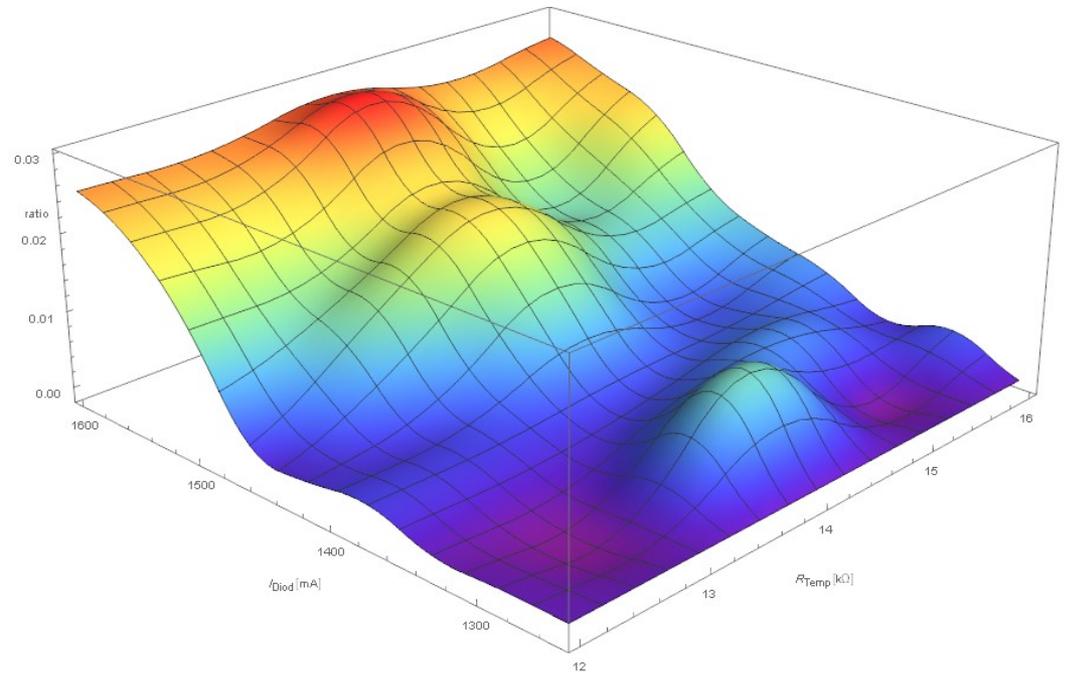


Eine Einführung in Wolfram Mathematica



- 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[""]

