

Mathematica Einführung

Softwarekurse zum physikalischen Praktikum

Patrick Mischke

20.02.2019

1 Installation

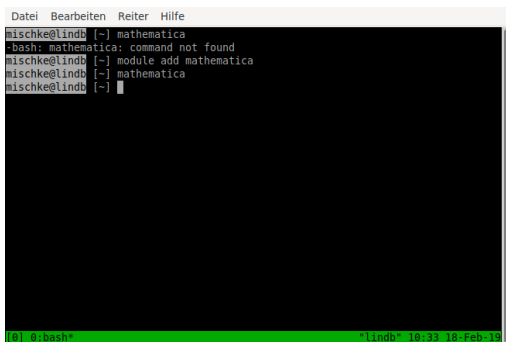
- Linda/Lindb
- Eigener Rechner

2 Verwendung

- Grundlagen
- Import von Daten
- Plotten
- Fitten

Mathematica auf Linda/Lindb

Auf den Servern Linda/Lindb ist Mathematica bereits installiert, muss aber aktiviert werden:

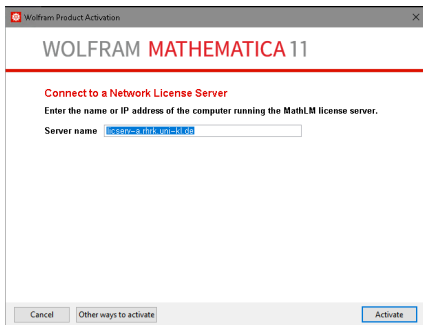


```

Datei Bearbeiten Reiter Hilfe
mischke@lindb [~] mathematica
-bash: mathematica: command not found
mischke@lindb [~] module add mathematica
mischke@lindb [~] mathematica
mischke@lindb [~] █

[0] 0:bash* *lindb* 10:33 18-Feb-19
```


Mathematica wird beim Starten nach einem Lizenzserver fragen.
Dieser ist `licserv-a.rhrk.uni-kl.de`.



Du musst die Lizenz freischalten unter
<https://licserv-a.rhrk.uni-kl.de/firewall.php>:

Firewall der Lizenzserver freischalten

Im RHRK-Account erteileMe 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 131.246.146.4 freischalten...					
Abacus	Sie sind nicht berechtigt, diese Software zu verwenden.					
Abacus (Lehr)	Sie sind nicht berechtigt, diese Software zu verwenden.					
Alinea OOT	1 Stunde	4 Stunden	8 Stunden	1 Tag	7 Tage	30 Tage
Alisa Hyperworks	1 Stunde	4 Stunden	8 Stunden	1 Tag	7 Tage	30 Tage
ANSYS	Sie sind nicht berechtigt, diese Software zu verwenden.					
ANSYS (Lehr)	Eine Freischaltung ist nicht notwendig, da diese IP-Adresse immer freigeschaltet ist.					
ANSYS Apache	Sie sind nicht berechtigt, diese Software zu verwenden.					
AutCAD	Sie sind nicht berechtigt, diese Software zu verwenden.					
Autodesk	Sie sind nicht berechtigt, diese Software zu verwenden.					
Aut. Fre	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.					
Excelp Products	Sie sind nicht berechtigt, diese Software zu verwenden.					
FlexPro	Sie sind nicht berechtigt, diese Software zu verwenden.					
Fluent / GAMBIT	Sie sind nicht berechtigt, diese Software zu verwenden.					
Gaussian	Eine Freischaltung ist nicht möglich, da derzeit ein anderer Lizenzserver verwendet wird.					
Guabi	Eine Freischaltung ist nicht notwendig, da diese IP-Adresse immer freigeschaltet ist.					
Intel Compiler Suite	1 Stunde	4 Stunden	8 Stunden	1 Tag	7 Tage	30 Tage
Lumental	Sie sind nicht berechtigt, diese Software zu verwenden.					
Mathematica	1 Stunde	4 Stunden	8 Stunden	1 Tag	7 Tage	30 Tage
Matlab	1 Stunde	4 Stunden	8 Stunden	1 Tag	7 Tage	30 Tage
Maple	1 Stunde	4 Stunden	8 Stunden	1 Tag	7 Tage	30 Tage
MAXEDA	Sie sind nicht berechtigt, diese Software zu verwenden.					

