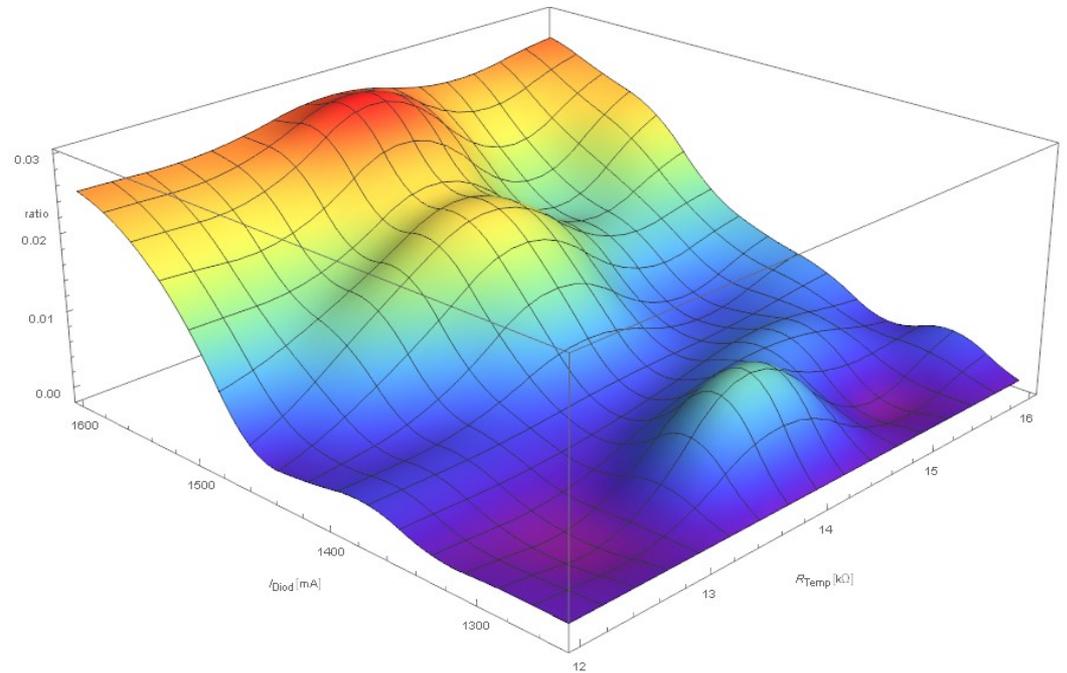


Eine Einführung in Wolfram Mathematica



- Programmübersicht
- Grundlegende Funktionen
- Listen, Vektoren und Matrizen
- Operationen auf Listen
- Plotten
- Plotten von Datenpunkten
- Fitten von Datenpunkten
- Plotten von Datenpunkten mit Funktionen
- Fehlerbalken
- 3D Plots
- Wolfram Hilfe

Bsp.nb * - Wolfram Mathematica 10.2

File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

```
In[90]= Needs["ErrorBarPlots`"]
er = Table[{i - 1, RandomReal[0.2] + 0.1*i^2, 0.1, Abs[i] * 0.05 * RandomReal[{0.9, 1.1}] + 0.1}, {i, -5, 10}];
err = Import["C:\\Users\\mephisto\\Desktop\\Bsp.csv"]
el = err /. {a_, b_, c_, d_} -> {{a, b}, ErrorBar[c, d]};
tofit = err /. {a_, b_, c_, d_} -> {a, b};
Quad[x_] := a + b * (x - c)^2
param = FindFit[tofit, Quad[x], {a, {b, 0.1}, c}, x];
nlm = NonlinearModelFit[tofit, Quad[x], {a, {b, 0.1}, c}, x]
nlm[""]

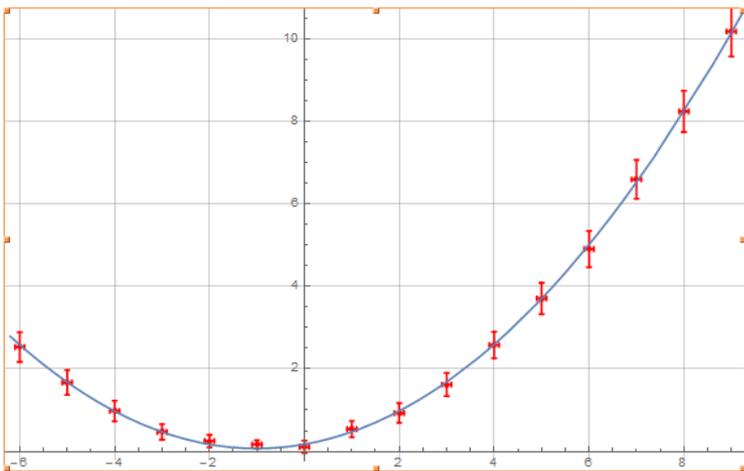
Show[ErrorListPlot[el, PlotStyle -> {Red, PointSize[Medium]}], Plot[Normal[nlm], {x, -6.2, 999.2}], GridLines -> Automatic]
```