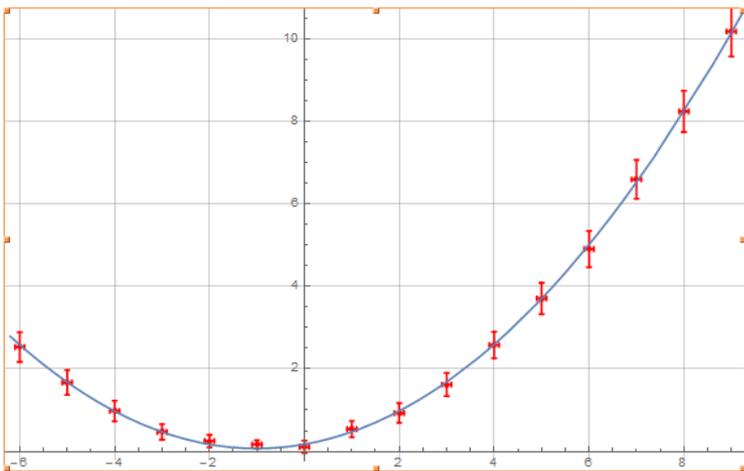
Show[ErrorListPlot[el, PlotStyle -> {Red, PointSize[Medium]}], Plot[Normal[nlm], {x, -6.2, 999.2}], GridLines -> Automatic]
```

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,o,10}]`

`Import["pfad/Datei"]` importiert Daten

$a + b$

$a \cdot b$

$a - b$

$a \div b$

\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 „ $D[x^2,x] = 2x$ “

Funktionen können selbst definiert werden:

`F[x_]:=2a*x+Cos[x]`

„ $F[\text{Pi}] = 2a\pi - 1$ “

„ $F[a] = 2a^2 + \text{Cos}[a]$ “

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

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

$$\{\{a,b\},\{c,d\}\}.\{\{w,x\},\{y,z\}\} = \{\{aw+by, ax+bz\},\{cw+dy, cx+dz\}\}$$

Dyadisches Produkt: Outer[Times,x,y]

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

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

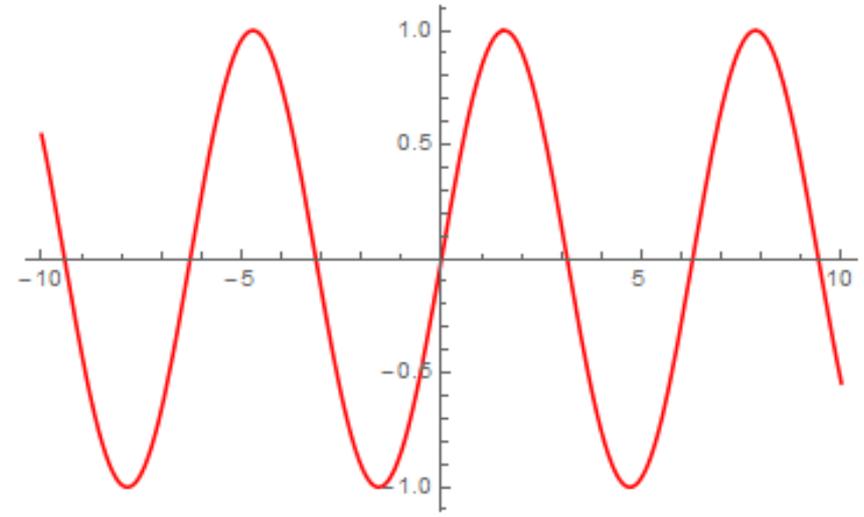
„Table[i,{i,1,5}] = {1,2,3,4,5}“
 „Table[2i+1,{i,1,5}] = {3,5,7,9,11}“
 „Table[D[Sin[x],{x,i}],{i,0,4}] = {Sin[x],Cos[x],-Sin[x],-Cos[x],Sin[x]}“

Um den n-ten Eintrag einer liste zu erhalten: *list[[n]]*

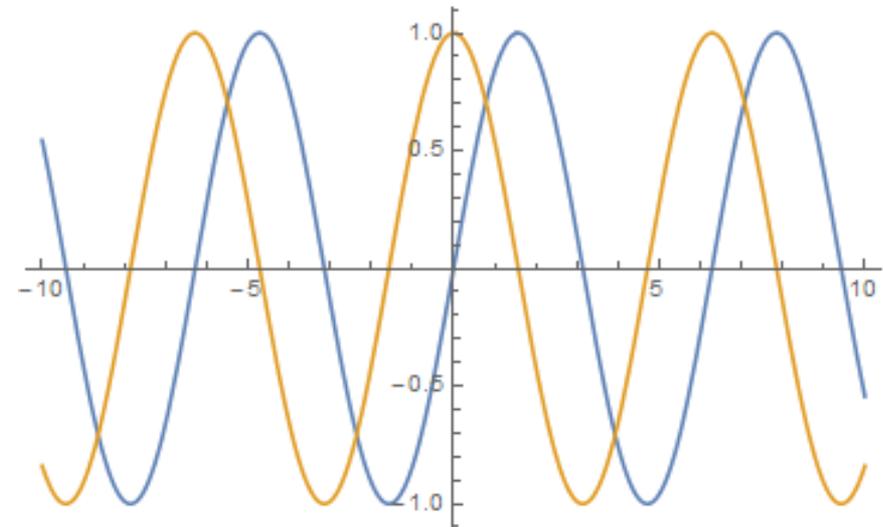
list={2,A,17,k}
 „List[[3]] =17“

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

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



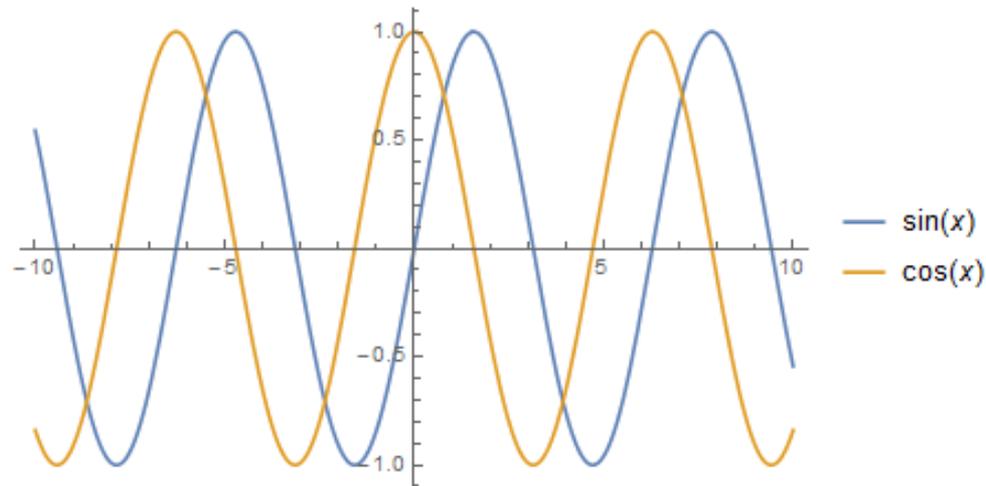
$\text{Plot}[\{\text{Sin}[x], \text{Cos}[x]\}, \{x, -10, 10\}]$



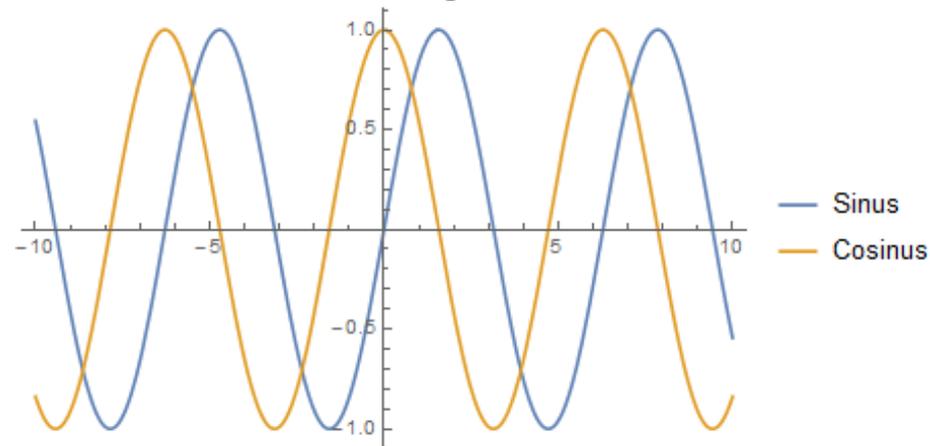
Plots können einfach via
„Rechtsklick → save as“
exportiert werden

Legenden anlegen

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

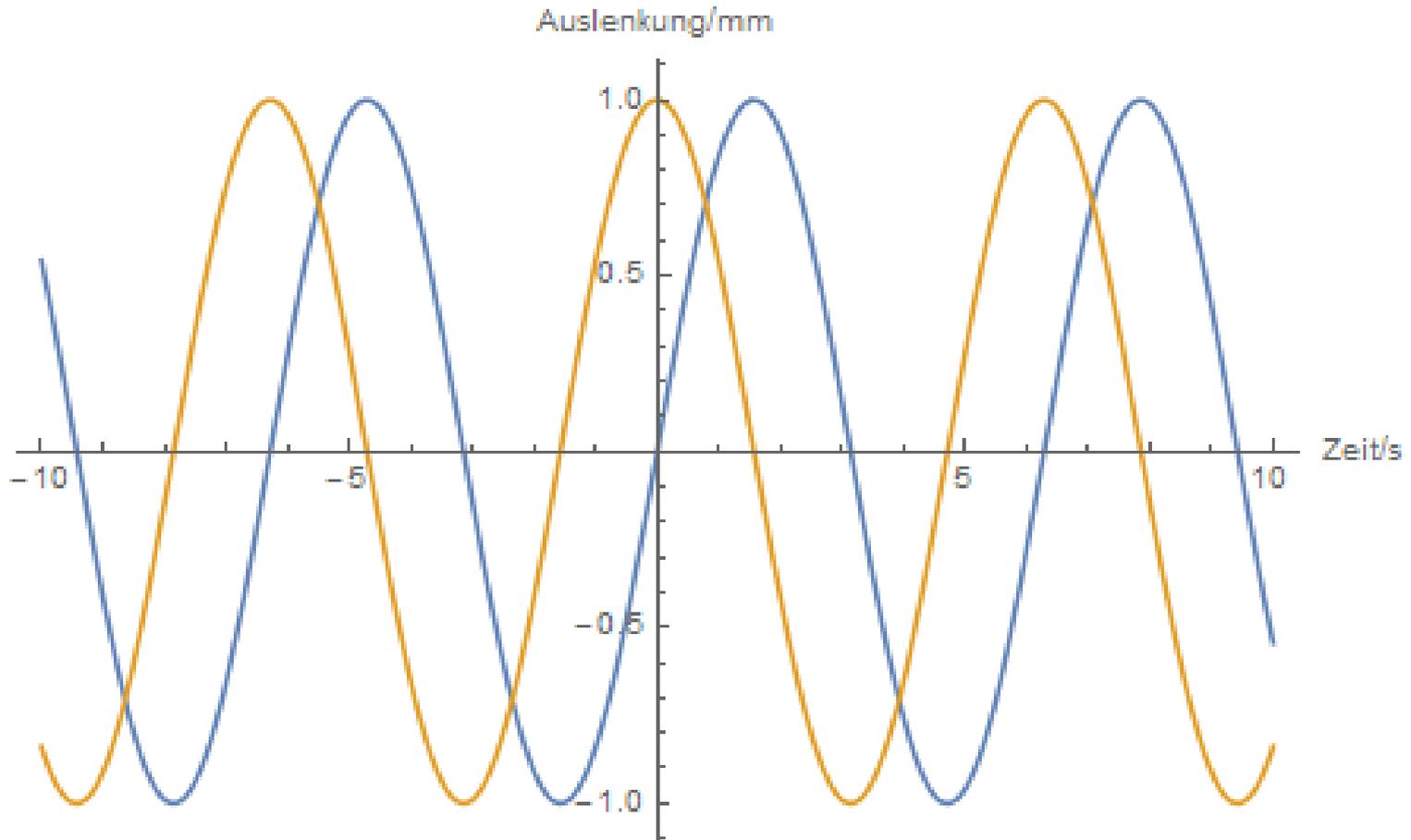