Grundrechenarten:

In[630]:= $3 * 7 / \text{Pi} + 2 * (3 - \text{I}) ^ 2$
 |Kreiszahl π |imaginär

Out[630]= $(16 - 12 i) + \frac{21}{\pi}$

Numerische Werte:

In[632]:= $\text{N}[\text{Sin}[2 + 3 i]]$
 |· |Sinus

Out[632]= $9.1545 - 4.16891 i$

Variablen und Funktionen definieren und aufrufen:

In[627]:= $\mathbf{a = 3;}$
 $\mathbf{F[x_, y_] := 3 x + y^2 - x * y;}$
 $\mathbf{F[3 a, 7 b + c]}$

Out[629]= $27 - 9 (7 b + c) + (7 b + c)^2$

Vektoren und Matrizen

Vektoren sind Listen, und Matrizen Listen aus Listen. Sie werden mit geschweiften Klammern $\{\}$ geschrieben.

```
In[638]:= mat = IdentityMatrix[3]
           |Einheitsmatrix
```

```
Out[638]= {{1, 0, 0}, {0, 1, 0}, {0, 0, 1}}
```

```
In[639]:= vec = {1, 5, 2, 30}
```

```
Out[639]= {1, 5, 2, 30}
```

Mit //Grid lassen sie sich hübsch anzeigen:

```
In[640]:= mat // Grid
           |Gitter
```

```
      1  0  0
Out[640]= 0  1  0
           0  0  1
```

Grundrechenarten werden elementenweise angewendet:

```
In[041]:= a = {1, 2, 3};  
         b = {1, 1, 2};  
         a * b
```

```
In[044]:= {1, 2, 6}
```

Mit `.` werden die üblichen Matrix/Vektormultiplikationen notiert:

```
In[055]:= a . b
```

```
Out[055]= 9
```

```
In[053]:= mat . a
```

```
Out[053]= {1, 2, 3}
```

```
In[054]:= mat . mat
```

```
Out[054]= {{1, 0, 0}, {0, 1, 0}, {0, 0, 1}}
```


Eine Funktion kann mit `/@` auf alle Einträge einer Liste angewendet werden:

```
In[866]:= F[x_] := x^2;  
F /@ fib
```

```
Out[867]:= {0, 1, 1, 4, 9, 25, 64, 169, 441, 1156, 3025}
```

Import von Daten

Der Import Befehl kann mit den richtigen Einstellungen nahezu alles Importieren.

```

In[10] := data = Import["C:\\Users\\Patrick\\Desktop\\MathematicaVortrag\\Beispiele_einfach.txt", "Table", NumberPoint -> ",", ""];
Out[10] :=
data // Grid
Table
t1      In      S      2.5*sin(0.1*t1-3,14)+2  10*sin(t1/2)*exp(-0,01*t1)  1,5*exp(-0,005*(t1-30,5)*(t1-30,5))  E=5*exp[-0,005*(t1-100,25)*(t1-100,25)]  10*exp(-0,05*t1)  5*exp(0,5*t1-10,5)
0      2.04577    0.032009    0.0526754    0.0526754    10.0099    0.00903962
1.      1.76803    4.7597      0.196836    0.196836    9.68273    0.0959423
1.5     1.79545    0.84374     0.100906    0.100906    9.47658    0.0955144
2      1.58048    8.37397     0.0944973    0.0944973    9.11035    0.0126018
2.75   1.48011    9.72886     0.0409349    0.0409349    8.75778    0.135164
3.4    1.30703    9.77753     0.056354    0.056354    8.6317    0.161713
4.05   1.14532    8.7483      0.112602    0.112602    8.29466    0.0198123
4.7    1.05765    6.96027     0.196193    0.196193    8.02185    0.155952
5.35   0.741212   4.42473     0.0746273    0.0746273    7.81237    0.194006
6.     0.675827   1.42166     0.176335    0.176335    7.51029    0.0259394
6.65   0.538194   -1.68035    0.127459    0.127459    7.3395    0.122392
7.3    0.363031   -4.3668     0.175842    0.175842    7.13956    0.126622
7.95   0.395637   -6.65633    0.309665    0.309665    6.82546    0.103372
8.6    0.200401   -8.27175    0.140379    0.140379    6.69563    0.154419
9.25   0.0633839  -9.03488    0.327185    0.327185    6.41344    0.0659386
9.9    -0.00827256 -8.73203    0.236743    0.236743    6.20698    0.190016
10.55  -0.101064  -7.471      0.222745    0.222745    5.97512    0.0706495
11.2   -0.0930985 -5.5906     0.311048    0.311048    5.0921    0.12069
11.85  -0.317107  -2.96403    0.309276    0.309276    5.60018    0.116295
11.9   0.549744   0.14106    0.474000    0.474000    6.44603    0.444006
  