Out[92]= {{-6, 2.53, 0.1, 0.36}, {-5, 1.67, 0.1, 0.3}, {-4, 0.98, 0.1, 0.25}, {-3, 0.47, 0.1, 0.19},
{-2, 0.25, 0.1, 0.15}, {-1, 0.17, 0.1, 0.1}, {0, 0.11, 0.1, 0.15}, {1, 0.54, 0.1, 0.2}, {2, 0.93, 0.1, 0.24}, {3, 1.62, 0.1, 0.28},
{4, 2.58, 0.1, 0.32}, {5, 3.71, 0.1, 0.38}, {6, 4.91, 0.1, 0.44}, {7, 6.6, 0.1, 0.47}, {8, 8.25, 0.1, 0.5}, {9, 10.19, 0.1, 0.61}}

Out[97]= FittedModel[$0.0699395 + 0.100793(1.00521 + x)^2$]

Out[98]= $0.0699395 + 0.100793(1.00521 + x)^2$

Out[99]=



labels... axes ▾ image size ▾ background ▾ more...   

100%

Enter erzeugt Zeilenumbruch
Umschalt+Enter startet Berechnung

Funktionen werden groß geschrieben und benutzen eckige Klammern „[]“

a+b

a*b

a-b

a/b

Sqrt[a]

Exp[a]

Log[a]

Log[b,a]

Sin[a]

ArcSin[a]

Pi

I

D[f,{x,n}]

Integrate[f,x]

Integrate[f,{x,0,10}]

Import["pfad/Datei"] importiert Daten

a + b

a • b

a – b

a ÷ b

\sqrt{a}

e^a

$\ln(a)$

$\log_b(a)$

$\sin(a)$

$\sin^{-1}(a)$

π

i

$\partial^n f / \partial x^n$

$\int f dx$

$\int_0^{10} f dx$

Mathematica arbeitet symbolisch, d.h. Variablen werden als Variablen behandelt, bis ihnen ein Wert zugewiesen wird
 \Rightarrow „D[x²,x] = 2x“

Funktionen können selbst definiert werden:

$F[x_]:=2a*x+\text{Cos}[x]$

„F[Pi]=2aπ-1“

„F[a]=2a²+Cos[a]“

Listen werden durch geschweifte Klammern eingefasst „{}“, Trennung durch Komma

Eine Liste kann Variablen, Zahlen, Funktionen, andere Listen... enthalten

$$\{a,b,c,d,e\} \quad \{1,2,3,4,5\} \quad \{1, 2x, 4x^2-2\} \quad \{\{a,b\},\{c,d\}\}$$

Eine Liste aus Listen stellt eine Matrix da

$$\{\{a, b\}, \{c, d\}\} \hat{=} \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Matrix-/Skalarprodukt: „. „

VORSICHT: $\{a,b\}*\{c,d\}=\{ac,bd\}$

Normale Rechenzeichen wirken auf Listeneinträge

$$\{a,b\}.\{c,d\} = ac+bd,$$

$$\{\{ab\},\{cd\}\}.\{\{w,x\},\{y,z\}\} = \{\{a w+b y, a x+b z\},\{c w+d y, c x+d z\}\}$$

Dyadisches Produkt: Outer[Times,x,y]

$$\text{„Outer[Times,\{a,b\},\{c,d\}] = \{\{a c,a d\},\{b c,b d\}\}“}$$

Kreuzprodukt: Cross[x,y]