```
Plot[{Sin[x],Cos[x]},{x,-10,10},PlotLegends -> {"Sinus","Cosinus"}]
```



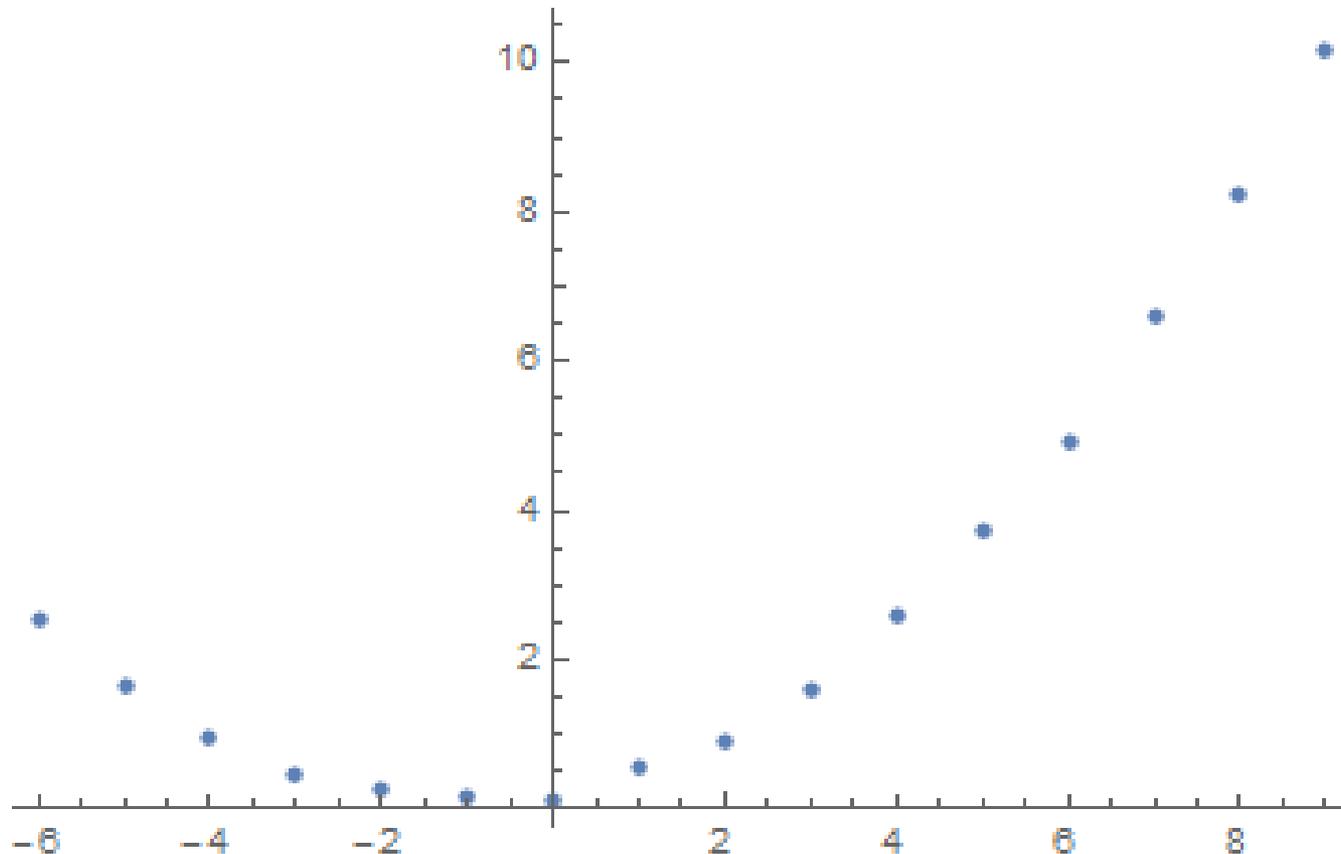
Achsenbeschriftungen anlegen