```

Import von Daten

Excel Dateien werden als 3D-Array eingelesen:
Tabellenblatt, Zeile und Spalte

```
in[41]:= dataXLSX = Import["C:\\Users\\Patrick\\Desktop\\MathematicaVortrag\\einfacheBeispiele.xlsx"]
Importiere
```

```
in[42]:= dataXLSX[[1]] // Grid
Grid
```

t1	in s	2.5 sin(0.1 t1 - 3.14) + 2	10 sin(t1/2) exp(-0.01 t1)	1.5 exp(-0.005 (t1 - 30.5) + (t1 - 30.5))	E + 5 exp(-0.005 (t1 - 100.25) + (t1 - 100.25))	10 exp(-0.05 t1)	5 exp(0.5 t1 - 10.5)
0.	2.01296	0.105876	0.0057871	0.0057871	0.0057871	10.0094	0.0757197
1.	1.89615	4.92177	0.0976503	0.0976503	0.0976503	9.55778	0.198228
1.5	1.68246	6.82107	0.0257277	0.0257277	0.0257277	9.40299	0.171564
2.	1.53319	8.35188	0.0766646	0.0766646	0.0766646	9.00517	0.0122859
2.75	1.3225	9.58141	0.122935	0.122935	0.122935	8.82882	0.0716859
3.4	1.29578	9.77075	0.18835	0.18835	0.18835	8.52498	0.172038
4.05	1.14498	8.66384	0.23257	0.23257	0.23257	8.21448	0.0138991
4.7	0.932589	6.89445	0.245311	0.245311	0.245311	8.03539	0.105462
5.35	0.828965	4.45674	0.114068	0.114068	0.114068	7.82908	0.0650247
6.	0.748	1.37541	0.141697	0.141697	0.141697	7.56864	0.0956399
6.65	0.590002	-1.56147	0.0098005	0.0098005	0.0098005	7.23871	0.0681198
7.3	0.420992	-4.47196	0.111472	0.111472	0.111472	7.00462	0.0665024
7.95	0.251539	-6.69173	0.128623	0.128623	0.128623	6.76269	0.0930431
8.6	0.243978	-8.22794	0.162288	0.162288	0.162288	6.68643	0.0849771
9.25	0.156831	-9.02834	0.261502	0.261502	0.261502	6.45897	0.165187
9.9	-0.0158059	-8.62868	0.292453	0.292453	0.292453	6.12144	0.160716
10.55	-0.105745	-7.58255	0.315413	0.315413	0.315413	6.04557	0.174778
11.2	-0.159982	-5.59680	0.244849	0.244849	0.244849	5.73	0.0470341
11.85	-0.176102	-3.02231	0.272944	0.272944	0.272944	5.71882	0.193503
12.5	-0.200329	-0.147092	0.356658	0.356658	0.356658	5.40675	0.207946
13.15	-0.249972	2.56854	0.505441	0.505441	0.505441	5.29916	0.238527
13.8	-0.300344	6.17663	0.676017	0.676017	0.676017	6.07602	0.176001

Import von Daten

Unterstützte Dateiformate:

3DS, ACO, Affymetrix, AgilentMicroarray, AIFF, ApacheLog, ArcGRID, AU, AVI, Base64, BDF, Binary, Bit, BMP, Byte, BYU, BZIP2, CDED, CDF, Character16, Character8, CIF, Complex128, Complex256, Complex64, CSV, CUR, DBF, DICOM, DIF, DIMACS, Directory, DOT, DXF, EDF, EPS, ExpressionML, FASTA, FASTQ, FCS, FITS, FLAC, GenBank, GeoTIFF, GIF, GPX, Graph6, Graphlet, GraphML, GRIB, GTOPO30, GXL, GZIP, HarwellBoeing, HDF, HDF5, HIN, HTML, ICC, ICNS, ICO, ICS, Integer128, Integer16, Integer24, Integer32, Integer64, Integer8, JCAMP-DX, JPEG, JPEG2000, JSON, JVX, KML, LaTeX, LEDA, List, LWO, MAT, MathML, MBOX, MDB, MESH, MGF, MIDI, MMCIF, MOL, MOL2, MP3, MPS, MTP, MTX, MX, NASACDF, NB, NDK, NetCDF, NEXUS, NOFF, OBJ, ODS, OFF, OGG, OpenEXR, Package, Pajek, PBM, PCX, PDB, PDF, PGM, PLY, PNG, PNM, PPM, PXR, QuickTime, Raw, RawBitmap, RawJSON, Real128, Real32, Real64, RIB, RSS, RTF, SCT, SDF, SDTS, SDTSDem, SFF, SHP, SMILES, SND, SP3, Sparse6, STL, String, SurferGrid, SXC, Table, TAR, TerminatedString, Text, TGA, TGF, TIFF, TIGER, TLE, TSV, UnsignedInteger128, UnsignedInteger16, UnsignedInteger24, UnsignedInteger32, UnsignedInteger64, UnsignedInteger8, USGSDEM, UUE, VCF, VCS, VTK, WAV, Wave64, WDX, WebP, XBM, XHTML, XHTMLMathML, XLS, XLSX, XML, XPORT, XYZ, ZIP

