19481588054246313`],  
List[-1.2275838049165742`, 0.18521572373110567`],  
List[-1.2214743617585684`, 0.17155776588451163`],  
List[-1.2150369184940277`, 0.1538563691722545`],  
List[-1.203523571823205`, 0.11438509646840048`],  
List[-1.1616357322503013`, -0.0838667012692308`],  
List[-1.1410574771688768`, -0.18355519085192498`],  
List[-1.1309454699394663`, -0.2252446687030288`],  
List[-1.1216144865495554`, -0.2570070238434482`],  
List[-1.116498155471344`, -0.2711457519497164`],  
List[-1.1108655908628202`, -0.28370556740513797`],  
List[-1.1081370509617479`, -0.2885858899753463`],  
List[-1.1052034083602063`, -0.29291770319564026`],  
List[-1.1021050224055187`, -0.2964329900999418`],  
List[-1.0992038714781798`, -0.29871236701229176`],  
List[-1.0937451922278991`, -0.3002768983062175`],  
List[-1.0885588150162093`, -0.29839061424267377`],  
List[-1.0839033652069439`, -0.2938521069914662`],  
List[-1.0788242866321036`, -0.28581302032434724`],  
List[-1.0735924733113682`, -0.27417965132220024`],  
List[-1.068666299969677`, -0.26016454226758307`],  
List[-1.0594607465548975`, -0.22631156310060757`],  
List[-1.0388287578280821`, -0.11852835113159345`],

List[-1.0017158165421325`, 0.1380386876955374`],  
List[-0.9804818490676098`, 0.27714025981739737`],  
List[-0.9701152472705907`, 0.3320839289435045`],  
List[-0.9606680178167547`, 0.37136370846023714`],  
List[-0.9555429871463601`, 0.3876711007347637`],  
List[-0.9506903685808147`, 0.39957150651606144`],  
List[-0.94556841761877`, 0.4081946077693798`],  
List[-0.9426751414885124`, 0.411212892719486`],  
List[-0.9399996360794677`, 0.41278513550084933`],  
List[-0.934770753521844`, 0.4124157216304582`],  
List[-0.9317946152243468`, 0.41014558532629825`],  
List[-0.929001985645631`, 0.4066458604324708`],  
List[-0.9260688269140263`, 0.40153938048657495`],  
List[-0.9233808299909719`, 0.39557434627057825`],  
List[-0.9173242395579699`, 0.3776634222547061`],  
List[-0.9127976387820719`, 0.3602999772862939`],  
List[-0.9079642254182297`, 0.3381144016387083`],  
List[-0.8992602803223766`, 0.28915729392454115`],  
List[-0.8795104906864795`, 0.1413842470955593`],  
List[-0.8397190375423287`, -0.23005651682605105`],  
List[-0.8187971422559015`, -0.4015572594838865`],  
List[-0.8082518687084622`, -0.4669266707753546`],  
List[-0.8022604864947778`, -0.49579074972261117`],  
List[-0.7966578833395991`, -0.5167287009438404`],  
List[-0.7935177520634596`, -0.525757956681514`],  
List[-0.7905381888494231`, -0.5324618998910939`],  
List[-0.7878635232886223`, -0.5368991395999997`],  
List[-0.7849493165407382`, -0.5400002249881906`],  
List[-0.7817880476133191`, -0.541289292474234`],  
List[-0.7784241259699886`, -0.5402608150821243`],  
List[-0.7752499569500357`, -0.5370024066246588`],  
List[-0.7723771954811092`, -0.5321307725565154`],  
List[-0.7668323840651169`, -0.5175679461624914`],  
List[-0.7637114688582315`, -0.5064022525594425`],  
List[-0.7608453028627378`, -0.4942884130331306`],  
List[-0.750001386238209`, -0.43286649730882254`],  
List[-0.7398238797968588`, -0.35436495342595553`],  
List[-0.7303395497421928`, -0.2654707763053889`],  
List[-0.7087968591039506`, -0.023453258580960444`],  
List[-0.6862069434280346`, 0.2506456308828193`],  
List[-0.6653448287717248`, 0.47409810939227504`],  
List[-0.6550941824327547`, 0.5599189734032894`],  
List[-0.6498581570937355`, 0.5956349265395152`],  
List[-0.6442239577120495`, 0.6271947855867068`],  
List[-0.6412828970965597`, 0.6406690475025774`],  
List[-0.6385912090240337`, 0.651124447712424`],  
List[-0.633312731602339`, 0.6662557045388081`],  
List[-0.6303669227819514`, 0.6715264501165197`],  
List[-0.6271504370779537`, 0.6746302040365503`],  
List[-0.6241122714698554`, 0.6749890352056996`],  
List[-0.6212907785054942`, 0.673065392685534`],  
List[-0.615848103126219`, 0.6631864529584339`],  
List[-0.610719214050287`, 0.6464572445824115`],  
List[-0.6061140961990733`, 0.6253747705571909`],  
List[-0.6011310493901006`, 0.5962432752291937`],  
List[-0.58993621358246`, 0.5080588617703872`],  
List[-0.5796332258514509`, 0.40179883304703257`],  
List[-0.5419911185092119`, -0.11563789469737708`],

List[-0.5212657131179157`, -0.41140376620744473`],  
List[-0.5016259570844893`, -0.6383082637499763`],  
List[-0.49139111028154175`, -0.7225368945842895`],  
List[-0.4856072258904732`, -0.7575352067107914`],  
List[-0.4802543250749207`, -0.78114955618428`],  
List[-0.4757632963228615`, -0.7941758653567549`],  
List[-0.47080452614718676`, -0.8011599707142121`],  
List[-0.46562566801066907`, -0.800017069509857`],  
List[-0.460762181661922`, -0.7910321667400466`],  
List[-0.45609740432417917`, -0.7752294176464226`],  
List[-0.4511846581763733`, -0.7510703522206259`],  
List[-0.4469901738226814`, -0.7244650488260634`],  
List[-0.4424004989015632`, -0.6892291015728184`],  
List[-0.43214488080401514`, -0.5887125083849086`],  
List[-0.4226003398105357`, -0.4709297597147562`],  
List[-0.37797369624595195`, 0.25162584454478143`],  
List[-0.35605021441036094`, 0.5894204860628905`],  
List[-0.34642322744337106`, 0.7085825762585028`],  
List[-0.34131725061385637`, 0.7619387936024965`],  
List[-0.3358302221056187`, 0.8107423480186627`],  
List[-0.33042975244711636`, 0.8495136641902389`],  
List[-0.3254936995516747`, 0.8764609305371227`],  
List[-0.32053192780411477`, 0.8950417017536064`],  
List[-0.3158200401288461`, 0.9045714043821929`],  
List[-0.3105605721446212`, 0.9057062735594216`],  
List[-0.3056826236909325`, 0.8977335870214513`],  
List[-0.30032936887031186`, 0.8790188868261573`],  
List[-0.2973379125422991`, 0.8640692854201357`],  
List[-0.29462830641223864`, 0.847791590281556`],  
List[-0.2835658740663681`, 0.7553319633410993`],  
List[-0.27733670569774405`, 0.6860096772305007`],  
List[-0.2716113406541768`, 0.6123778152682854`],  
List[-0.24994991234572664`, 0.2655800740728669`],  
List[-0.21225773815848598`, -0.43057614557528495`],  
List[-0.2014045185385447`, -0.607001214670649`],  
List[-0.1911333494049853`, -0.7490273389929445`],  
List[-0.18191240706908424`, -0.8505632497565038`],  
List[-0.1767416129962974`, -0.8952652342589389`],  
List[-0.1718920901267761`, -0.9285918556554922`],  
List[-0.16697957316058187`, -0.9535023649777139`],  
List[-0.161562565091676`, -0.9703221049620906`],  
List[-0.15628418144238038`, -0.9757476325207776`],  
List[-0.15331730465130441`, -0.9740039752842746`],  
List[-0.15059977644037958`, -0.9693770667343177`],  
List[-0.1460165797564408`, -0.9550405828469034`],  
List[-0.1410081756896768`, -0.9301062215571163`],  
List[-0.13577962138178978`, -0.893991278427419`],  
List[-0.13091999659907888`, -0.8515076827970871`],  
List[-0.12123049018585531`, -0.7427944269588428`],  
List[-0.11243516664911235`, -0.6192856959607796`],  
List[-0.09270736290683174`, -0.2771819714654778`],  
List[-0.07074128821999481`, 0.15456352776450047`],  
List[-0.047508779695364454`, 0.5802291446050808`],  
List[-0.03794148475458846`, 0.7245979117086273`],  
List[-0.027673730365563417`, 0.8500519974179697`],  
List[-0.021783670915650262`, 0.9061559525105312`],  
List[-0.01640722304970641`, 0.9463871901825228`],  
List[-0.01100269176883796`, 0.9757675666255191`],

List[-0.005877094856298342`, 0.9930655997505854`],  
List[-0.0012603543457706003`, 0.9996807302191512`],  
List[0.003841377090691665`, 0.9970355032139022`],  
List[0.00916971378005925`, 0.9831476806414581`],  
List[0.01417765010626921`, 0.9598745223762447`],  
List[0.018835778805047276`, 0.9295480348256523`],  
List[0.02311152799549632`, 0.8945820542534499`],  
List[0.03275314394312561`, 0.792159363745938`],  
List[0.05323437432389867`, 0.4834057057361746`],  
List[0.07482189650603684`, 0.07387515592061891`],  
List[0.09497769065648072`, -0.3199678960821519`],  
List[0.1165347071741706`, -0.6795558291669739`],  
List[0.12290188181040415`, -0.7637091472760054`],  
List[0.1289196612394384`, -0.8316141273358323`],  
List[0.13477808256826967`, -0.8859263593475684`],  
List[0.14018569726502234`, -0.9251000595565696`],  
List[0.14531416564759841`, -0.9521232602335068`],  
List[0.15074987886152633`, -0.9697081023973886`],  
List[0.15620612109614104`, -0.9757460076225583`],  
List[0.1593087430762231`, -0.9739713489533526`],  
List[0.16221476370581453`, -0.9688971227041098`],  
List[0.16745001383881453`, -0.9515105805200534`],  
List[0.1703767756527555`, -0.9372399699444143`],  
List[0.17302994423284257`, -0.9215400634230125`],  
List[0.18306876655623647`, -0.8393213096217108`],  
List[0.18869925168730767`, -0.7783995770475729`],  
List[0.194788749285674`, -0.7016958861226725`],  
List[0.2058274871176703`, -0.5379247908600145`],  
List[0.2472775023612705`, 0.21734939392563712`],  
List[0.2669876956883489`, 0.5466263082463674`],  
List[0.27775578978345566`, 0.6910376340383966`],  
List[0.28780683415627745`, 0.7955882301276718`],  
List[0.29221298913057425`, 0.8311225084438845`],  
List[0.2963862394019538`, 0.858646541028629`],  
List[0.30119657032466773`, 0.8827537327474217`],  
List[0.30558165684475946`, 0.8974768328574726`],  
List[0.3104357848654642`, 0.905610656162231`],  
List[0.31492594941144675`, 0.9054747211033035`],  
List[0.319854075345852`, 0.8969036163134283`],  
List[0.3250885838720468`, 0.8783015513818669`],  
List[0.32986816478766834`, 0.8529956085057786`],  
List[0.33509542949091004`, 0.8165707134217922`],  
List[0.34562891803050866`, 0.717360860055138`],  
List[0.3644582901051175`, 0.4686814149708279`],  
List[0.40689090862089167`, -0.2373117913483952`],  
List[0.426125604486471`, -0.5169094904177725`],  
List[0.43689074850830234`, -0.6388263696340332`],  
List[0.44197946198477883`, -0.6856842565689083`],  
List[0.44667041262385826`, -0.7222151789885232`],  
List[0.4510207771170432`, -0.7501332725563638`],  
List[0.4558052679116797`, -0.7740071803631008`],  
List[0.4606148703666571`, -0.7906404333586745`],  
List[0.4657059069711432`, -0.8001009858284969`],  
List[0.4701818412566341`, -0.8014806461151873`],  
List[0.47498938854669887`, -0.7957813656725395`],  
List[0.4795274484879301`, -0.7836835448282652`],  
List[0.48367316714241243`, -0.7670632421985295`],  
List[0.48910652724614345`, -0.7375047588806098`],

