Karlsruher Institut für Technologie

Erfassung, Auswertung und Präsentation von Messergebnissen

Moritz Klammler 16. Februar 2012

Fachschaft f. Chemie u. Biowissenschaften · Institut f. Physikalische Chemie

Hinweis

Typographische Änderungen im Vergleich zur im Seminar gezeigten Präsentation und Errata sind in der schönen Farbe KIT-Violett markiert.

Kolophon

Diese Präsentation wurde mit LaTeX und der *beamer* Klasse von Till Tantau, Joseph Wright und Vedran Miletić erstellt. Das verwendete Theme ist $k_{it} \rightarrow$ von Moritz Klammler. Grafiken wurden mit *tikz* von Till Tantau erstellt. Das Titelbild ist ein Ausschnitt aus dem Gemälde Waldinneres mit kleiner Kuh- und Schafherde von Cornelis J. de Vogel (1824–1879), Öl auf Holz, 70 cm × 55 cm. Die Grundschrift ist *Linux Biolinum* mit *Latin Modern Typewriter Proportional* als dicktengleiche Schrift.

Copyright 2012 Moritz Klammler

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
`http://www.gnu.org/licenses/fdl.txt`

Der LaTeX Quellcode ist auf Anfrage (`moritz.klammler@gmail.com`) vom Autor erhältlich.

Organisatorisches

9^{45} –11^{15}	Praktikumsvorbesprechung	HS II
11^{30}–13^{00}	Theoretischer Teil	HS II
Mittagspause		
14^{00}–16^{00}	Übung	Pool G + H (SCC)