Einige interessante Optionen für Listen

Um bestimmte Regeln auf alle Elemente einer liste an zu wenden: *list /. rules*

„{x,y,2x,3}/. x->2 = {2,y,4,3}“

„{x,y,2x,3}/. x->{a,b} = {{a,b},y,{2a,2b},3}“

„{{a,b,c},{d,e,f}}/.{x_,y_,z_}->{2x,y*z,z}={{2a,bc,c},{2d,ef,f}}“

Um eine Liste nach bestimmten Regeln zu erstellen: *Table[expr,{i,i_{min},i_{max}}]*

Funktioniert wie eine for Schleife, die ein neues Element in jedem Durchlauf anlegt

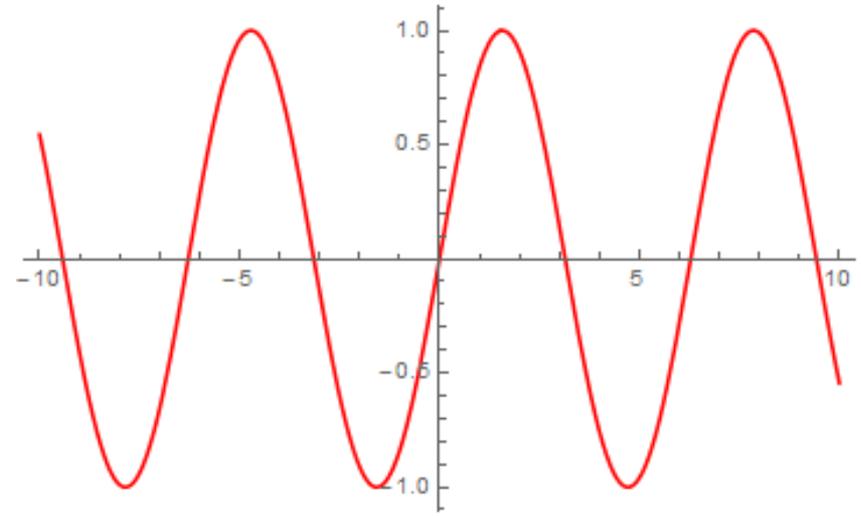
„Table[i,{i,1,5}] = {1,2,3,4,5}“

„Table[2i+1,{i,1,5}] = {3,5,7,9,11}“

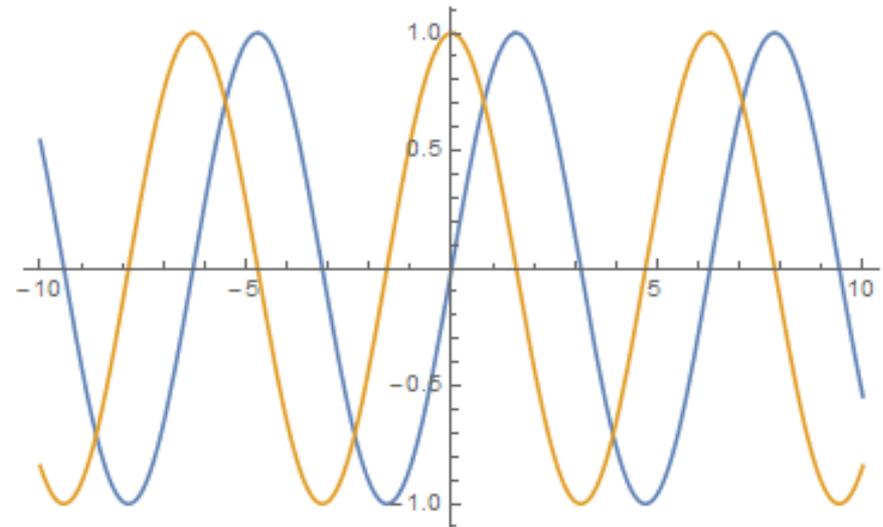
„Table[D[Sin[x],{x,i}],{i,0,4}] = {Sin[x],Cos[x],-Sin[x],-Cos[x],Sin[x]}“

Zum Plotten benutze die Funktion: $\text{Plot}[\text{function}, \{x, x_{\min}, x_{\max}\}]$

$\text{Plot}[\text{Sin}[x], \{x, -10, 10\}, \text{PlotStyle} \rightarrow \text{Red}]$