List[0.49421732943212265`, -0.7020199626620278`],  
List[0.5037632278937437`, -0.6174246056732089`],  
List[0.5251478059134861`, -0.35894759636214624`],  
List[0.5635945404101352`, 0.19857175803531235`],  
List[0.5742159503756941`, 0.33764078266505365`],  
List[0.5854201872696259`, 0.4642184562076209`],  
List[0.59498618095222`, 0.5515984138888202`],  
List[0.6003741082859045`, 0.591255829543805`],  
List[0.60547052133523`, 0.6219780313442358`],  
List[0.6083075285152632`, 0.6361250407609035`],  
List[0.6113889278134459`, 0.6490479456023531`],  
List[0.6167770249727986`, 0.6654477396485784`],  
List[0.6198296973349456`, 0.6712117051612357`],  
List[0.623067888318758`, 0.6745310254743232`],  
List[0.6258324575645496`, 0.6750943051423131`],  
List[0.6288793992299165`, 0.6733081258630773`],  
List[0.634303848090634`, 0.663966651997967`],  
List[0.6373520249762497`, 0.6553223823800816`],  
List[0.6401426151420501`, 0.6453200158057888`],  
List[0.645553737508074`, 0.6204158430707607`],  
List[0.650700713891915`, 0.5902905186685075`],  
List[0.6939669926869411`, 0.15752673366231526`],  
List[0.7152020721601373`, -0.09959206992178356`],  
List[0.7347515180442568`, -0.308535470979283`],  
List[0.7439550305288842`, -0.388503320333675`],  
List[0.7525526748937331`, -0.44947412460545044`],  
List[0.761520709343437`, -0.4973020699793587`],  
List[0.7662730077468173`, -0.5157230754805726`],  
List[0.7713706752869288`, -0.5299915837104358`],  
List[0.7764655824982817`, -0.5385135508046426`],  
List[0.7820111997625422`, -0.5412697136677055`],  
List[0.7872487221463473`, -0.5377047719381315`],  
List[0.7920974807320128`, -0.5291826799478381`],  
List[0.7964529521305389`, -0.5173784076900075`],  
List[0.8012635440655589`, -0.49995684200916335`],  
List[0.8110073898497123`, -0.45154940718504377`],  
List[0.8216866556673058`, -0.38077890138613835`],  
List[0.8314843412770514`, -0.30276272782875757`],  
List[0.8535723501576126`, -0.09933208457858644`],  
List[0.8742799127520443`, 0.09562488950229762`],  
List[0.8935577024893068`, 0.25127274525933935`],  
List[0.9034528690271358`, 0.3141373454867022`],  
List[0.9140306207806459`, 0.3653613330128868`],  
List[0.9200662975061342`, 0.38653503814918144`],  
List[0.9230318897571291`, 0.39471010144585883`],  
List[0.9257679017110405`, 0.40093262096077703`],  
List[0.9308868297973646`, 0.40915363270998506`],  
List[0.9364394027231373`, 0.41303250288556703`],  
List[0.9418327858488762`, 0.4118351189538449`],  
List[0.9468818702625253`, 0.4063788296314459`],  
List[0.9524211948594483`, 0.3957343604592793`],  
List[0.9555317848759625`, 0.3877026178486205`],  
List[0.9583757626343072`, 0.37911768876508517`],  
List[0.9702935393392168`, 0.33123670459888427`],  
List[0.9816146322866746`, 0.2704990999329876`],  
List[1.0219337574845162`, -0.006447852680366173`],  
List[1.0440580327404785`, -0.1494976259339504`],  
List[1.0549091834196738`, -0.2060834353790531`],

List[1.0648858169822157`, -0.2474436640448388`],  
List[1.0743602626221473`, -0.27609822989107596`],  
List[1.0791303978563087`, -0.2863886585377893`],  
List[1.0843627332104027`, -0.2944202724381302`],  
List[1.0896214981734995`, -0.29904808873811173`],  
List[1.0925692255875215`, -0.30013993168987985`],  
List[1.0952550142087834`, -0.30020482778189655`],  
List[1.1004751463501758`, -0.2978354880883429`],  
List[1.1054168621295604`, -0.2926351096973051`],  
List[1.1098516796268472`, -0.28561328923965135`],  
List[1.1145598420490528`, -0.2758356038124899`],  
List[1.1230130242444196`, -0.25271804093753775`],  
List[1.142140568536079`, -0.17872230408925346`],  
List[1.1815509438027387`, 0.01708703617472436`],  
List[1.2022395560575547`, 0.10941576698635769`],  
List[1.2126386342258384`, 0.1464307316264005`],  
List[1.218596390232883`, 0.16405381673229888`],  
List[1.2240060265504031`, 0.17759683511463492`],  
List[1.2288279588669109`, 0.18761035167194878`],  
List[1.2340514887294487`, 0.19620888071038206`],  
List[1.2395697767980838`, 0.20271393807973756`],  
List[1.2447757706178375`, 0.20641373354368245`],  
List[1.2496593073706657`, 0.20775038993425438`],  
List[1.2548260055427813`, 0.20695695951905246`],  
List[1.2592371268498836`, 0.20453200348774384`],  
List[1.26408599477186`, 0.20007719367288554`],  
List[1.268870350843575`, 0.19392842885484418`],  
List[1.2732388716663274`, 0.18687694203521504`],  
List[1.2830649217771246`, 0.1664640675381528`],  
List[1.3008089017854554`, 0.11683131393049553`],  
List[1.3395619533221224`, -0.014559309921745341`],  
List[1.3497807250584448`, -0.04656505959061242`],  
List[1.360812454108068`, -0.07699189854131996`],  
List[1.3705376522004298`, -0.09930216737751288`],  
List[1.3807913421137497`, -0.11751844389690592`],  
List[1.3866981275031691`, -0.1253500384725379`],  
List[1.389824237087754`, -0.12868205488181444`],  
List[1.393168712185386`, -0.13161783385082187`],  
List[1.3962490792977973`, -0.1337452708365822`],  
List[1.399052023099811`, -0.1352016057928178`],  
List[1.4045335760246098`, -0.13673969740956485`],  
List[1.4098285490083289`, -0.13660754156569366`],  
List[1.4128172144948126`, -0.13584859261362373`],  
List[1.4155767598509525`, -0.13472094801234416`],  
List[1.4206443460942033`, -0.1316169197489221`],  
List[1.4260171995048834`, -0.12693301271192878`],  
List[1.436311228356535`, -0.1143182202119677`],  
List[1.4459559617815967`, -0.09877639029056667`],  
List[1.4679201985139687`, -0.054223359972494584`],  
List[1.4887579343823243`, -0.00762072873950793`],  
List[1.5115059793294574`, 0.0382345442533714`],  
List[1.5219052911983821`, 0.05512766037480968`],  
List[1.5329927229796834`, 0.06937991404491133`],  
List[1.5383799169472308`, 0.07476578756256917`],  
List[1.5434555481172565`, 0.07887721208535046`],  
List[1.5480128760885281`, 0.08176009227545963`],  
List[1.5529432631496285`, 0.08401317903186631`],  
List[1.5577999319927756`, 0.08535730267782553`],

```

List[1.562334603946491`, 0.08583929036751099`],
List[1.5673100084113765`, 0.08552953552719611`],
List[1.5726443116134108`, 0.08425618533669176`],
List[1.5753219956329856`, 0.08326380557115205`],
List[1.5782350884201926`, 0.08192639160190762`],
List[1.5835319886303225`, 0.07883722949145197`],
List[1.5934602592130633`, 0.07096490215862193`],
List[1.6128244376616974`, 0.04948420103577662`],
List[1.6307422583466153`, 0.02543404632466773`],
List[1.6491132192575328`, 0.00029382516263982446`],
List[1.6684570224553763`, -0.023063944014359752`],
List[1.6785864950413831`, -0.032992068553395676`],
List[1.6896682018240128`, -0.04155430222370065`],
List[1.6945391991512138`, -0.044491418002653244`],
List[1.6997435873078401`, -0.047049906538513514`],
List[1.7046552079069046`, -0.04890844329652339`],
List[1.7091185902618236`, -0.05012830831614879`],
List[1.711901482632756`, -0.05066418084712502`],
List[1.7148481318213324`, -0.05104538406562282`],
List[1.7201314630241844`, -0.051257364803291655`],
List[1.7230685323472925`, -0.051119227645411644`],
List[1.725816724432411`, -0.05082882337742177`],
List[1.7288318026150509`, -0.050335628120028626`],
List[1.7320712690896647`, -0.049608398055116146`],
List[1.7378798445858235`, -0.047816864809224556`],
List[1.743394830049025`, -0.045575434520178475`],
List[1.7537426239565226`, -0.04012069650849713`],
List[1.7737875026298968`, -0.026116835698228463`],
List[1.7956411814708972`, -0.00850661872577543`],
List[1.8180209034331518`, 0.008440018725300728`],
List[1.8289944674381577`, 0.01538353194694639`],
List[1.8390785712365771`, 0.020647516279422223`],
List[1.8482787300585155`, 0.024393268036740877`],
List[1.8529669537056004`, 0.02589139337639477`],
List[1.8579867443579592`, 0.02718112978745319`],
List[1.8630295576621545`, 0.028148004455478503`],
List[1.8677318350704086`, 0.02875436483964524`],
List[1.8706049904237505`, 0.028986380194225708`],
List[1.873323538248825`, 0.0291106936040121`],
List[1.8784157905220895`, 0.02909979399104083`],
List[1.8834819593167174`, 0.028783911606270977`],
List[1.8863644484863291`, 0.028473886915831758`],
List[1.889067323789397`, 0.0281007099146851`],
List[1.8945097503439552`, 0.027118590924287478`],
List[1.9003696902891154`, 0.025740332252873793`],
List[1.9109593009628465`, 0.022514858084424577`],
List[1.920657051873991`, 0.018892401701557857`],
List[1.9665514341784773`, -0.0012774939524654253`],
List[1.9999983333333333`, -0.0122153593511319`]]],
List[Rule[PlotRange, Automatic], Rule[AspectRatio,
  Power[GoldenRatio, -1]],
  RuleDelayed[DisplayFunction, $DisplayFunction],
  Rule[ColorOutput, Automatic],
  Rule[Axes, Automatic],
  Rule[AxesOrigin, Automatic],
  Rule[PlotLabel, None],
  Rule[AxesLabel, None],
  Rule[Ticks, Automatic],

```

```

Rule[GridLines, None],
Rule[Prolog, List[]],
Rule[Epilog, List[]],
Rule[AxesStyle, Automatic],
Rule[Background, Automatic],
Rule[DefaultColor, Automatic],
RuleDelayed[
  DefaultFont, $DefaultFont],
Rule[RotateLabel, True],
Rule[Frame, False],
Rule[FrameStyle, Automatic],
Rule[FrameTicks, Automatic],
Rule[FrameLabel, None],
Rule[PlotRegion, Automatic],
Rule[ImageSize, Automatic],
RuleDelayed[
  TextStyle, $TextStyle],
RuleDelayed[FormatType, $FormatType]]]