```
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}`

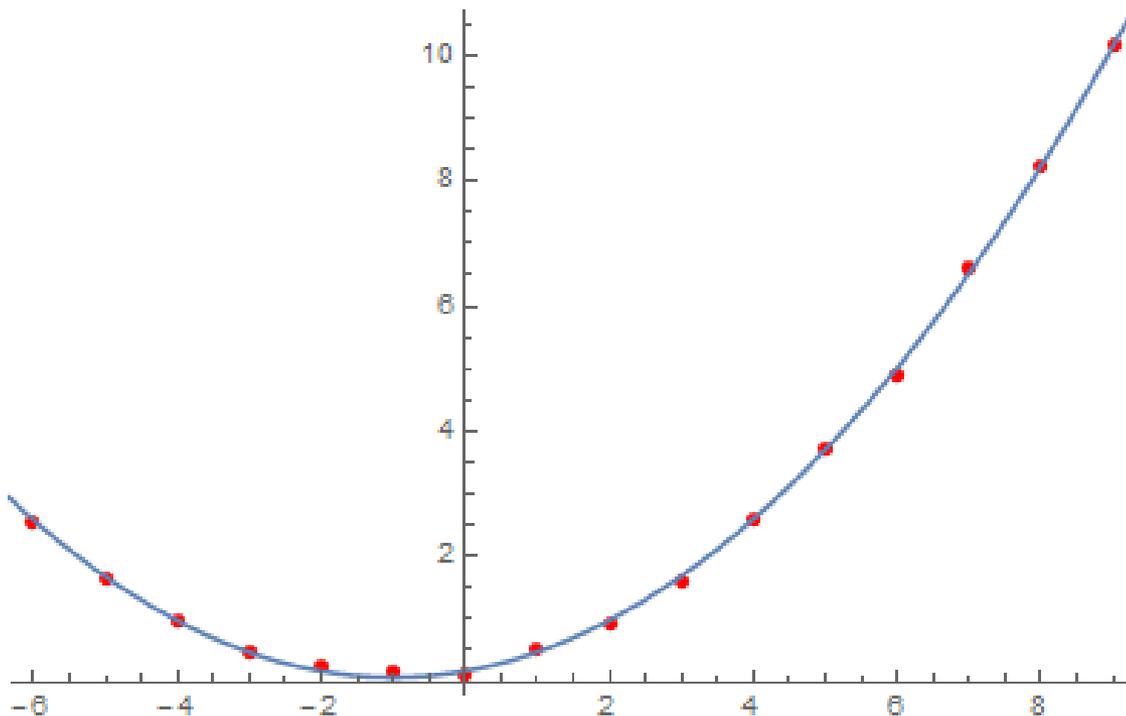
Für Standardabweichung etc.: `nlm["ParameterTable"]`