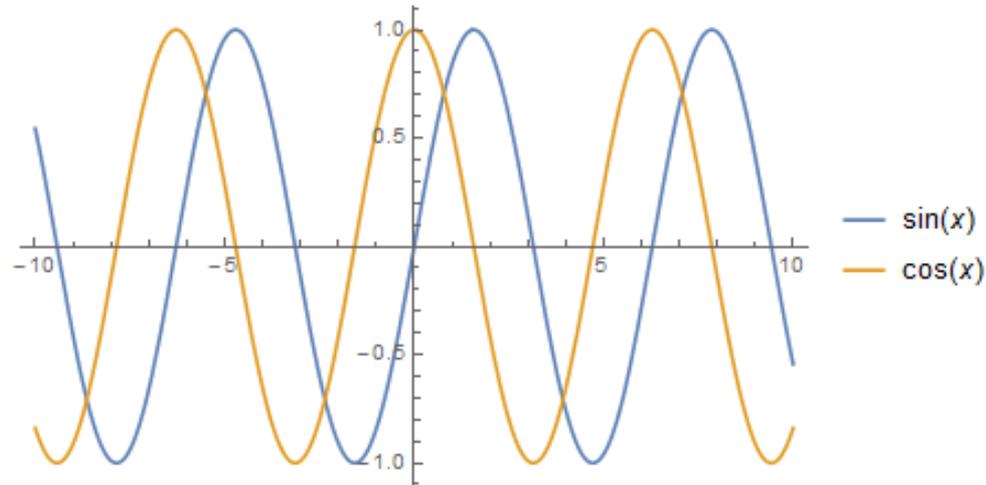
$\text{Plot}[\{\text{Sin}[x], \text{Cos}[x]\}, \{x, -10, 10\}]$



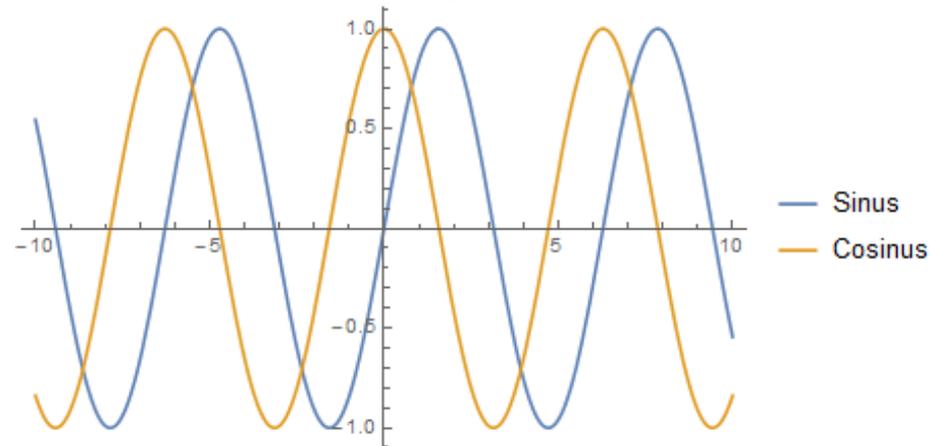
Plots können einfach via
„Rechtsklick → save as“
exportiert werden

Legenden anlegen

```
Plot[{Sin[x],Cos[x]},{x,-10,10},PlotLegends -> "Expressions"]
```