```

In[186]:=

```
Get["C:\\work\\MathematicaData\\postscri.ps"]
```

Out[186]=

```
- Graphics -
```

In[187]:=

```
Get["C:\\work\\MathematicaData\\postscri.ps"] // FullForm
```

Out[187]//FullForm=

```

Graphics[List[List[Line[List[List[-1.9999998333333333`, -0.0122153593511319`],
  List[-1.9623003192812842`, 0.0005082587300581328`],
  List[-1.9210073169479194`, 0.018752002927052425`],
  List[-1.910299579465502`, 0.022740269212148626`], List[-1.9001774790199295`,
  0.025790484842825256`], List[-1.8945829997783903`, 0.027103353077953987`],
  List[-1.8916352820184055`, 0.02767476304778044`],
  List[-1.8885103672286825`, 0.028184000386560142`],
  List[-1.8829909524966886`, 0.028827453836099467`],
  List[-1.8778952782513714`, 0.029115238245558964`],
  List[-1.8731413317543735`, 0.029105227315202174`],
  List[-1.8680274614917154`, 0.028783053185285283`],
  List[-1.863199958792915`, 0.028174933726430717`],
  List[-1.8587831546785645`, 0.027355730435410536`],
  List[-1.8536232573178666`, 0.026078553418176047`],
  List[-1.8479909934251308`, 0.024292164406973862`],
  List[-1.8377320337083367`, 0.020012423042739088`],
  List[-1.8257366220651903`, 0.013444784972129038`],
  List[-1.8144362944250474`, 0.005937356649683411`],
  List[-1.7933141736895866`, -0.010391844332965274`],
  List[-1.768652621938747`, -0.030023715053337733`],
  List[-1.7578358910677812`, -0.03757544820569005`],
  List[-1.745962817885428`, -0.04436555940559883`],
  List[-1.7409954301270045`, -0.04661226368848812`],
  List[-1.7356703543983865`, -0.04856988304671834`],
  List[-1.7306441655063993`, -0.04995350510306155`],
  List[-1.726061421963292`, -0.05079554189295374`],
  List[-1.7208899413754872`, -0.05123896481969352`],
  List[-1.7179679228759093`, -0.05124293728587858`],
  List[-1.7152644782688042`, -0.05108393336125241`],
  List[-1.7103050520507237`, -0.0503777772427448`],

```

List[-1.705036285194253`, -0.04903000884182273`],  
List[-1.6998130954536856`, -0.04707998343149028`],  
List[-1.6941824661510267`, -0.04429399736242225`],  
List[-1.6839927695262276`, -0.03748701340302931`],  
List[-1.6718751692383962`, -0.026621373541510396`],  
List[-1.660764800562505`, -0.01436785949148811`],  
List[-1.6223373700539767`, 0.03698336903382436`],  
List[-1.6122046687997946`, 0.05026582756968046`],  
List[-1.6012204218638049`, 0.0631825070687631`],  
List[-1.5915368555668616`, 0.07268572843936418`],  
List[-1.58624016644988`, 0.07694697859595899`],  
List[-1.5812846856736067`, 0.08024862424865138`],  
List[-1.5754383042802425`, 0.0832155003092454`],  
List[-1.572337956846547`, 0.08435492782676386`],  
List[-1.5690072708401817`, 0.08522809918894149`],  
List[-1.5659728443536363`, 0.08569752092152322`],  
List[-1.5630921326197909`, 0.08584815274454088`],  
List[-1.5576873525951036`, 0.08533590009119145`],  
List[-1.5523642925764947`, 0.08379516017296415`],  
List[-1.5474289001941306`, 0.08143362486126446`],  
List[-1.5420125470492216`, 0.07780448278522988`],  
List[-1.536246079580289`, 0.07275700026813073`],  
List[-1.5249155505844803`, 0.059402686355185516`],  
List[-1.5142556242299647`, 0.04300384325934713`],  
List[-1.4943432743542384`, 0.004472710594773914`],  
List[-1.4532279345574657`, -0.08515390290044901`],  
List[-1.4432335331173123`, -0.10348325385511208`],  
List[-1.432533564692643`, -0.11946821817256431`],  
List[-1.42638894081693`, -0.12655827281836432`],  
List[-1.4207938199321637`, -0.13150559927652575`],  
List[-1.41809797650952`, -0.13334094680801603`],  
List[-1.4151693384029775`, -0.13491283836164045`],  
List[-1.4098481479164913`, -0.13660415416094238`],  
List[-1.4050416145772238`, -0.13679532253598312`],  
List[-1.4005351214442552`, -0.1357881187725129`],  
List[-1.3955913202634447`, -0.13333739136336295`],  
List[-1.3902688897887092`, -0.12910990712973433`],  
List[-1.3849829640264526`, -0.1232818098314748`],  
List[-1.3794251016434265`, -0.11542481340098515`],  
List[-1.36949576036685`, -0.09713667609456161`],  
List[-1.3516769177229944`, -0.05213560711740385`],  
List[-1.3322109850418382`, 0.010049649308713196`],  
List[-1.311814606942803`, 0.0805873886472376`],  
List[-1.301081564243093`, 0.11597117145006212`],  
List[-1.2897332454943118`, 0.14947293224243818`],  
List[-1.2797977572941663`, 0.17390652324509817`],  
List[-1.2745323012358132`, 0.1845386245184353`],  
List[-1.2688830972260063`, 0.19390980702104588`],  
List[-1.2636935060450525`, 0.20050558155842912`],  
List[-1.2589579198010443`, 0.20473215653759175`],  
List[-1.2541975830843266`, 0.20717275659823553`],  
List[-1.249684335929672`, 0.20775197540724452`],  
List[-1.247041828138711`, 0.20728903288692044`],  
List[-1.2441226325567252`, 0.20607895510875865`],  
List[-1.2410738117547113`, 0.20402551530318114`],  
List[-1.2382092114607313`, 0.20135719496762308`],  
List[-1.2330981005800692`, 0.19481588054246313`],  
List[-1.2275838049165742`, 0.18521572373110567`],

List[-1.2214743617585684`, 0.17155776588451163`],  
List[-1.2150369184940277`, 0.1538563691722545`],  
List[-1.203523571823205`, 0.11438509646840048`],  
List[-1.1616357322503013`, -0.0838667012692308`],  
List[-1.1410574771688768`, -0.18355519085192498`],  
List[-1.1309454699394663`, -0.2252446687030288`],  
List[-1.1216144865495554`, -0.2570070238434482`],  
List[-1.116498155471344`, -0.2711457519497164`],  
List[-1.1108655908628202`, -0.28370556740513797`],  
List[-1.1081370509617479`, -0.2885858899753463`],  
List[-1.1052034083602063`, -0.29291770319564026`],  
List[-1.1021050224055187`, -0.2964329900999418`],  
List[-1.0992038714781798`, -0.29871236701229176`],  
List[-1.0937451922278991`, -0.3002768983062175`],  
List[-1.0885588150162093`, -0.29839061424267377`],  
List[-1.0839033652069439`, -0.2938521069914662`],  
List[-1.0788242866321036`, -0.28581302032434724`],  
List[-1.0735924733113682`, -0.27417965132220024`],  
List[-1.068666299969677`, -0.26016454226758307`],  
List[-1.0594607465548975`, -0.22631156310060757`],  
List[-1.0388287578280821`, -0.11852835113159345`],  
List[-1.0017158165421325`, 0.1380386876955374`],  
List[-0.9804818490676098`, 0.27714025981739737`],  
List[-0.9701152472705907`, 0.3320839289435045`],  
List[-0.9606680178167547`, 0.37136370846023714`],  
List[-0.9555429871463601`, 0.3876711007347637`],  
List[-0.9506903685808147`, 0.39957150651606144`],  
List[-0.94556841761877`, 0.4081946077693798`],  
List[-0.9426751414885124`, 0.411212892719486`],  
List[-0.9399996360794677`, 0.41278513550084933`],  
List[-0.934770753521844`, 0.4124157216304582`],  
List[-0.9317946152243468`, 0.41014558532629825`],  
List[-0.929001985645631`, 0.4066458604324708`],  
List[-0.9260688269140263`, 0.40153938048657495`],  
List[-0.9233808299909719`, 0.39557434627057825`],  
List[-0.9173242395579699`, 0.3776634222547061`],  
List[-0.9127976387820719`, 0.3602999772862939`],  
List[-0.9079642254182297`, 0.3381144016387083`],  
List[-0.8992602803223766`, 0.28915729392454115`],  
List[-0.8795104906864795`, 0.1413842470955593`],  
List[-0.8397190375423287`, -0.23005651682605105`],  
List[-0.8187971422559015`, -0.4015572594838865`],  
List[-0.8082518687084622`, -0.4669266707753546`],  
List[-0.8022604864947778`, -0.49579074972261117`],  
List[-0.7966578833395991`, -0.5167287009438404`],  
List[-0.7935177520634596`, -0.525757956681514`],  
List[-0.7905381888494231`, -0.5324618998910939`],  
List[-0.7878635232886223`, -0.5368991395999997`],  
List[-0.7849493165407382`, -0.5400002249881906`],  
List[-0.7817880476133191`, -0.541289292474234`],  
List[-0.7784241259699886`, -0.5402608150821243`],  
List[-0.7752499569500357`, -0.5370024066246588`],  
List[-0.7723771954811092`, -0.5321307725565154`],  
List[-0.7668323840651169`, -0.5175679461624914`],  
List[-0.7637114688582315`, -0.5064022525594425`],  
List[-0.7608453028627378`, -0.4942884130331306`],  
List[-0.750001386238209`, -0.43286649730882254`],  
List[-0.7398238797968588`, -0.35436495342595553`],

List[-0.7303395497421928`, -0.2654707763053889`],  
List[-0.7087968591039506`, -0.023453258580960444`],  
List[-0.6862069434280346`, 0.2506456308828193`],  
List[-0.6653448287717248`, 0.47409810939227504`],  
List[-0.6550941824327547`, 0.5599189734032894`],  
List[-0.6498581570937355`, 0.5956349265395152`],  
List[-0.6442239577120495`, 0.6271947855867068`],  
List[-0.6412828970965597`, 0.6406690475025774`],  
List[-0.6385912090240337`, 0.651124447712424`],  
List[-0.633312731602339`, 0.6662557045388081`],  
List[-0.6303669227819514`, 0.6715264501165197`],  
List[-0.6271504370779537`, 0.6746302040365503`],  
List[-0.6241122714698554`, 0.6749890352056996`],  
List[-0.6212907785054942`, 0.673065392685534`],  
List[-0.615848103126219`, 0.6631864529584339`],  
List[-0.610719214050287`, 0.6464572445824115`],  
List[-0.6061140961990733`, 0.6253747705571909`],  
List[-0.6011310493901006`, 0.5962432752291937`],  
List[-0.58993621358246`, 0.5080588617703872`],  
List[-0.5796332258514509`, 0.40179883304703257`],  
List[-0.5419911185092119`, -0.11563789469737708`],  
List[-0.5212657131179157`, -0.41140376620744473`],  
List[-0.5016259570844893`, -0.6383082637499763`],  
List[-0.49139111028154175`, -0.7225368945842895`],  
List[-0.4856072258904732`, -0.7575352067107914`],  
List[-0.4802543250749207`, -0.78114955618428`],  
List[-0.4757632963228615`, -0.7941758653567549`],  
List[-0.47080452614718676`, -0.8011599707142121`],  
List[-0.46562566801066907`, -0.800017069509857`],  
List[-0.460762181661922`, -0.7910321667400466`],  
List[-0.45609740432417917`, -0.7752294176464226`],  
List[-0.4511846581763733`, -0.7510703522206259`],  
List[-0.4469901738226814`, -0.7244650488260634`],  
List[-0.4424004989015632`, -0.6892291015728184`],  
List[-0.43214488080401514`, -0.5887125083849086`],  
List[-0.4226003398105357`, -0.4709297597147562`],  
List[-0.37797369624595195`, 0.25162584454478143`],  
List[-0.35605021441036094`, 0.5894204860628905`],  
List[-0.34642322744337106`, 0.7085825762585028`],  
List[-0.34131725061385637`, 0.7619387936024965`],  
List[-0.3358302221056187`, 0.8107423480186627`],  
List[-0.33042975244711636`, 0.8495136641902389`],  
List[-0.3254936995516747`, 0.8764609305371227`],  
List[-0.32053192780411477`, 0.8950417017536064`],  
List[-0.3158200401288461`, 0.9045714043821929`],  
List[-0.3105605721446212`, 0.9057062735594216`],  
List[-0.3056826236909325`, 0.8977335870214513`],  
List[-0.30032936887031186`, 0.8790188868261573`],  
List[-0.2973379125422991`, 0.8640692854201357`],  
List[-0.29462830641223864`, 0.847791590281556`],  
List[-0.2835658740663681`, 0.7553319633410993`],  
List[-0.27733670569774405`, 0.6860096772305007`],  
List[-0.2716113406541768`, 0.6123778152682854`],  
List[-0.24994991234572664`, 0.2655800740728669`],  
List[-0.21225773815848598`, -0.43057614557528495`],  
List[-0.2014045185385447`, -0.607001214670649`],  
List[-0.1911333494049853`, -0.7490273389929445`],  
List[-0.18191240706908424`, -0.8505632497565038`],