Plotten von Funktionen

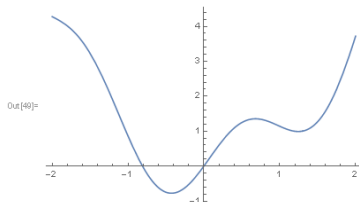
Der Plot Befehl plottet Funktionen:

```
In[40]:= F[x_] := Sin[3 x] + x^2;
```

Sinus

```
Plot[F[x], {x, -2, 2}]
```

stelle Funktion graphisch dar



Fancy Beispiel vom Plot Befehl:

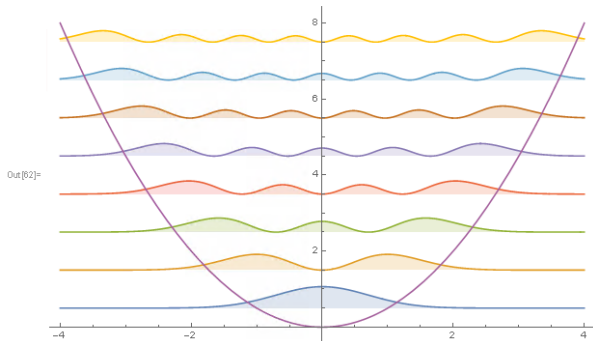
```
In[81]:= f[n_, x_] := Abs[(1/Pi)^(1/4) HermiteH[n, x] / (E^(x^2/2) Sqrt[2^n n!])]^2;
```

|Absolutw... |Kreiszahl π |hermitesches Polynom H |Exponentialk... |Quadratwurzel

```
Plot[Evaluate@ Append[Table[f[n, x] + n + 1/2, {n, 0, 7}], x^2/2], {x, -4, 4}, Filling -> Table[n -> n - 1/2, {n, 1, 8}]]
```

|stell... |werte aus |hänge ar |Tabelle

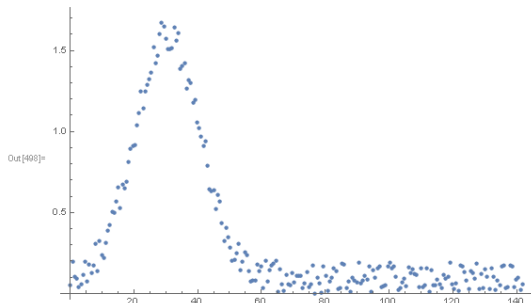
|Füllung |Tabelle



Plotten von Daten

Daten lassen sich mit dem ListPlot Befehl plotten:

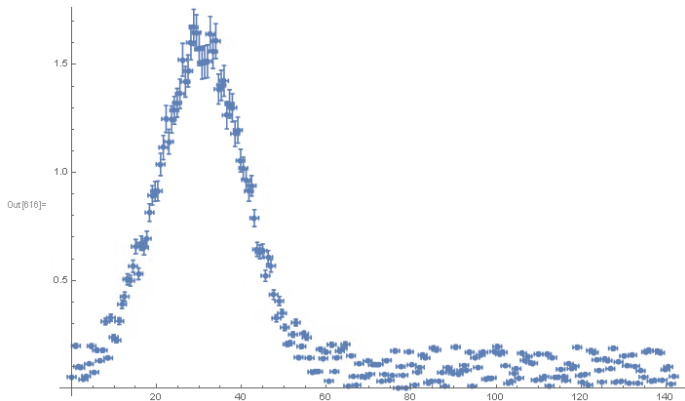
```
In[406]:= data = Import["C:\\Users\\Patrick\\Desktop\\MathematicaVortrag\\Beispiele_einfach.txt", "Table", NumberPoint -> ","];
           Importiere
           Dezimalpunkt
data = data[[2 ;;, {1, 4}]];
ListPlot[data]
           listenbezogene Graphik
```



Es lassen sich die Optionen vom Plot Befehl nutzen.