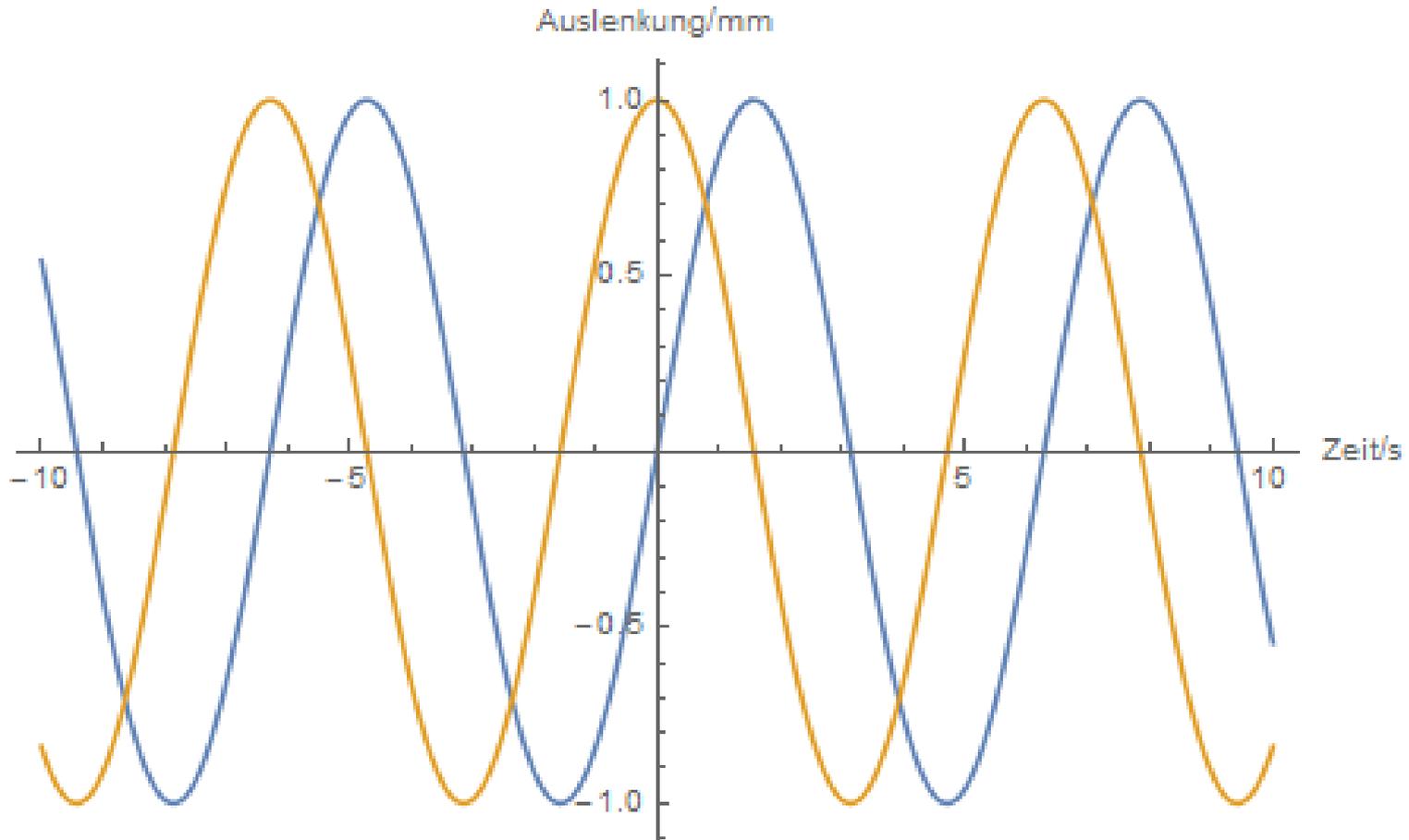


```
Plot[{Sin[x],Cos[x]},{x,-10,10},PlotLegends -> {"Sinus","Cosinus"}]
```