List[-0.1767416129962974`, -0.8952652342589389`],  
List[-0.1718920901267761`, -0.9285918556554922`],  
List[-0.16697957316058187`, -0.9535023649777139`],  
List[-0.161562565091676`, -0.9703221049620906`],  
List[-0.15628418144238038`, -0.9757476325207776`],  
List[-0.15331730465130441`, -0.9740039752842746`],  
List[-0.15059977644037958`, -0.9693770667343177`],  
List[-0.1460165797564408`, -0.9550405828469034`],  
List[-0.1410081756896768`, -0.9301062215571163`],  
List[-0.13577962138178978`, -0.893991278427419`],  
List[-0.13091999659907888`, -0.8515076827970871`],  
List[-0.12123049018585531`, -0.7427944269588428`],  
List[-0.11243516664911235`, -0.6192856959607796`],  
List[-0.09270736290683174`, -0.2771819714654778`],  
List[-0.07074128821999481`, 0.15456352776450047`],  
List[-0.047508779695364454`, 0.5802291446050808`],  
List[-0.03794148475458846`, 0.7245979117086273`],  
List[-0.027673730365563417`, 0.8500519974179697`],  
List[-0.021783670915650262`, 0.9061559525105312`],  
List[-0.01640722304970641`, 0.9463871901825228`],  
List[-0.01100269176883796`, 0.9757675666255191`],  
List[-0.005877094856298342`, 0.9930655997505854`],  
List[-0.0012603543457706003`, 0.9996807302191512`],  
List[0.003841377090691665`, 0.9970355032139022`],  
List[0.00916971378005925`, 0.9831476806414581`],  
List[0.01417765010626921`, 0.9598745223762447`],  
List[0.018835778805047276`, 0.9295480348256523`],  
List[0.02311152799549632`, 0.8945820542534499`],  
List[0.03275314394312561`, 0.792159363745938`],  
List[0.05323437432389867`, 0.4834057057361746`],  
List[0.07482189650603684`, 0.07387515592061891`],  
List[0.09497769065648072`, -0.3199678960821519`],  
List[0.1165347071741706`, -0.6795558291669739`],  
List[0.12290188181040415`, -0.7637091472760054`],  
List[0.1289196612394384`, -0.8316141273358323`],  
List[0.13477808256826967`, -0.8859263593475684`],  
List[0.14018569726502234`, -0.9251000595565696`],  
List[0.14531416564759841`, -0.9521232602335068`],  
List[0.15074987886152633`, -0.9697081023973886`],  
List[0.15620612109614104`, -0.9757460076225583`],  
List[0.1593087430762231`, -0.9739713489533526`],  
List[0.16221476370581453`, -0.9688971227041098`],  
List[0.16745001383881453`, -0.9515105805200534`],  
List[0.1703767756527555`, -0.9372399699444143`],  
List[0.17302994423284257`, -0.9215400634230125`],  
List[0.18306876655623647`, -0.8393213096217108`],  
List[0.18869925168730767`, -0.7783995770475729`],  
List[0.194788749285674`, -0.7016958861226725`],  
List[0.2058274871176703`, -0.5379247908600145`],  
List[0.2472775023612705`, 0.21734939392563712`],  
List[0.2669876956883489`, 0.5466263082463674`],  
List[0.27775578978345566`, 0.6910376340383966`],  
List[0.28780683415627745`, 0.7955882301276718`],  
List[0.29221298913057425`, 0.8311225084438845`],  
List[0.2963862394019538`, 0.858646541028629`],  
List[0.30119657032466773`, 0.8827537327474217`],  
List[0.30558165684475946`, 0.8974768328574726`],  
List[0.3104357848654642`, 0.905610656162231`],

List[0.31492594941144675`, 0.9054747211033035`],  
List[0.319854075345852`, 0.8969036163134283`],  
List[0.3250885838720468`, 0.8783015513818669`],  
List[0.32986816478766834`, 0.8529956085057786`],  
List[0.33509542949091004`, 0.8165707134217922`],  
List[0.34562891803050866`, 0.717360860055138`],  
List[0.3644582901051175`, 0.4686814149708279`],  
List[0.40689090862089167`, -0.2373117913483952`],  
List[0.426125604486471`, -0.5169094904177725`],  
List[0.43689074850830234`, -0.6388263696340332`],  
List[0.44197946198477883`, -0.6856842565689083`],  
List[0.44667041262385826`, -0.7222151789885232`],  
List[0.4510207771170432`, -0.7501332725563638`],  
List[0.4558052679116797`, -0.7740071803631008`],  
List[0.4606148703666571`, -0.7906404333586745`],  
List[0.4657059069711432`, -0.8001009858284969`],  
List[0.4701818412566341`, -0.8014806461151873`],  
List[0.47498938854669887`, -0.7957813656725395`],  
List[0.4795274484879301`, -0.7836835448282652`],  
List[0.48367316714241243`, -0.7670632421985295`],  
List[0.48910652724614345`, -0.7375047588806098`],  
List[0.49421732943212265`, -0.7020199626620278`],  
List[0.5037632278937437`, -0.6174246056732089`],  
List[0.5251478059134861`, -0.35894759636214624`],  
List[0.5635945404101352`, 0.19857175803531235`],  
List[0.5742159503756941`, 0.33764078266505365`],  
List[0.5854201872696259`, 0.4642184562076209`],  
List[0.59498618095222`, 0.5515984138888202`],  
List[0.6003741082859045`, 0.591255829543805`],  
List[0.60547052133523`, 0.6219780313442358`],  
List[0.6083075285152632`, 0.6361250407609035`],  
List[0.6113889278134459`, 0.6490479456023531`],  
List[0.6167770249727986`, 0.6654477396485784`],  
List[0.6198296973349456`, 0.6712117051612357`],  
List[0.623067888318758`, 0.6745310254743232`],  
List[0.6258324575645496`, 0.6750943051423131`],  
List[0.6288793992299165`, 0.6733081258630773`],  
List[0.634303848090634`, 0.663966651997967`],  
List[0.6373520249762497`, 0.6553223823800816`],  
List[0.6401426151420501`, 0.6453200158057888`],  
List[0.645553737508074`, 0.6204158430707607`],  
List[0.650700713891915`, 0.5902905186685075`],  
List[0.6939669926869411`, 0.15752673366231526`],  
List[0.7152020721601373`, -0.09959206992178356`],  
List[0.7347515180442568`, -0.308535470979283`],  
List[0.7439550305288842`, -0.388503320333675`],  
List[0.7525526748937331`, -0.44947412460545044`],  
List[0.761520709343437`, -0.4973020699793587`],  
List[0.7662730077468173`, -0.5157230754805726`],  
List[0.7713706752869288`, -0.5299915837104358`],  
List[0.7764655824982817`, -0.5385135508046426`],  
List[0.7820111997625422`, -0.5412697136677055`],  
List[0.7872487221463473`, -0.5377047719381315`],  
List[0.7920974807320128`, -0.5291826799478381`],  
List[0.7964529521305389`, -0.5173784076900075`],  
List[0.8012635440655589`, -0.49995684200916335`],  
List[0.8110073898497123`, -0.45154940718504377`],  
List[0.8216866556673058`, -0.38077890138613835`],

List[0.8314843412770514`, -0.30276272782875757`],  
List[0.8535723501576126`, -0.09933208457858644`],  
List[0.8742799127520443`, 0.09562488950229762`],  
List[0.8935577024893068`, 0.25127274525933935`],  
List[0.9034528690271358`, 0.3141373454867022`],  
List[0.9140306207806459`, 0.3653613330128868`],  
List[0.9200662975061342`, 0.38653503814918144`],  
List[0.9230318897571291`, 0.39471010144585883`],  
List[0.9257679017110405`, 0.40093262096077703`],  
List[0.9308868297973646`, 0.40915363270998506`],  
List[0.9364394027231373`, 0.41303250288556703`],  
List[0.9418327858488762`, 0.4118351189538449`],  
List[0.9468818702625253`, 0.4063788296314459`],  
List[0.9524211948594483`, 0.3957343604592793`],  
List[0.9555317848759625`, 0.3877026178486205`],  
List[0.9583757626343072`, 0.37911768876508517`],  
List[0.9702935393392168`, 0.33123670459888427`],  
List[0.9816146322866746`, 0.2704990999329876`],  
List[1.0219337574845162`, -0.006447852680366173`],  
List[1.0440580327404785`, -0.1494976259339504`],  
List[1.0549091834196738`, -0.2060834353790531`],  
List[1.0648858169822157`, -0.2474436640448388`],  
List[1.0743602626221473`, -0.27609822989107596`],  
List[1.0791303978563087`, -0.2863886585377893`],  
List[1.0843627332104027`, -0.2944202724381302`],  
List[1.0896214981734995`, -0.29904808873811173`],  
List[1.0925692255875215`, -0.30013993168987985`],  
List[1.0952550142087834`, -0.30020482778189655`],  
List[1.1004751463501758`, -0.2978354880883429`],  
List[1.1054168621295604`, -0.2926351096973051`],  
List[1.1098516796268472`, -0.28561328923965135`],  
List[1.1145598420490528`, -0.2758356038124899`],  
List[1.1230130242444196`, -0.25271804093753775`],  
List[1.142140568536079`, -0.17872230408925346`],  
List[1.1815509438027387`, 0.01708703617472436`],  
List[1.2022395560575547`, 0.10941576698635769`],  
List[1.2126386342258384`, 0.1464307316264005`],  
List[1.218596390232883`, 0.16405381673229888`],  
List[1.2240060265504031`, 0.17759683511463492`],  
List[1.2288279588669109`, 0.18761035167194878`],  
List[1.2340514887294487`, 0.19620888071038206`],  
List[1.2395697767980838`, 0.20271393807973756`],  
List[1.2447757706178375`, 0.20641373354368245`],  
List[1.2496593073706657`, 0.20775038993425438`],  
List[1.2548260055427813`, 0.20695695951905246`],  
List[1.2592371268498836`, 0.20453200348774384`],  
List[1.26408599477186`, 0.20007719367288554`],  
List[1.268870350843575`, 0.19392842885484418`],  
List[1.2732388716663274`, 0.18687694203521504`],  
List[1.2830649217771246`, 0.1664640675381528`],  
List[1.3008089017854554`, 0.11683131393049553`],  
List[1.3395619533221224`, -0.014559309921745341`],  
List[1.3497807250584448`, -0.04656505959061242`],  
List[1.360812454108068`, -0.07699189854131996`],  
List[1.3705376522004298`, -0.09930216737751288`],  
List[1.3807913421137497`, -0.11751844389690592`],  
List[1.3866981275031691`, -0.1253500384725379`],  
List[1.389824237087754`, -0.12868205488181444`],