	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`)

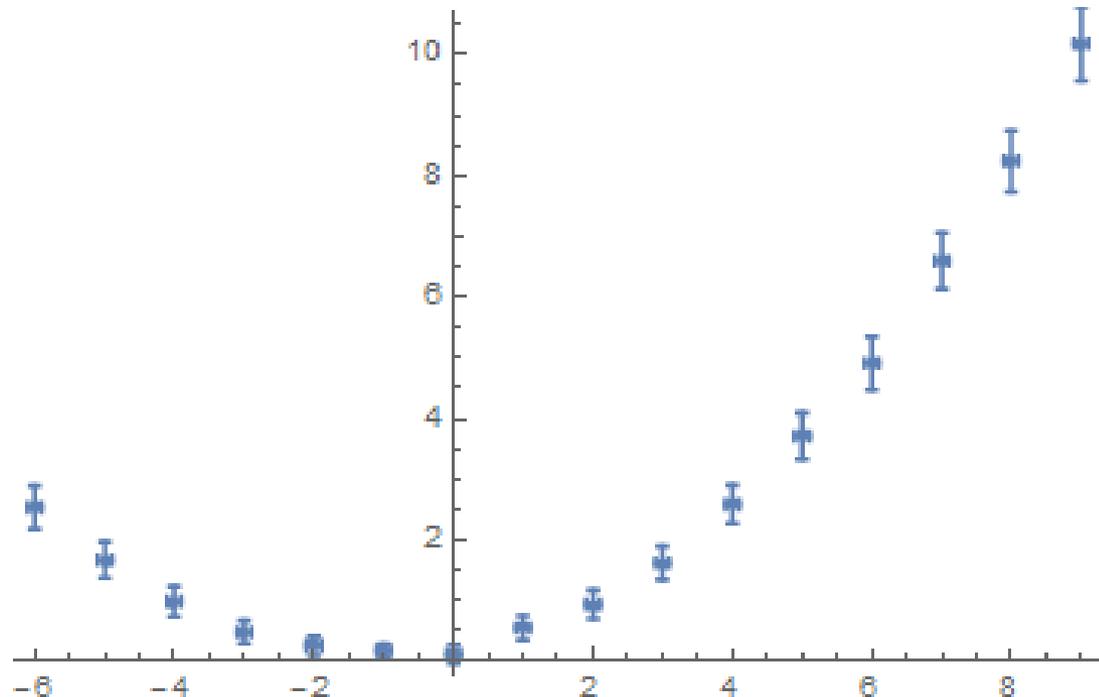
Um Fehlerbalken dar zu stellen wird "ErrorBarPlots`" benötigt

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`"]
```

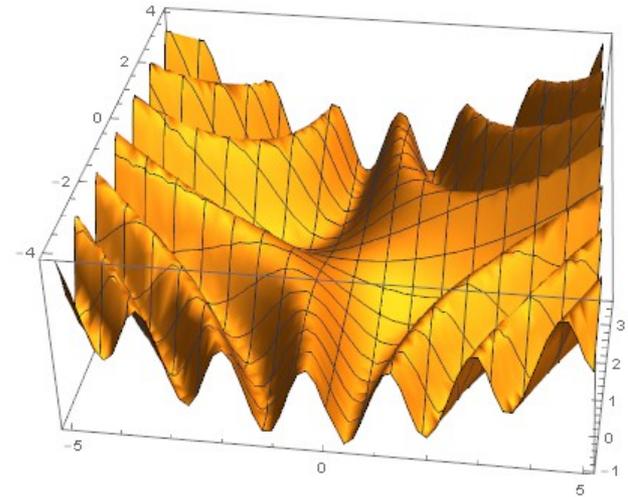
```
errorlist=mess/.{a_,b_,c_,d_}→{{a,b},ErrorBar[c,d]}
```