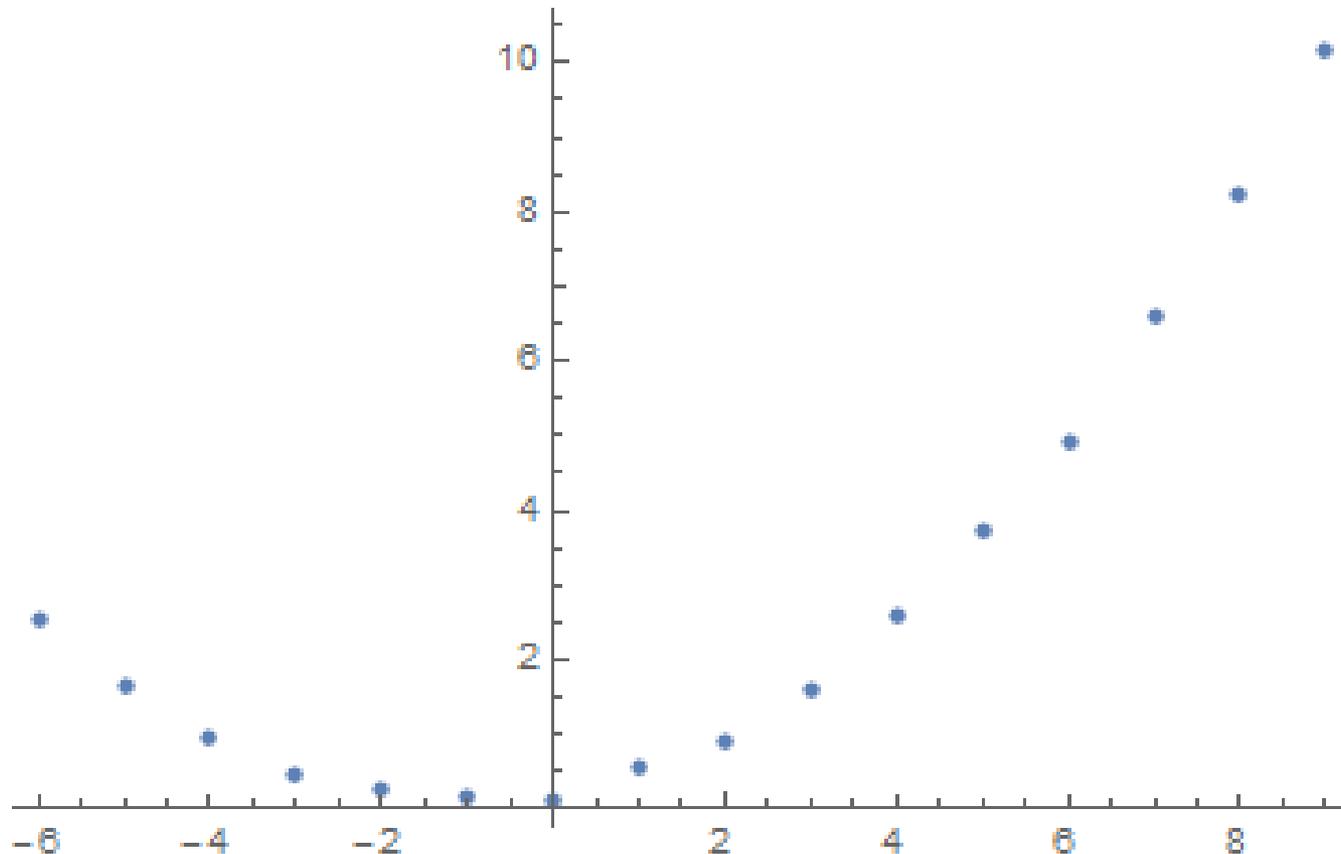
Achsenbeschriftungen anlegen

```
Plot[{Sin[x],Cos[x]},{x,-10,10} , AxesLabel->{"Zeit/s","Auslenkung/mm"}]
```



Liste aus Messpunkten: `mess = {{-6,2.53},{-5,1.67},{-4,0.98},{-3,0.47},{-2,0.25},{-1,0.17},{0,0.11},{1,0.54},{2,0.93},
{3,1.62},{4,2.58},{5,3.71},{6,4.91},{7,6.6},{8,8.25},{9,10.19}}`

`ListPlot[mess]`



Optionen analog zu „Plot[]“

Das Fitten von Funktionen an Datenpunkte

`NonlinearModelFit[data,function,{a,b...},x]`

Bsp: `Quad[x_]:=a+b*(x-c)^2`

`nlm=NonlinearModelFit[mess, Quad[x], {a,b,c}, x]`

`FittedModel [0.0699395 + 0.100793 (1.00521 + x)2]`

Um die reine Fkt zu bekommen: `nlm[" "]` oder `Normal[nlm]`
`0.0699295+0.100793(1.00521+x)2`

Für die Ausgabe der Fitparameter: `nlm["BestFitParameters"]`
`{a→0.0699395,b→0.100793,c→-1.00521}`