List[1.393168712185386`, -0.13161783385082187`],  
List[1.3962490792977973`, -0.1337452708365822`],  
List[1.399052023099811`, -0.1352016057928178`],  
List[1.4045335760246098`, -0.13673969740956485`],  
List[1.4098285490083289`, -0.13660754156569366`],  
List[1.4128172144948126`, -0.13584859261362373`],  
List[1.4155767598509525`, -0.13472094801234416`],  
List[1.4206443460942033`, -0.1316169197489221`],  
List[1.4260171995048834`, -0.12693301271192878`],  
List[1.436311228356535`, -0.1143182202119677`],  
List[1.4459559617815967`, -0.09877639029056667`],  
List[1.4679201985139687`, -0.054223359972494584`],  
List[1.4887579343823243`, -0.00762072873950793`],  
List[1.5115059793294574`, 0.0382345442533714`],  
List[1.5219052911983821`, 0.05512766037480968`],  
List[1.5329927229796834`, 0.06937991404491133`],  
List[1.5383799169472308`, 0.07476578756256917`],  
List[1.5434555481172565`, 0.07887721208535046`],  
List[1.5480128760885281`, 0.08176009227545963`],  
List[1.5529432631496285`, 0.08401317903186631`],  
List[1.5577999319927756`, 0.08535730267782553`],  
List[1.562334603946491`, 0.08583929036751099`],  
List[1.5673100084113765`, 0.08552953552719611`],  
List[1.5726443116134108`, 0.08425618533669176`],  
List[1.5753219956329856`, 0.08326380557115205`],  
List[1.5782350884201926`, 0.08192639160190762`],  
List[1.5835319886303225`, 0.07883722949145197`],  
List[1.5934602592130633`, 0.07096490215862193`],  
List[1.6128244376616974`, 0.04948420103577662`],  
List[1.6307422583466153`, 0.02543404632466773`],  
List[1.6491132192575328`, 0.00029382516263982446`],  
List[1.6684570224553763`, -0.023063944014359752`],  
List[1.6785864950413831`, -0.032992068553395676`],  
List[1.6896682018240128`, -0.04155430222370065`],  
List[1.6945391991512138`, -0.044491418002653244`],  
List[1.6997435873078401`, -0.047049906538513514`],  
List[1.7046552079069046`, -0.04890844329652339`],  
List[1.7091185902618236`, -0.05012830831614879`],  
List[1.711901482632756`, -0.05066418084712502`],  
List[1.7148481318213324`, -0.05104538406562282`],  
List[1.7201314630241844`, -0.051257364803291655`],  
List[1.7230685323472925`, -0.051119227645411644`],  
List[1.725816724432411`, -0.05082882337742177`],  
List[1.7288318026150509`, -0.050335628120028626`],  
List[1.7320712690896647`, -0.049608398055116146`],  
List[1.7378798445858235`, -0.047816864809224556`],  
List[1.743394830049025`, -0.045575434520178475`],  
List[1.7537426239565226`, -0.04012069650849713`],  
List[1.7737875026298968`, -0.026116835698228463`],  
List[1.7956411814708972`, -0.00850661872577543`],  
List[1.8180209034331518`, 0.008440018725300728`],  
List[1.8289944674381577`, 0.01538353194694639`],  
List[1.8390785712365771`, 0.020647516279422223`],  
List[1.8482787300585155`, 0.024393268036740877`],  
List[1.8529669537056004`, 0.02589139337639477`],  
List[1.8579867443579592`, 0.02718112978745319`],  
List[1.8630295576621545`, 0.028148004455478503`],  
List[1.8677318350704086`, 0.02875436483964524`],

```

List[1.8706049904237505`, 0.028986380194225708`],
List[1.873323538248825`, 0.0291106936040121`],
List[1.8784157905220895`, 0.02909979399104083`],
List[1.8834819593167174`, 0.028783911606270977`],
List[1.8863644484863291`, 0.028473886915831758`],
List[1.889067323789397`, 0.0281007099146851`],
List[1.8945097503439552`, 0.027118590924287478`],
List[1.9003696902891154`, 0.025740332252873793`],
List[1.9109593009628465`, 0.022514858084424577`],
List[1.920657051873991`, 0.018892401701557857`],
List[1.9665514341784773`, -0.0012774939524654253`],
List[1.9999983333333333`, -0.0122153593511319`]]]],
List[Rule[PlotRange, Automatic], Rule[AspectRatio,
  Power[GoldenRatio, -1]],
  RuleDelayed[DisplayFunction, $DisplayFunction],
  Rule[ColorOutput, Automatic],
  Rule[Axes, Automatic],
  Rule[AxesOrigin, Automatic],
  Rule[PlotLabel, None],
  Rule[AxesLabel, None],
  Rule[Ticks, Automatic],
  Rule[GridLines, None],
  Rule[Prolog, List[]],
  Rule[Epilog, List[]],
  Rule[AxesStyle, Automatic],
  Rule[Background, Automatic],
  Rule[DefaultColor, Automatic],
  RuleDelayed[
    DefaultFont, $DefaultFont],
  Rule[RotateLabel, True],
  Rule[Frame, False],
  Rule[FrameStyle, Automatic],
  Rule[FrameTicks, Automatic],
  Rule[FrameLabel, None],
  Rule[PlotRegion, Automatic],
  Rule[ImageSize, Automatic],
  RuleDelayed[
    TextStyle, $TextStyle],
  RuleDelayed[FormatType, $FormatType]]]

```

In[188]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.nb"]
```

Out[188]=

```
StandardForm
```

In[189]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.ps"]
```

Out[189]=

```
- Graphics -
```

In[190]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.ps"] // FullForm
```

Out[190]//FullForm=

```
Graphics[List[List[Line[List[List[-1.9999983333333333`, 0.018315651099163666`],
  List[-1.9210073169479194`, 0.024965282686839303`],
  List[-1.8377320337083367`, 0.034140905879298085`],
  List[-1.7532736698276103`, 0.0462372947296657`], List[-1.660764800562505`,
```

```
0.06340961817674262`], List[-1.5785622264174572`, 0.08275600472773463`],
List[-1.4906280683541824`, 0.10839513856535575`],
List[-1.3165568698541965`, 0.176696449796265`],
List[-1.1488517715417124`, 0.2671725985341265`],
List[-0.9875127734167299`, 0.3771239077878865`],
List[-0.8158732088125825`, 0.5139398247464876`],
List[-0.6505997443959369`, 0.6548952188218141`],
List[-0.49169238016679306`, 0.7852435168844217`],
List[-0.4033296982844994`, 0.8498674817137135`],
List[-0.32248444945848426`, 0.9012287039308049`],
List[-0.238040189216551`, 0.9449123267840369`],
List[-0.1596426189376772`, 0.9748362548994524`],
List[-0.1210479274390275`, 0.9854542262161895`],
List[-0.1012837918018679`, 0.9897940315067599`],
List[-0.07990432059860685`, 0.9936356384406656`],
List[-0.06766226347739546`, 0.9954322820012288`],
List[-0.056398986936803536`, 0.9968242078026867`],
List[-0.04584278769357836`, 0.997900645550053`],
List[-0.03425568918503252`, 0.9988272359843137`],
List[-0.029232125698399822`, 0.9991458478229799`],
List[-0.023846829181437094`, 0.9994314904008477`],
List[-0.019192329184449866`, 0.9996317223313506`],
List[-0.01412923021916404`, 0.9998003847791838`],
List[-0.011162534426455067`, 0.9998754055877068`],
List[-0.008417754493378532`, 0.9999291439196999`],
List[-0.003007142183362502`, 0.999990957136776`],
List[0.001909918777716465`, 0.9999963522169157`],
List[0.007133677874793013`, 0.9999491119348202`],
List[0.011909983895002052`, 0.9998581623435324`],
List[0.017134671691070795`, 0.9997064461214131`],
List[0.02258923333064591`, 0.9994898567048957`],
List[0.027664022579297086`, 0.9992349946206686`],
List[0.03676506101645184`, 0.9986492433825143`],
List[0.046487856778484764`, 0.9978412127124359`],
List[0.0687153236097172`, 0.9952893344535297`],
List[0.08891189341574972`, 0.9921258401122844`],
List[0.1284035999582669`, 0.9836476901737676`],
List[0.16568627149973345`, 0.9729214394860457`],
List[0.24974313299960782`, 0.9395336606784316`],
List[0.327244529916323`, 0.8984457236123577`],
List[0.4927372546245758`, 0.7844362264404803`],
List[0.6518638791453268`, 0.6538178279030031`],
List[0.8212910701452427`, 0.5094013725987232`],
List[0.9843521609576569`, 0.37948159673203985`],
List[1.157713818249236`, 0.26176679030019007`],
List[1.3247093753533132`, 0.17293232791205787`],
List[1.4853388322698888`, 0.1101148412255609`],
List[1.5681280856924957`, 0.08551823139009224`],
List[1.6562688556656293`, 0.06436234497421982`],
List[1.7463673615481474`, 0.047368446316283896`],
List[1.831133345353033`, 0.0349775333298323`],
List[1.91562847179078`, 0.025485835749798215`],
List[1.9932656346654332`, 0.01881486762938725`],
List[1.9999983333333333`, 0.018315651099163666`]]],
List[Rule[PlotRange, Automatic], Rule[AspectRatio,
Power[GoldenRatio, -1]],
RuleDelayed[DisplayFunction, $DisplayFunction],
Rule[ColorOutput, Automatic],
```

```
Rule[Axes, Automatic],
Rule[AxesOrigin, Automatic],
Rule[PlotLabel, None],
Rule[AxesLabel, None],
Rule[Ticks, Automatic],
Rule[GridLines, None],
Rule[Prolog, List[]],
Rule[Epilog, List[]],
Rule[AxesStyle, Automatic],
Rule[Background, Automatic],
Rule[DefaultColor, Automatic],
RuleDelayed[DefaultFont, $DefaultFont],
Rule[RotateLabel, True],
Rule[Frame, False],
Rule[FrameStyle, Automatic],
Rule[FrameTicks, Automatic],
Rule[FrameLabel, None],
Rule[PlotRegion, Automatic],
Rule[ImageSize, Automatic],
RuleDelayed[TextStyle, $TextStyle],
RuleDelayed[FormatType, $FormatType]]]
```

In[191]:=

```
Remove[expl]
```

In[192]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.ps",expl]
```

Get::notencode : Warning: the file C:\work\MathematicaData\postsc\_1.ps is not encoded. Mehr...

Out[192]=

```
- Graphics -
```

In[193]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.ps",expl]; Show[expl]
```

Get::notencode : Warning: the file C:\work\MathematicaData\postsc\_1.ps is not encoded. Mehr...

Show::gtype : Symbol is not a type of graphics. Mehr...

Out[193]=

```
Show[expl]
```

In[194]:=

```
Get["C:\\work\\MathematicaData\\postscri.ma"]
```

Out[194]=

```
StandardForm
```

In[195]:=

```
Get["C:\\work\\MathematicaData\\postscri.ma"] // FullForm
```

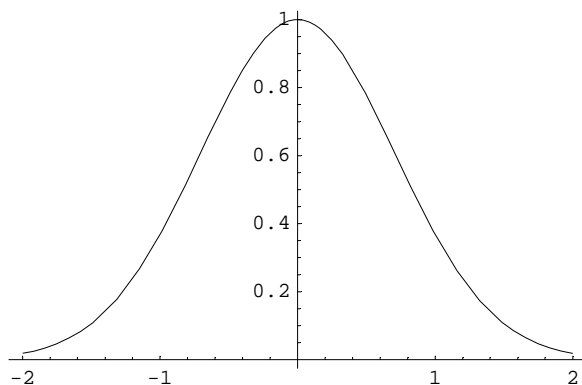
Out[195]//FullForm=

```
StandardForm
```

In[196]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.ma",exp1]; Show[exp1]
```

Get::notencode : Warning: the file C:\work\MathematicaData\postsc\_1.ma is not encoded. Mehr...



Out[196]=

- Graphics -

In[197]:=

```
Remove[expCos, exp1]
```

In[198]:=

```
Get["C:\\work\\MathematicaData\\postscri.nb"]
```

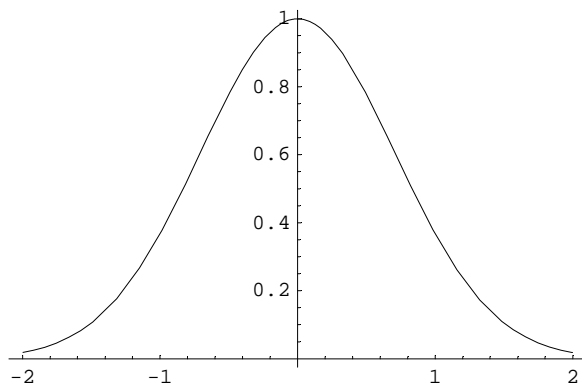
Out[198]=

StandardForm

In[199]:=

```
Get["C:\\work\\MathematicaData\\postsc_1.nb",exp1]; Show[exp1]
```

Get::notencode : Warning: the file C:\work\MathematicaData\postsc\_1.nb is not encoded. Mehr...