```
ErrorListPlot[errorlist]
```

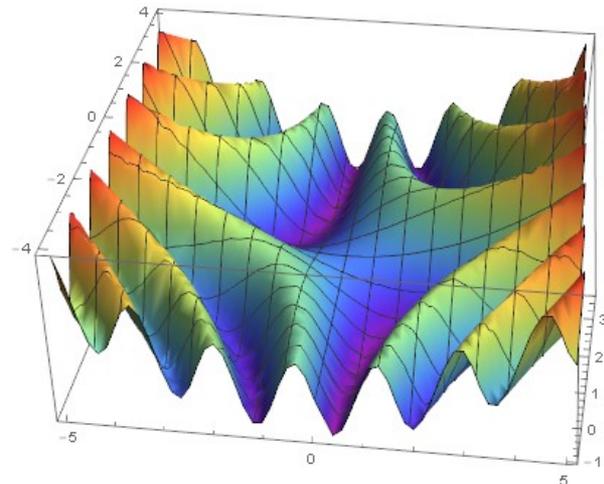


3D Plots funktionieren wie 2d Plots mit der Funktion $\text{Plot3D}[f, \{x, x_{\min}, x_{\max}\}, \{y, y_{\min}, y_{\max}\}]$

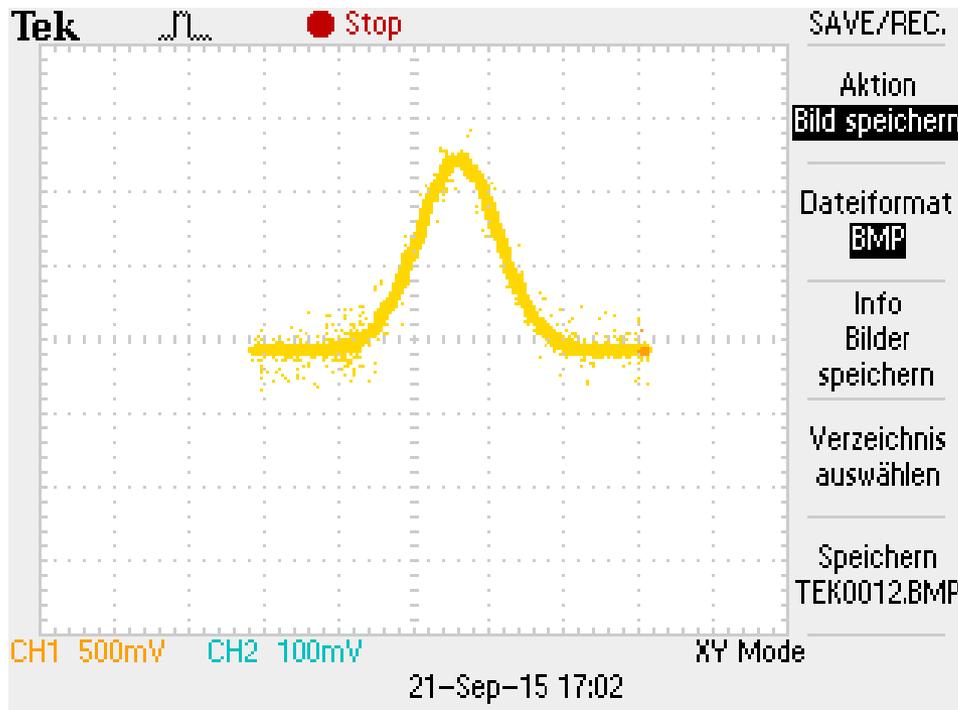
$\text{Plot3D}[\text{Sin}[x*y]+x^2/10, \{x, -5, 5\}, \{y, -4, 4\}]$



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



Mir sind in einer Messung die Daten des Oszilloskops kaputt gegangen, sodass ich nur noch die Bilder hatte. Ich musste die Breite der Gaußkurven bestimmen um daraus die Winkeldivergenz eines Lasers zu berechnen.



Idee:

suche alle gelben Pixel und fitte eine Gaußkurve hindurch.

Durchführung:

Lese das Bild als Farbmatrix in Mathematica ein und suche alle Positionen der Pixel mit einem bestimmten Farbwert
 → diese Tupel sind nur Datenwerte, die gefittet werden können

Winkeldivergenz_ss.nb - Wolfram Mathematica 10.2

File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

```
SetDirectory["/media/DataVolume/Data/Uni/FP-I/Laser/sphaerisch-sphaerisch"];
```

```
(*Hole die Bilddateien*)
```

```
pics = Table[Import["BILD" <> ToString[j] <> ".BMP"], {j, 101, 121}];
```

```
(*lese anhand der Farbwerte die Position der Messpunkte aus*)
```

```
data = Table[Position[Transpose[ImageData[pics[[j]], "Byte"], {255, 215, 0}], {j, 1, 21}];
```

```
(*Fitte alle Funktionen mit Gaußkurven*)
```

```
parameter = Table[FindFit[data[[j]], a + b*Exp[-((x - μ)^2) / (2*σ^2)] / Sqrt[2*Pi*σ^2], {{a, 120}, {σ, 1.5}, {μ, 142}, b}, x], {j, 1, 21}];
```

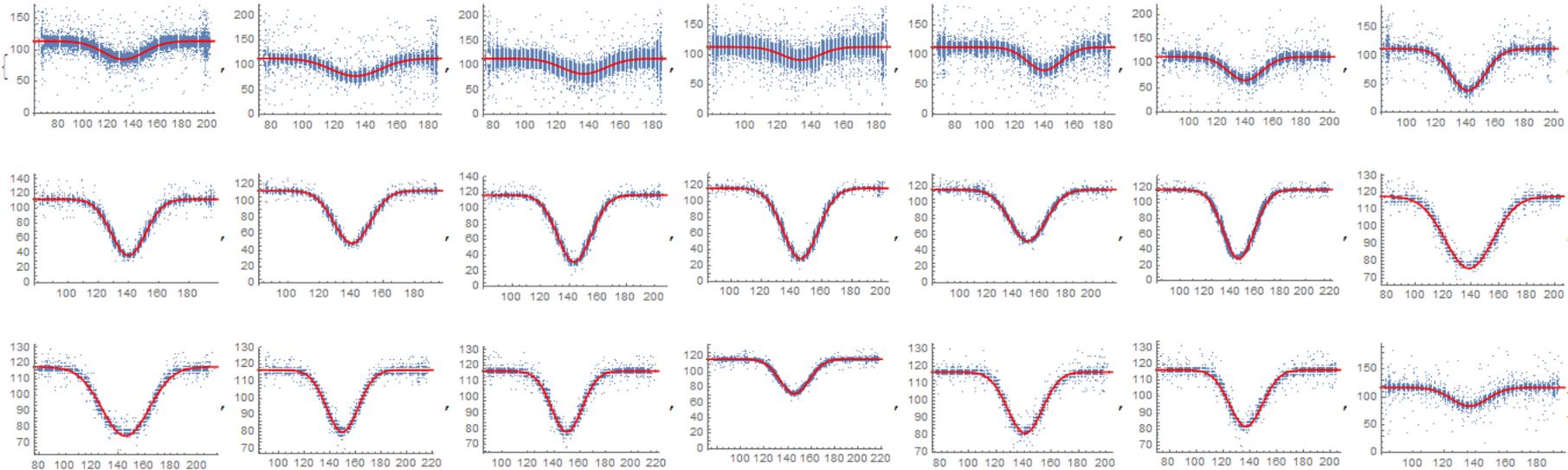
```
(*grafischer Vergleich des Fits mit Messpunkten*)
```