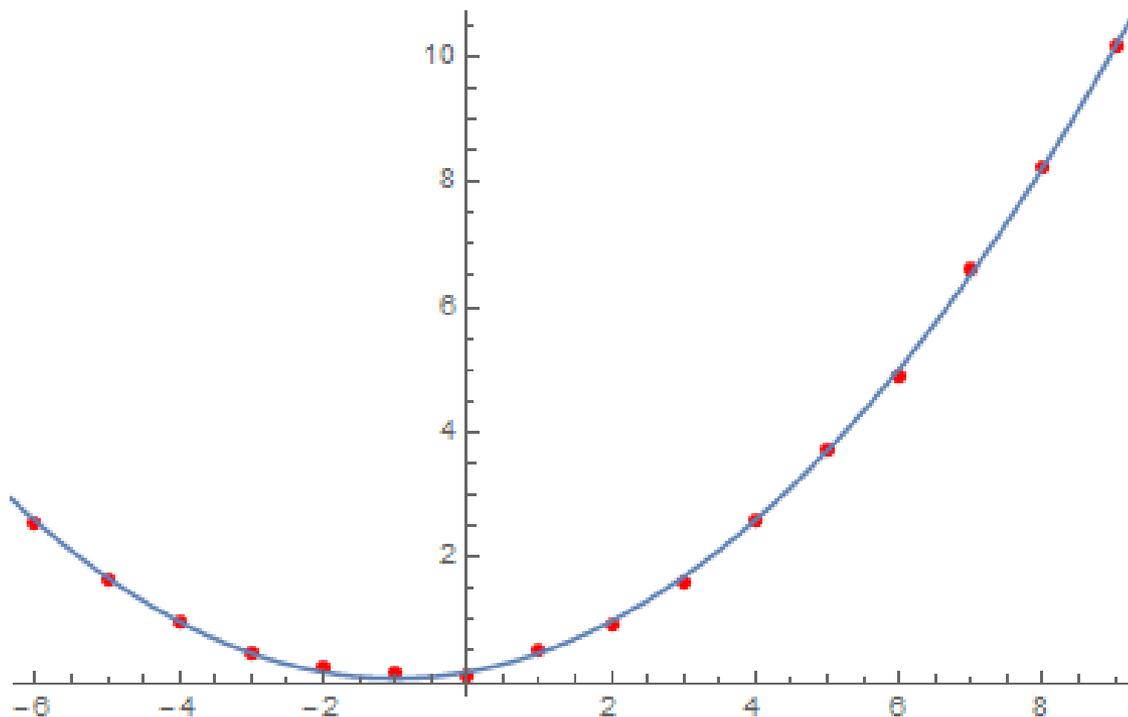
Für Standardabweichung etc.: `nlm["ParameterTable"]`

	Estimate	Standard Error	t-Statistic	P-Value
a	0.0699395	0.0219391	3.18789	0.00713192
b	0.100793	0.000837238	120.388	3.36771×10^{-21}
c	-1.00521	0.0268855	-37.3885	1.2802×10^{-14}

Um mehrere Grafiken zusammen darzustellen wird `Show[graphics,options]` verwendet

Bsp: Plote die Punkte mess mit dem Plot der Fitfunktion:

```
Show[ ListPlot[mess, PlotStyle->Red] , Plot[ Normal[nlm] , {x,-11,11} ] ]
```



Beachte: die erste Grafik/Plot legt die Ausdehnung des Plots fest (vgl. die Optionen `PlotRange`, `DataRange`)

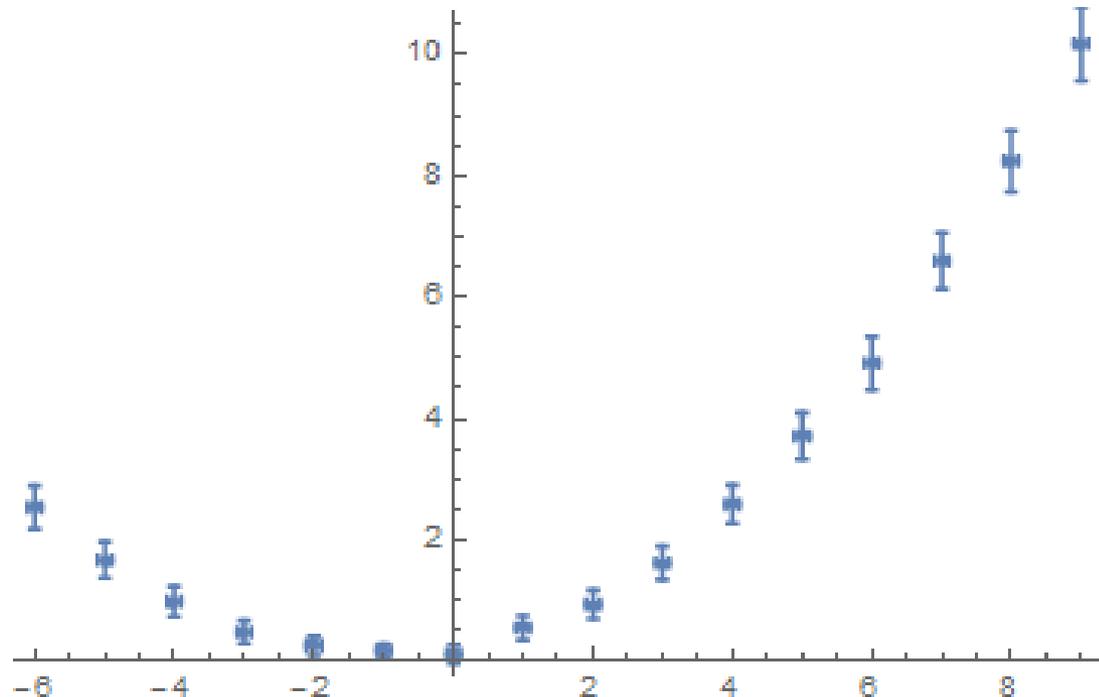
Um Fehlerbalken dar zu stellen wird "ErrorBarPlots`" benötigt