Out[199]=

- Graphics -

In[200]:=

```
exp1
```

Out[200]=

- Graphics -

## 4. 14. Experimentieren mit Animationen

### ■ Expérimenter les animations

#### 4.14.1. Nachfolgend zwei Beispiele zu Animationen (Bsp. von U.Altmann)

##### ■ Ensuite deux exemples d'animations (exemple pris de U.Altmann)

Achtung! Die Rechnungen dauern vielleicht etwas Zeit!  
Vorschlag: Erst am Schluss der Arbeit ausführen!



Attention! Il faut du temps aux calculs! Proposition:  
Executer à la fin du travail!

Package laden:

■ Charger Package:

`In[201]:=`

```
?ImplicitPlot
```

```
Information::notfound: Symbol ImplicitPlot not found. Mehr...
```

`In[202]:=`

```
Remove[ImplicitPlot]
```

`In[203]:=`

```
<< Graphics`ImplicitPlot`
```

`In[204]:=`

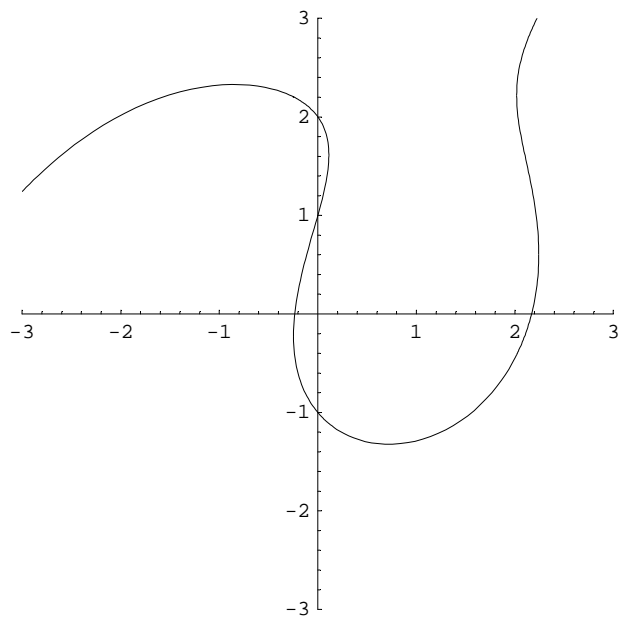
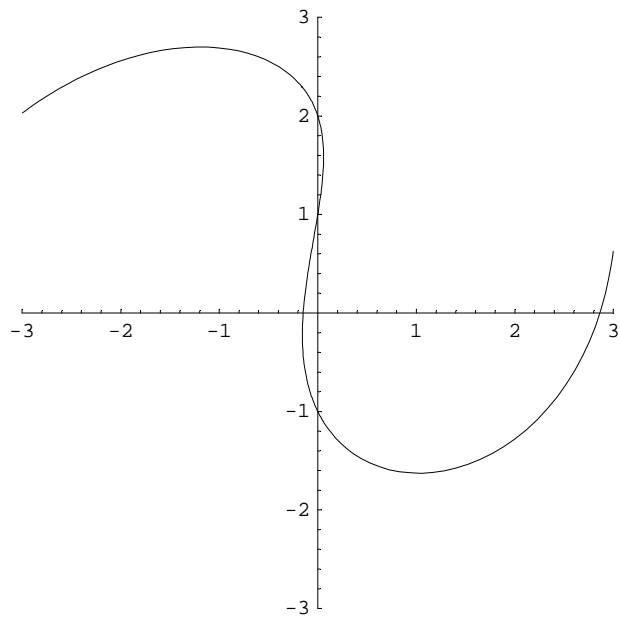
```
(* Old: *)
(* Needs["Graphics`ImplicitPlot`"] *)
```

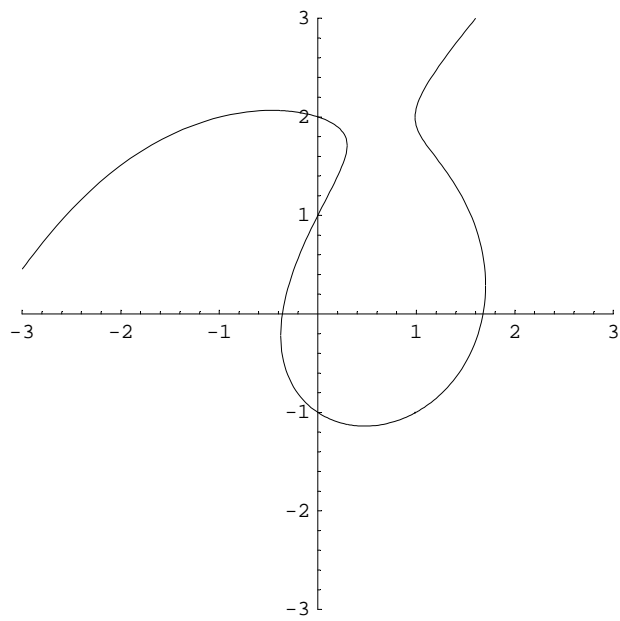
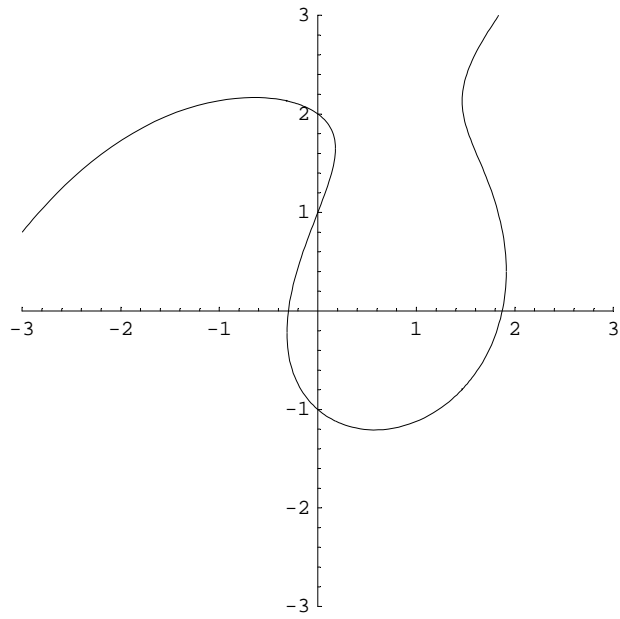
Beispiel: Zuerst  $n = 128$  Graphiken rechnen (das dauert!!!). Um Ueberraschungen zu vermeiden, ist  $n$  auf 10 eingestellt. Das geht zwar schneller, aber die Animation wird so nicht gut. Sonst kann es schon eine Lektion dauern!

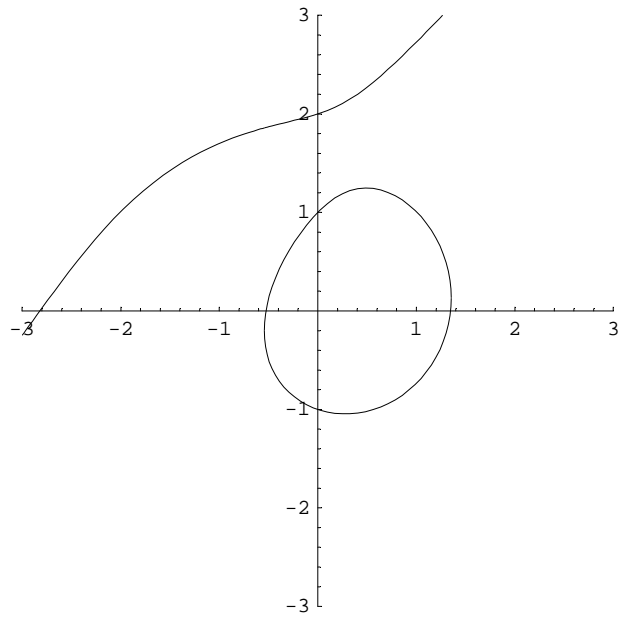
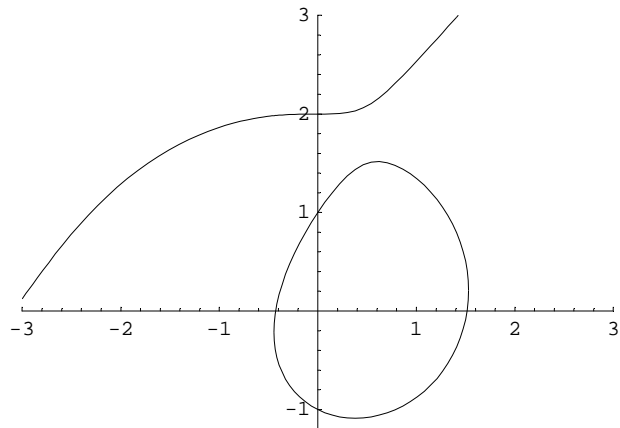
■ Exemple: Calculer d'abord  $n = 128$  graphiques (cela dure!!!). Pour éviter des surprises, il faut mettre  $n$  sur 10. Cela va plus vite, mais l'animation n'est pas si bonne. Si non cela peut durer toute une leçon.