```
Table[Show[ListPlot[data[[j]], Plot[a + b*Exp[-((x - μ)^2) / (2*σ^2)] / Sqrt[2*Pi*σ^2] /. parameter[[j]], {x, 0, 300}, PlotStyle -> Red, PlotRange -> All]], {j, 1, 21}]
```

```
(*lese nun die σ aus und transformiere aus px in mm (aus der Justage des potentiometers folgt: 50px ^= 6,41mm*)
```

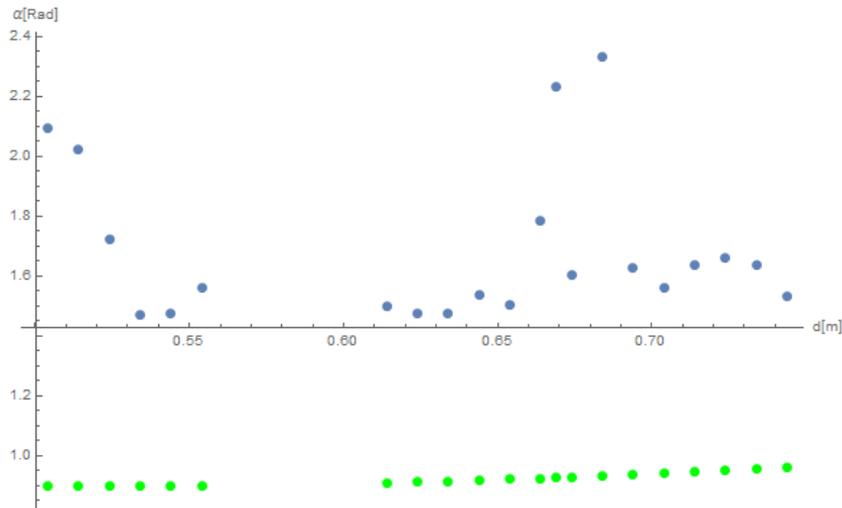
```
dist = Table[Import["intensitaet"]][[j]][[1]], {j, 1, 21}];
```

```
abw = Table[{(145.7 - 2.3 - dist[[j]]) * 10^-2, Abs[σ] * 2 * (0.641 / 50) * Sqrt[2] * 10^-2 /. parameter[[j]][[2]]}, {j, 1, 21}];
```



```
(*die Winkeldivergenzen, fehlerrechnung und theoriwerte*)
div = Table[1000*ArcTan[abw[[j]][[2]] / (2*((dist[[j]] + 151.5 + 15.5)*10^-2 + (abw[[j]][[1]]/2))], {j, 1, 21}];
δdiv = Table[1000* Sqrt[(5*10^-4 / (2*((dist[[j]] + 151.5 + 15.5)*10^-2 + (abw[[j]][[1]]/2) + (1 + (abw[[j]][[2]] / (2*((dist[[j]] + 151.5 + 15.5)*10^-2 + (abw[[j]][[1]]/2)))^2)))^2 +
(8*10^-3*abw[[j]][[2]] / ((2*((dist[[j]] + 151.5 + 15.5)*10^-2 + (abw[[j]][[1]]/2))^2 * (1 + (abw[[j]][[2]] / (2*((dist[[j]] + 151.5 + 15.5)*10^-2 + (abw[[j]][[1]]/2)))^2))^2)], {j, 1, 21}];
theodiv = Table[1000*Sqrt[2*632.8*10^-9/Pi]*(2*(abw[[j]][[1]])*0.5 - (abw[[j]][[1]])^2)^(-1/4), {j, 1, 21}];
gera = FindFit[Table[{abw[[j]][[1]], div[[j]]}, {j, 1, 21}], c+d*x, {c, d}, x];
Show[ListPlot[Table[{abw[[j]][[1]], div[[j]]}, {j, 1, 21}], ListPlot[Table[{abw[[j]][[1]], theodiv[[j]]}, {j, 1, 21}], PlotStyle -> Green] (*,
Plot[c+d*x/.gera, {x, 0, 1}, PlotStyle -> Red, PlotRange -> All] *), PlotRange -> All, AxesLabel -> {"d[m]", "α[Rad]"}];