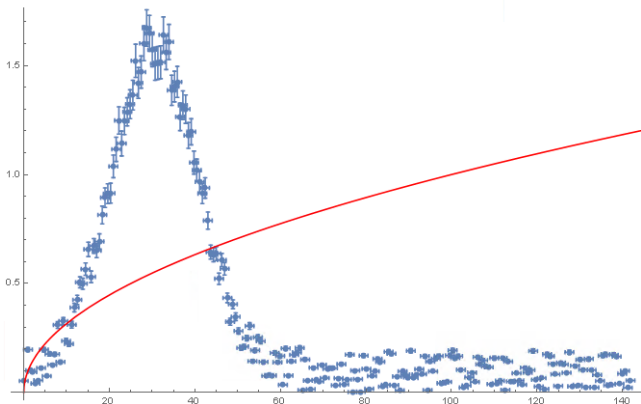
Fehlerbalken lassen sich mit `ErrorListPlot` plotten:

```
In[14]:= Needs["ErrorBarPlots`"];  
          benötigt  
dataMitFehler = ({#[[1]], #[[2]]}, ErrorBar[1, .05*#[[2]]] & /@ data;  
ErrorListPlot[dataMitFehler]
```



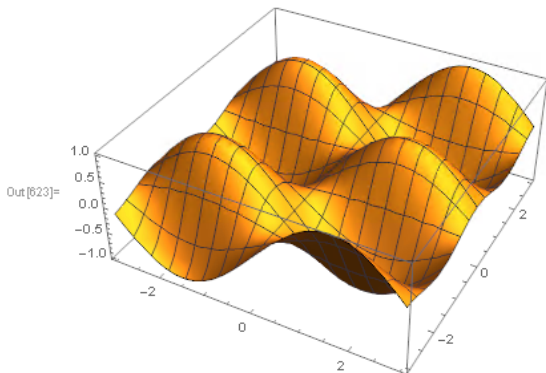
Mehrere Plots lassen sich mit Show kombinieren:

```
In[622]:= Show [ErrorListPlot[dataMitFehler], Plot[Sqrt[x] / 10, {x, 0, 150}, PlotStyle -> Red]]
```



Dreidimensionale Plots lassen sich mit Plot3D erstellen:

```
In[623]:= Plot3D[Sin[x] * Cos[2 y], {x, -3, 3}, {y, -3, 3}]  
|stelle ... |Sinus |Kosinus
```



Zum Speichern von Plots (und allem anderen) lässt sich der Export Befehl verwenden:

```
plot = Plot3D[Sin[x] * Cos[2 y], {x, -3, 3}, {y, -3, 3}];
```

stelle ... Sinus Kosinus

```
Export["Plot.pdf", plot]
```

exportiere

Out [825]= Plot.pdf

Assuming a file name | Use as a *generic text string* instead

open

import

open file directory

full path filename

more...



Fitten von Daten

Die Methode `NonlinearModelFit` erstellt Fits:

```
In[479]:= data = Import["C:\\Users\\Patrick\\Desktop\\MathematicaVortrag\\Beispiele_einfach.txt", "Table", NumberPoint -> ","];
           |importiere                                     |Dezimalpunkt
data = data[[2 ;;, {1, 3}]];
FitFunktion[x_] := a * Sin[b * x + c] * Exp[-d * x];
           |Sinus                                         |Exponentialfunktion
fit = NonlinearModelFit[data, FitFunktion[x], {a, b, c, d}, x];
           |nichtlineare Modellanpassung
```

Der Fit lässt sich mit `Normal` auslesen:

```
Normal[fit]
|normal
Out[488]= -10.012 e-0.0100214 x Sin[0.00201499 - 0.500022 x]
```

Für die Parameter lassen sich einige statistische Daten auslesen

```
In[491]:= fit["ParameterTable"]
```

	Estimate	Standard Error	t-Statistic	P-Value
a	10.012	0.0305932	327.263	4.74704×10^{-292}
b	0.500022	0.000057057	8763.55	$6.251798274051461 \times 10^{-599}$
c	-0.00201499	0.0030376	-0.663348	0.507818
d	0.0100214	0.0000569241	176.049	2.18693×10^{-234}

Geplottet sieht das so aus:

```
In[492]:= Show[ListPlot[data], Plot[fit[x], {x, 0, 200}]]
```

zeige · listenbezogene Gra · stelle Funktion graphisch dar