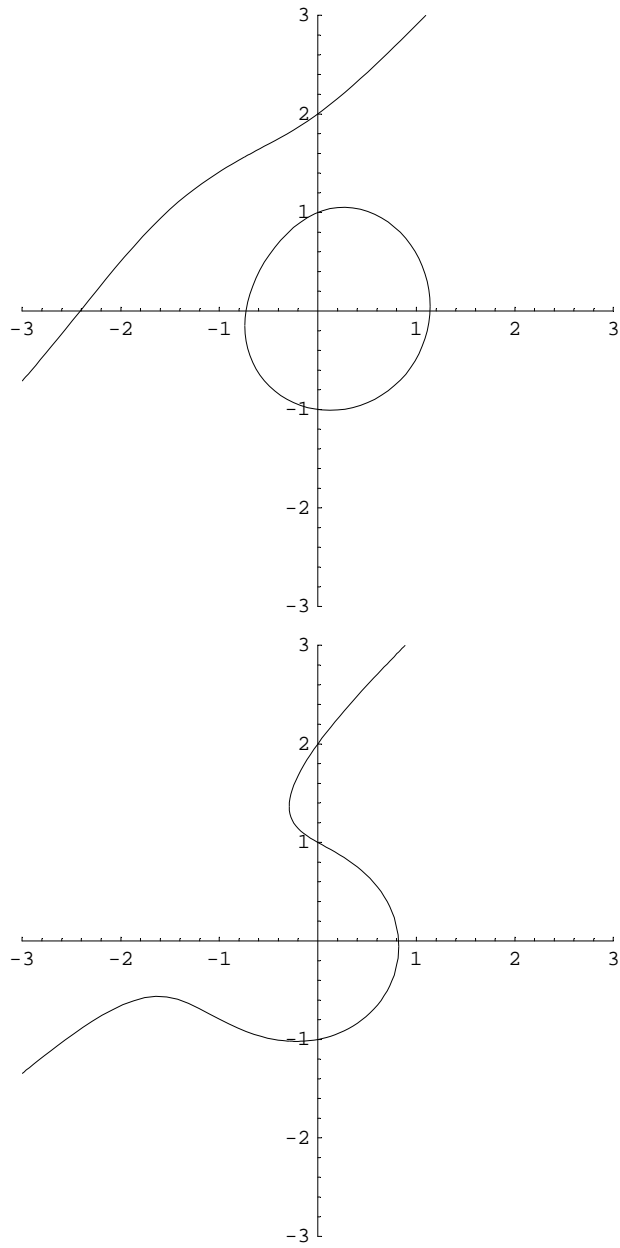
`In[206]:=`

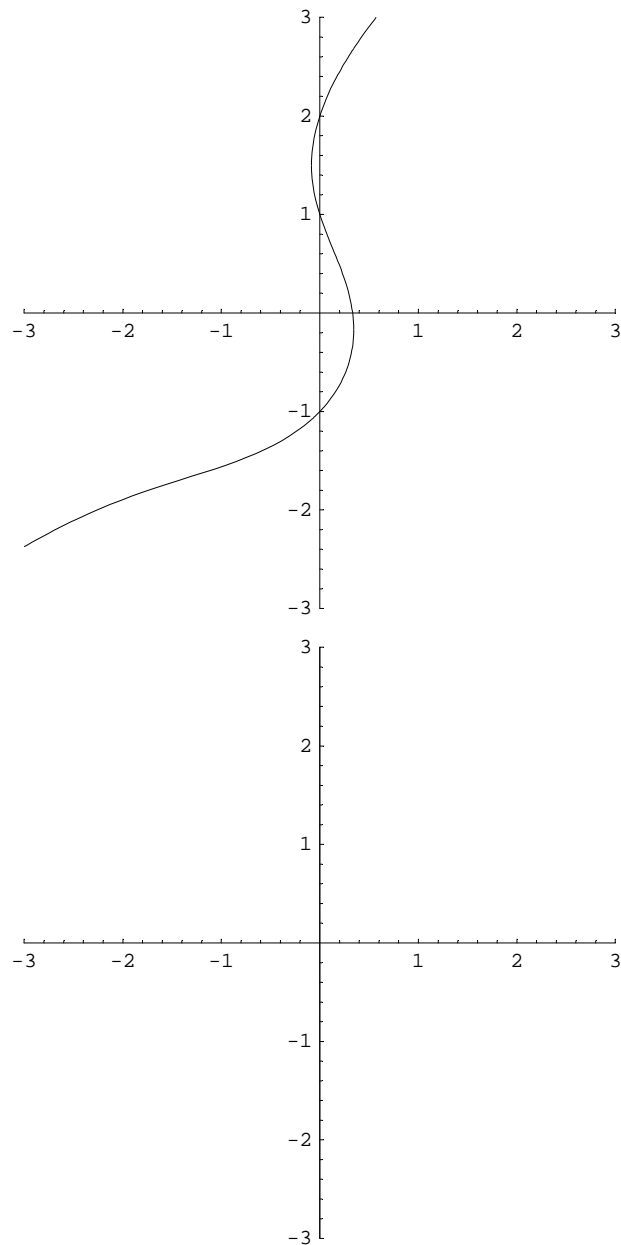
```
a=3;
(* n=128; *)
n=10;
tabelle1=Table[
  ImplicitPlot[(x-y+2)(x^2+y^2-1) ==
    N[3.(1.-Tan[b])] x, {x,-a,a},
    PlotRange->{{-a,a},{-a,a}}, AspectRatio->1.],
  {b,-Pi/2+Pi/n,Pi/2-Pi/n,Pi/n}];
Prepend[tabelle1,
  ParametricPlot[{0.,t},{t,-a,a},
    PlotRange->{{-a,a},{-a,a}}, AspectRatio->1.]];
```











Dann animieren!

■ Ensuite animer!

## 2. Beispiel:

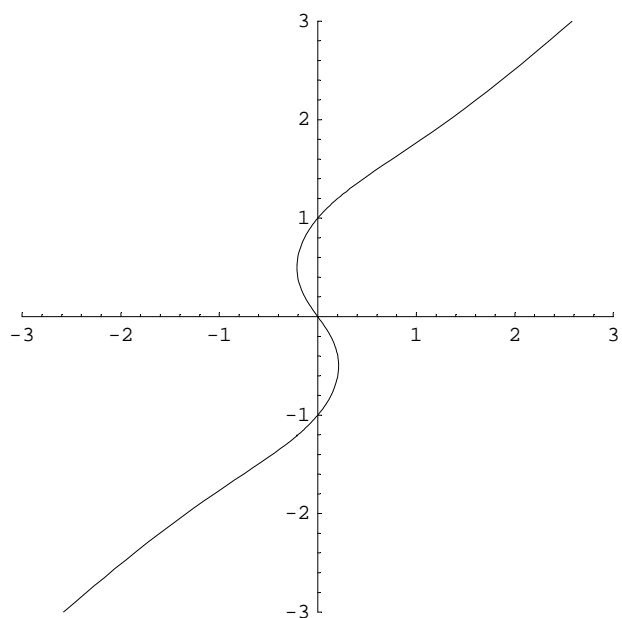
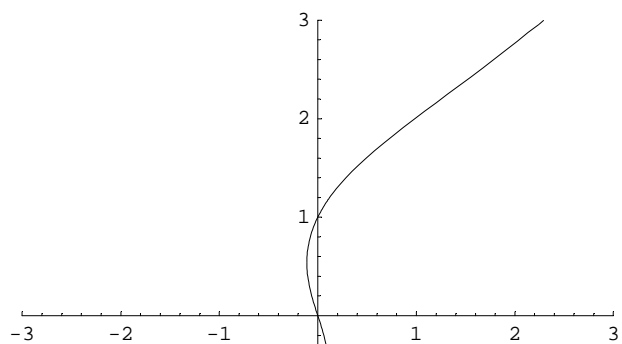
Zuerst  $n = 128$  Graphiken rechnen. (Um Ueberraschungen zu vermeiden, ist  $n$  auf 10 eingestellt. Das geht zwar schneller, aber die Animation wird so nicht gut.)

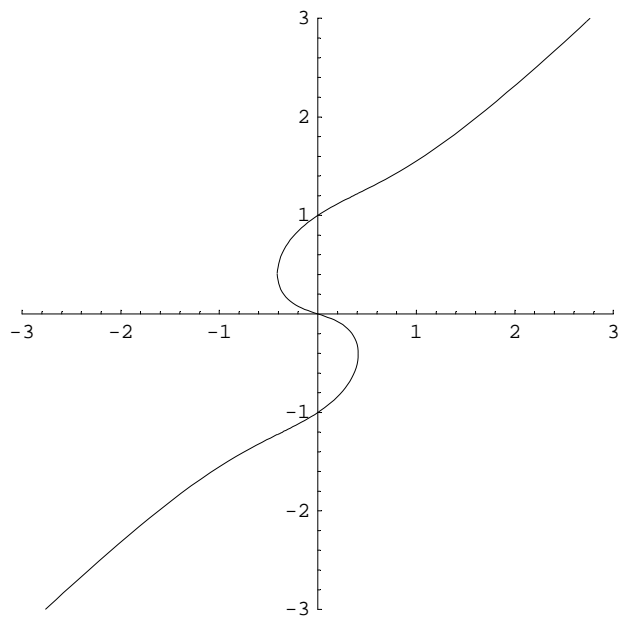
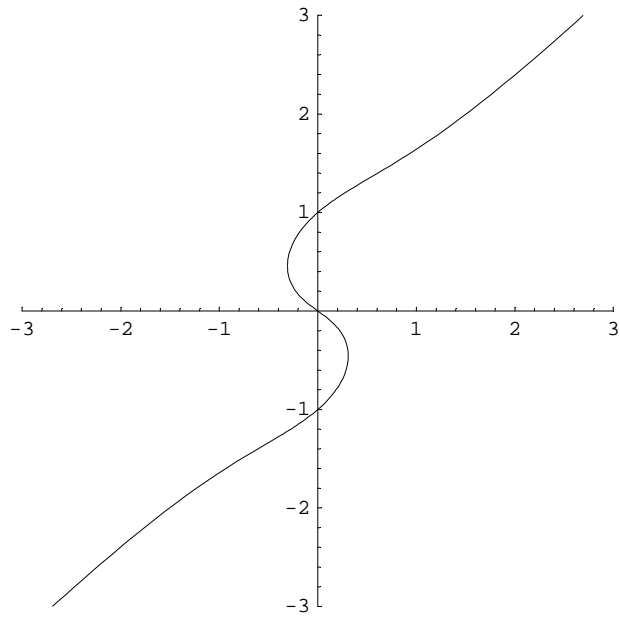
■ 2e exemple:

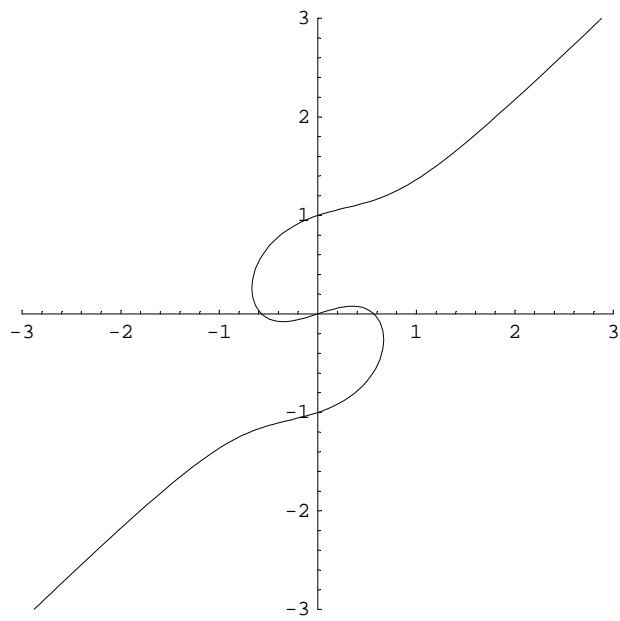
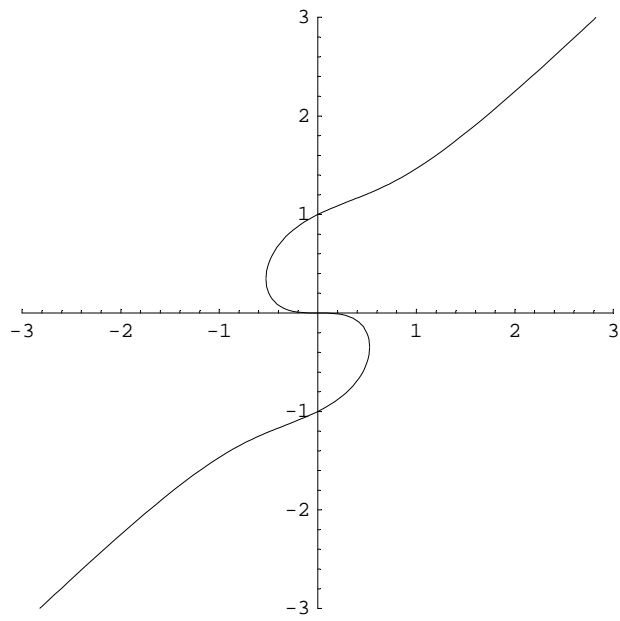
D'abord calculer  $n = 128$  graphiques. (Pour éviter des surprises, il faut mettre  $n$  sur 10. Cela va plus vite, mais l'animation n'est pas si bonne.)