Sei mess eine Liste aus Messpunkten der Form: $\{\{x_1, y_1, \Delta x_1, \Delta y_1\}, \{x_2, y_2, \Delta x_2, \Delta y_2\}, \dots\}$

Needs["ErrorBarPlots`"]

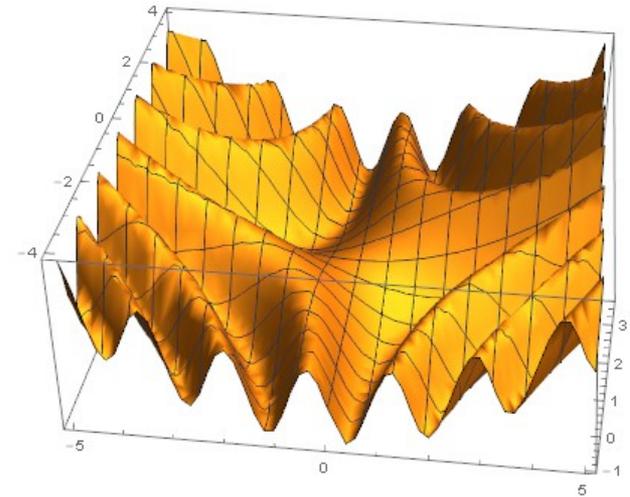
errorlist=mess/.{a_,b_,c_,d_}→{{a,b},ErrorBar[c,d]}

ErrorListPlot[errorlist]

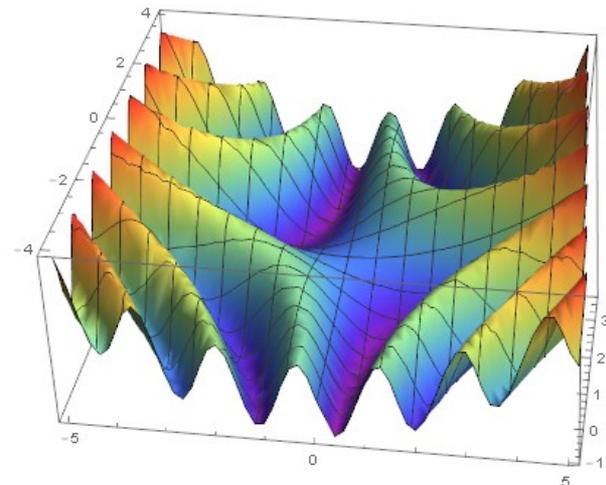


3D Plots funktionieren wie 2d Plots mit der Funktion $\text{Plot3D}[f, \{x, x_{\min}, x_{\max}\}, \{y, y_{\min}, y_{\max}\}]$

$\text{Plot3D}[\text{Sin}[x*y]+x^2/10, \{x, -5, 5\}, \{y, -4, 4\}]$



$\text{Plot3D}[\text{Sin}[x*y]+x^2/10, \{x, -5, 5\}, \{y, -4, 4\}, \text{ColorFunction} \rightarrow \text{"Rainbow"}]$



Weitere Infos unter:

<http://reference.wolfram.com/language/>

Bietet hilfreiche Dokumentationen zu jeder implementierten Funktion mit vielen Beispielen und Optionen

Mathematica lässt sich durch ein vorangestelltes „`==`“ auch mit Wolfram Alpha Syntax bedienen