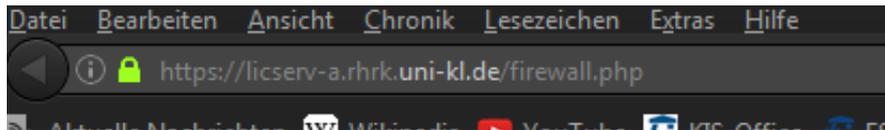
(*tabellarische Aufbereitung der Daten*)
tab = Table[{abw[[j]][[1]], abw[[j]][[2]], div[[j]], δdiv[[j]], theodiv[[j]]}, {j, 1, 21}];
Grid[Prepend[tab, {"d[m]", "w[m]", "α[mRad]", "δα[mRad]", "αTheo[mRad]"}], Frame -> {All, 1 -> True}, Background -> {None, {{White, LightGray}}}]
```



d [m]	w [m]	α [mRad]	δα [mRad]	α _{Theo} [mRad]
0.504	0.0119507	2.09514	0.0746402	0.897626
0.514	0.0115144	2.02219	0.0747477	0.897787
0.524	0.00977877	1.7204	0.0748447	0.898129
0.534	0.00833174	1.46841	0.0749462	0.898652
0.544	0.00834314	1.47301	0.0750588	0.899358
0.554	0.00881459	1.559	0.0751751	0.900248
0.614	0.00838615	1.49913	0.0758568	0.909671
0.624	0.00822296	1.47259	0.075971	0.911969
0.634	0.0082273	1.47601	0.0760867	0.914494
0.644	0.00853698	1.53432	0.076205	0.917253
0.654	0.0083495	1.50333	0.07632	0.920258
0.664	0.00989918	1.78556	0.0764493	0.92352
0.674	0.00886732	1.60233	0.0765578	0.927052
0.669	0.012356	2.23072	0.0765321	0.925251
0.684	0.0128855	2.33263	0.0767145	0.930867
0.694	0.00896169	1.62526	0.0767939	0.934982
0.704	0.00858002	1.55887	0.0769091	0.939414
0.714	0.00899693	1.63759	0.077031	0.944185
0.724	0.00910616	1.66049	0.0771509	0.949317
0.734	0.00896456	1.63766	0.077269	0.954836
0.744	0.00836929	1.53171	0.077384	0.960771

Um die Lizenz der TU zu nutzen muss als Lizenzserver *licserv-a.rhrk.uni-kl.de* angegeben werden

Um die firewall des Lizenzservers für eure IP frei zu schalten:



<https://licserv-a.rhrk.uni-kl.de/firewall>

Dort die firewall für Mathematica freischalten:

Startseite

Lizenzen

Firewall freischalten

Firewall prüfen

Aktuelle Auslastung

Statistik mit OpenLM

Firewall der Lizenzserver freischalten

Ihr RHRK-Account **koster** ist für die nachfolgend genannte Software freigeschaltet, deren Nutzung an die Ausgabe einer Lizenz durch die Lizenzserver der TU Kaiserslautern gebunden ist. Die Ausgabe von Lizenzen wird durch eine Firewall überwacht. Für die Nutzung von Lizenzen außerhalb des Campus-Netztes sowie im WLAN der TU Kaiserslautern ist eine Freischaltung erforderlich. Diese zeitlich befristete Freischaltung können Sie an dieser Stelle vornehmen.

Software	Firewall für IP-Adresse 131.246.81.231 freischalten...			
Abaqus	Sie sind nicht berechtigt, diese Software zu verwenden.			
Allinea DDT	1 Stunde	4 Stunden	8 Stunden	1 Tag
Altair Hyperworks	1 Stunde	4 Stunden	8 Stunden	1 Tag
ANSYS	Sie sind nicht berechtigt, diese Software zu verwenden.			
ANSYS (Lehre)	Eine Freischaltung ist nicht notwendig, da diese IP-Adresse immer freigeschaltet ist.			
ANSYS Apache	Sie sind nicht berechtigt, diese Software zu verwenden.			
ArcGIS	Sie sind nicht berechtigt, diese Software zu verwenden.			
Autodesk	Sie sind nicht berechtigt, diese Software zu verwenden.			
AVL Fire	Sie sind nicht berechtigt, diese Software zu verwenden.			
COMSOL	Sie sind nicht berechtigt, diese Software zu verwenden.			
CST	Sie sind nicht berechtigt, diese Software zu verwenden.			
Fluent / GAMBIT	Sie sind nicht berechtigt, diese Software zu verwenden.			
Gurobi	Eine Freischaltung ist nicht notwendig, da diese IP-Adresse immer freigeschaltet ist.			
Intel Compiler Suite	1 Stunde	4 Stunden	8 Stunden	1 Tag
Lumerical	Sie sind nicht berechtigt, diese Software zu verwenden.			
Mathematica	1 Stunde	4 Stunden	8 Stunden	1 Tag
Matlab	1 Stunde	4 Stunden	8 Stunden	1 Tag
Maple	1 Stunde	4 Stunden	8 Stunden	1 Tag
Maxwell Render	Eine Freischaltung ist nicht notwendig, da diese IP-Adresse immer freigeschaltet ist.			
Maxwell Render (Lehre)	Eine Freischaltung ist nicht notwendig, da diese IP-Adresse immer freigeschaltet ist.			
MSC Products	Sie sind nicht berechtigt, diese Software zu verwenden.			
OMD	Sie sind nicht berechtigt, diese Software zu verwenden.			
Origin	Sie sind nicht berechtigt, diese Software zu verwenden.			
PGI Compiler	1 Stunde	4 Stunden	8 Stunden	1 Tag
SPSS	1 Stunde	4 Stunden	8 Stunden	1 Tag

Jetzt könnt ihr Mathematica über den Lizenzserver der TU nutzen

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

Diesen Vortrag (und die anderen) könnt ihr finden unter:
<https://www.fs.physik.uni-kl.de/vortrag.html>