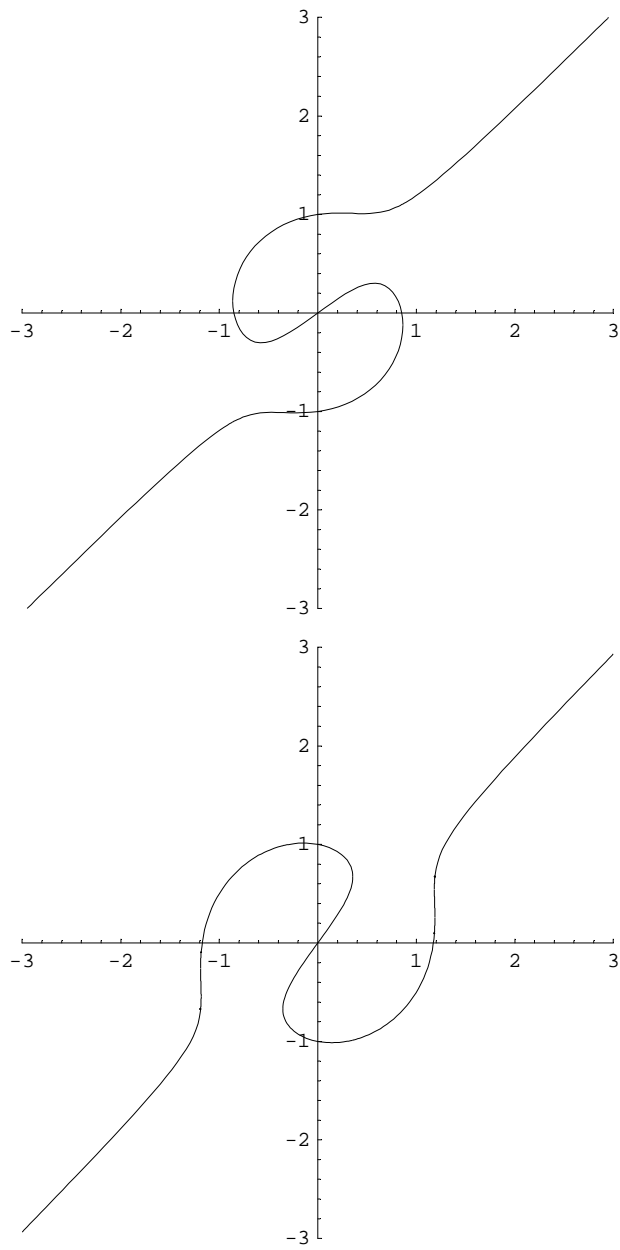
```
In[211]:=
```

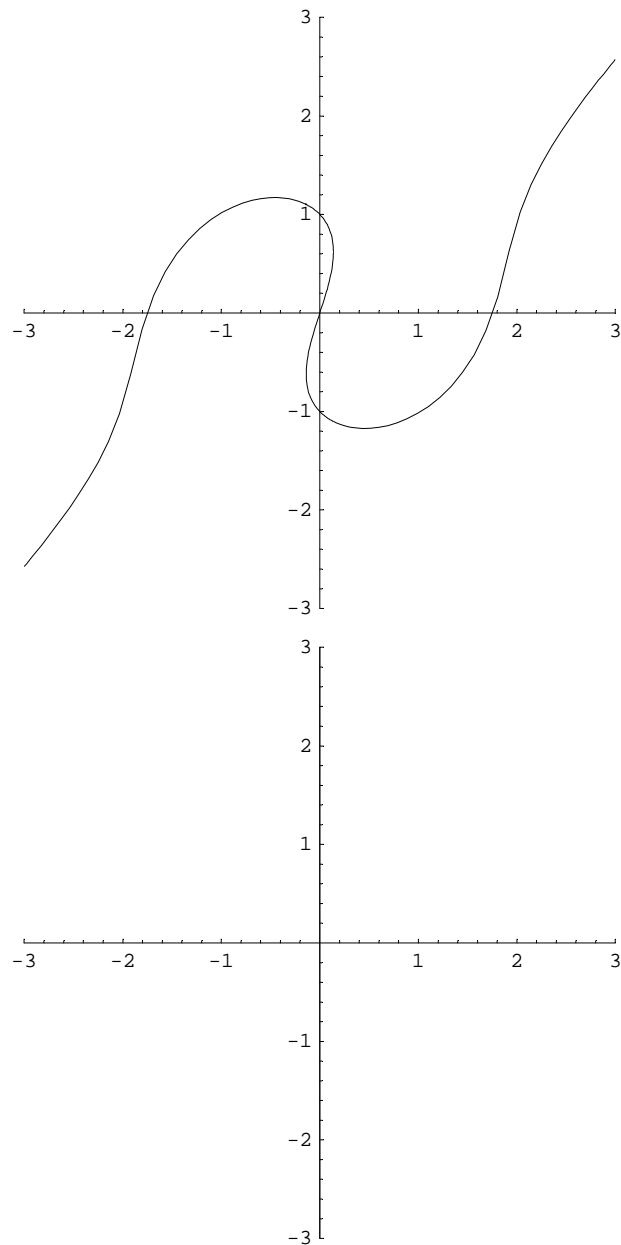
```
a=3;
(* n=128; *)
n=10;
tabelle2=Table[
  ImplicitPlot[(x-y)(x^2+y^2-1)==N[Tan[b]-1] x,
    {x,-a,a}, PlotRange->{{-a,a},{-a,a}},
    AspectRatio->1.],{b,-Pi/2+Pi/n,Pi/2-Pi/n,Pi/n}];
Prepend[tabelle2,
  ParametricPlot[{0.,t},{t,-a,a}, PlotRange->{{-a,a},
    {-a,a}},AspectRatio->1.]];
```











Dann animieren!

■ Ensuite animer!

#### 4.14.2. Noch ein Beispiel (Simulation einer Welle)

##### ■ Encore un exemple (simulation d'une onde)

**Erst rechnen, dann animieren!**

■ **D'abord calculer, ensuite animer!**

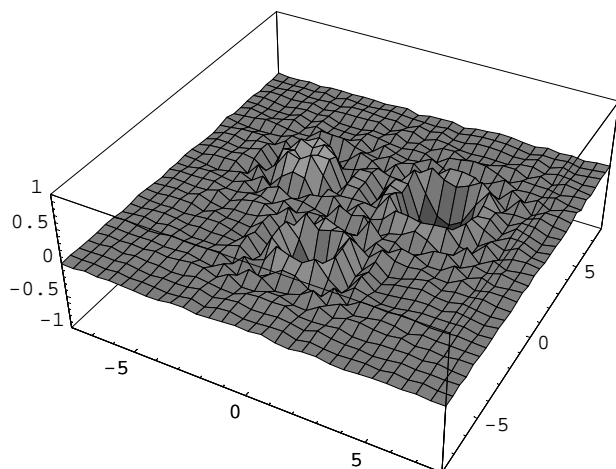
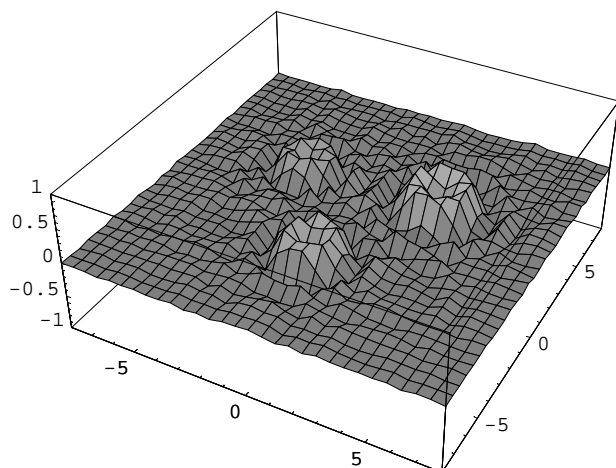
(Um Ueberraschungen zu vermeiden, sind statt 60 nur 5 Schritte eingestellt.

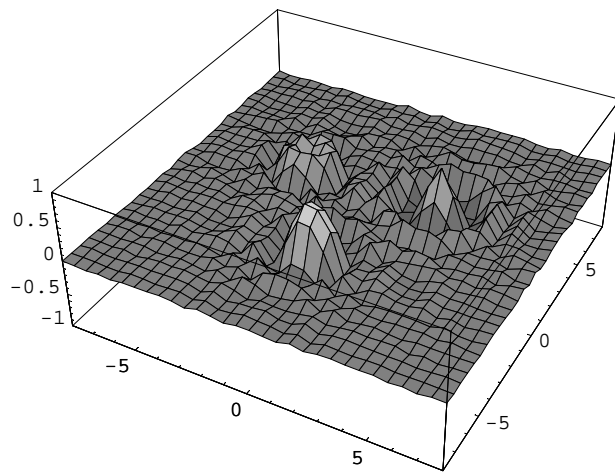
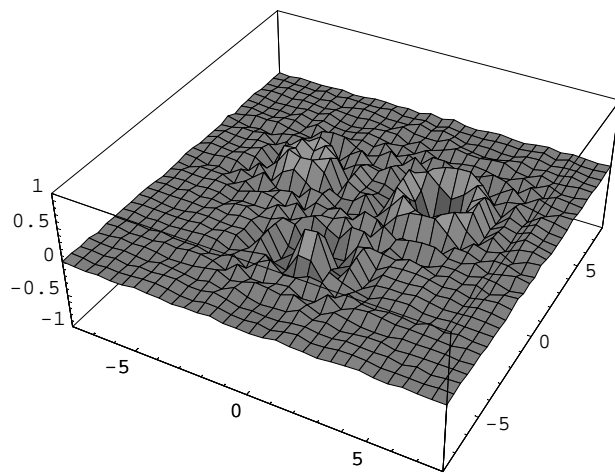
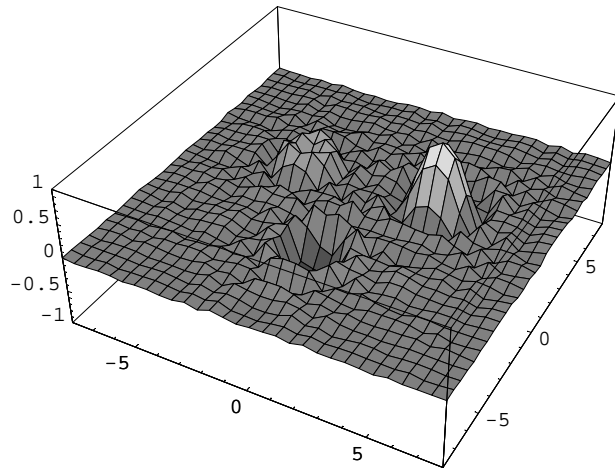
Das geht zwar schneller, aber die Animation wird so nicht gut.)

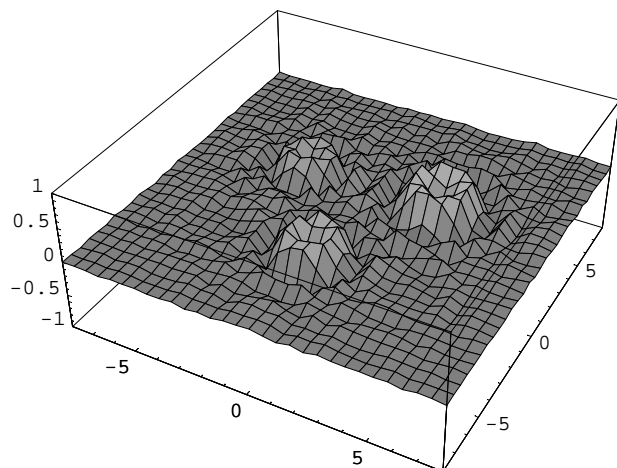
■ (Pour éviter des surprises, il faut 5 au lieu de 60 étapes. Cela va plus vite, mais l'animation réussit moins bien.)

In[216]:=

```
Table[Plot3D[
  Sin[3t+(x-3)^2+(y-2)^2]/(1+(x-3)^2+(y-2)^2)+
  3/5Sin[5/3(3t+(x+3)^2+(y-2)^2)]/(1+(x+3)^2+(y-2)^2)+
  3/4Sin[4/3(3t+x^2+(y+3)^2)]/(1+x^2+(y+3)^2),
  {x,-8,8},{y,-8,8},
  PlotRange->{{-8,8},{-8,8},{-1,1}},
  PlotPoints->30,
  (* {t,0,2Pi,2Pi/60}]; *)
  {t,0,2Pi,2Pi/5}];
```







Wähle die Zelle an, die alle ausgegebenen Bilder umschließt. Animiere mit "Animate Selected Graphics ...y" in Menu ("Graph") neu "Cell". Beobachte, welche Zonen sich bewegen und welche Zonen ruhig bleiben. Gibt es Interferenzen?

■ Choisir la cellule qui inclut toutes les images sorties. Anime par "Animate Selected Graphics ...y" dans le menu ("Graph") nouveau "Cell". Observe quelles zones restent tranquilles. Y a-t-il des interférences?

---

## 4.15. Probiere eigene Beispiele aus!

### ■ Essaie de propres exemples!

---

## "Putzmaschine" einsetzen

### ■ Employer la "machine de nettoyage"

```
In[217]:=
(* Old Form: Remove["Global`*"] *)
```

```
In[218]:=
Remove["Global`*"]
```