Übersicht

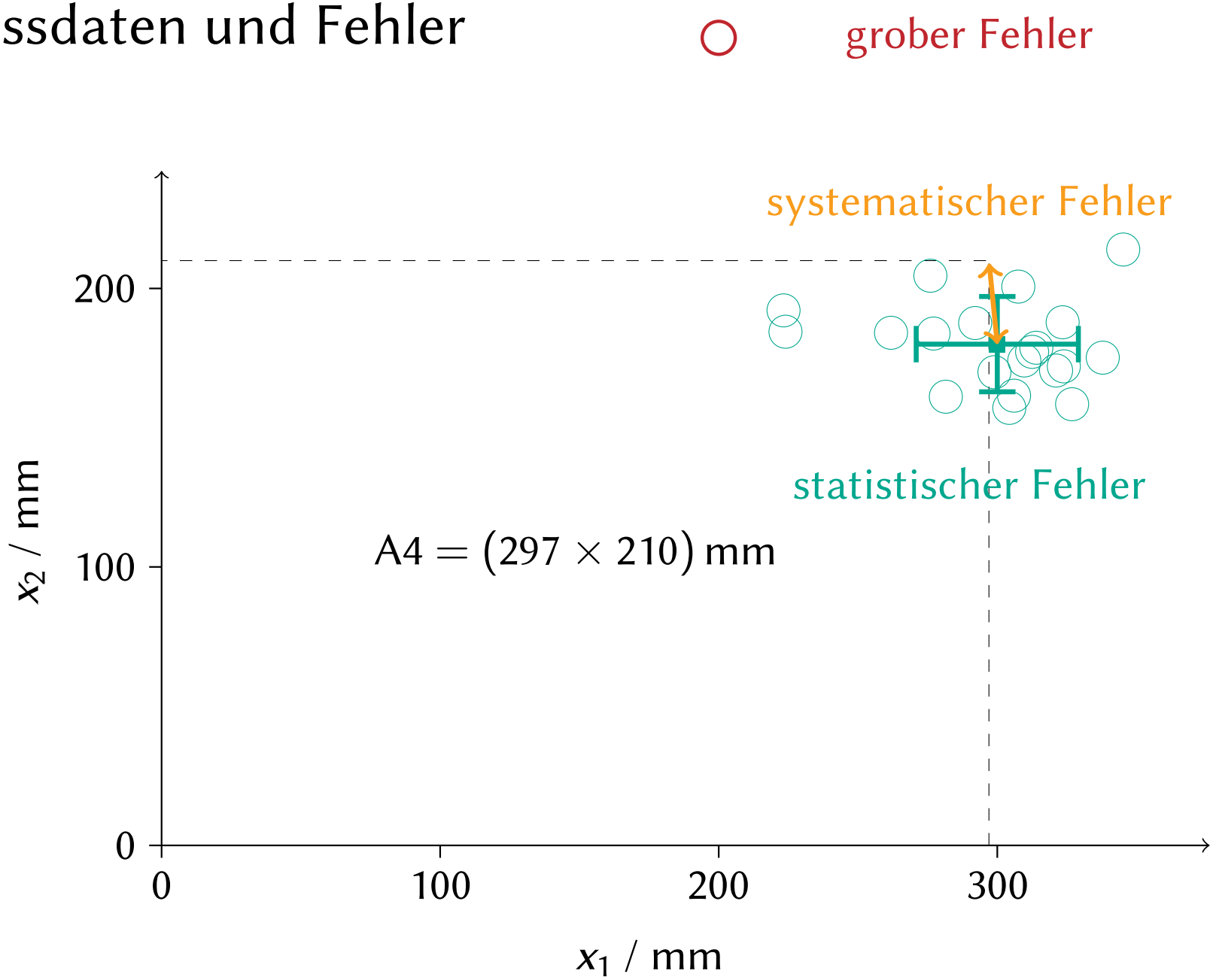
1 **Organisatorisches**

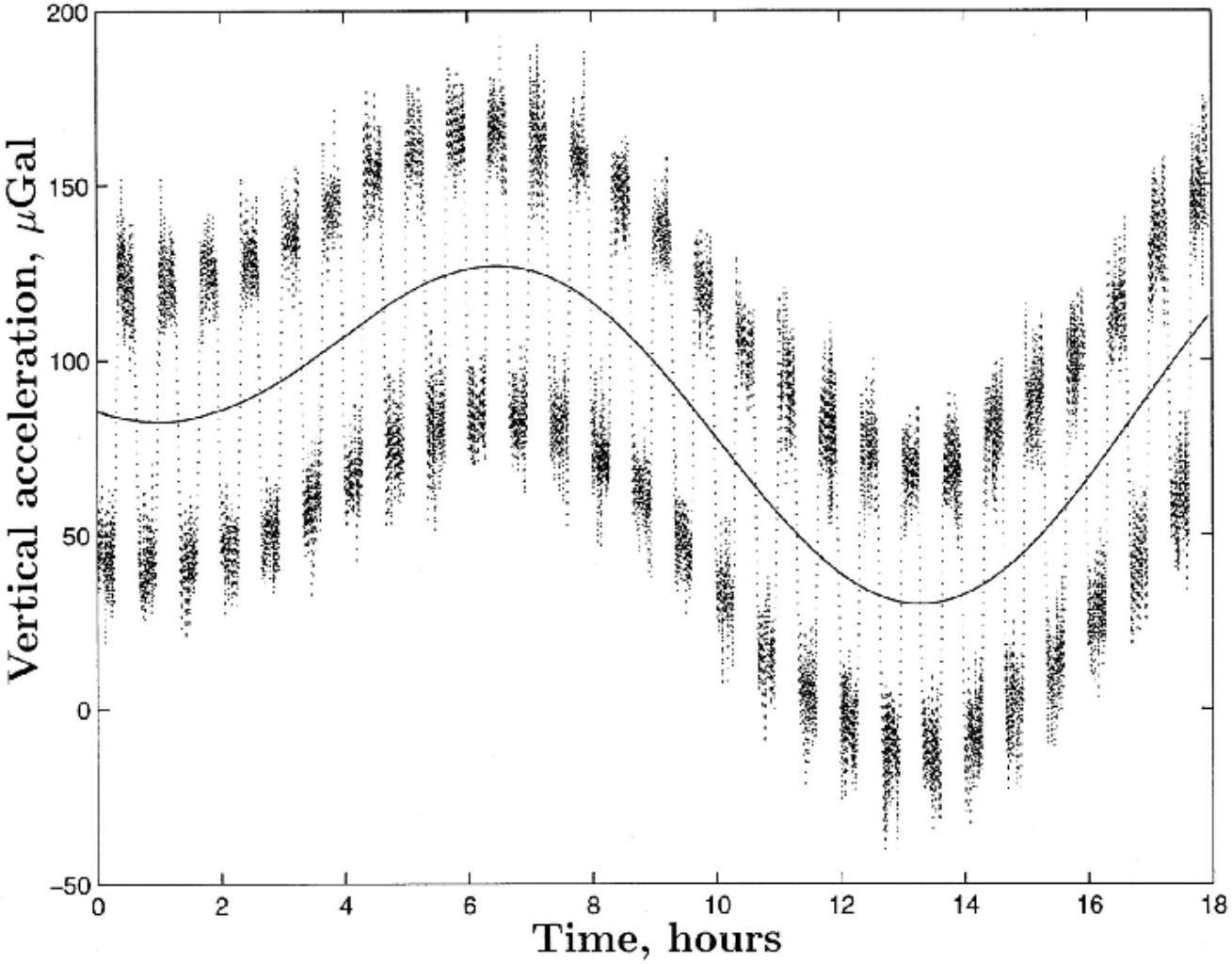
2 **Messdaten, Fehler und Statistik**
- Einleitung
- Mittelwert und Standardabweichung
- Konfidenzniveau und Konfidenzintervall
- Ausreißertests
- Fehlerfortpflanzung

3 **Regression**

4 **Infinitesimalrechnung mit empirischen Daten**
- Integration
- Differentiation

Messdaten und Fehler





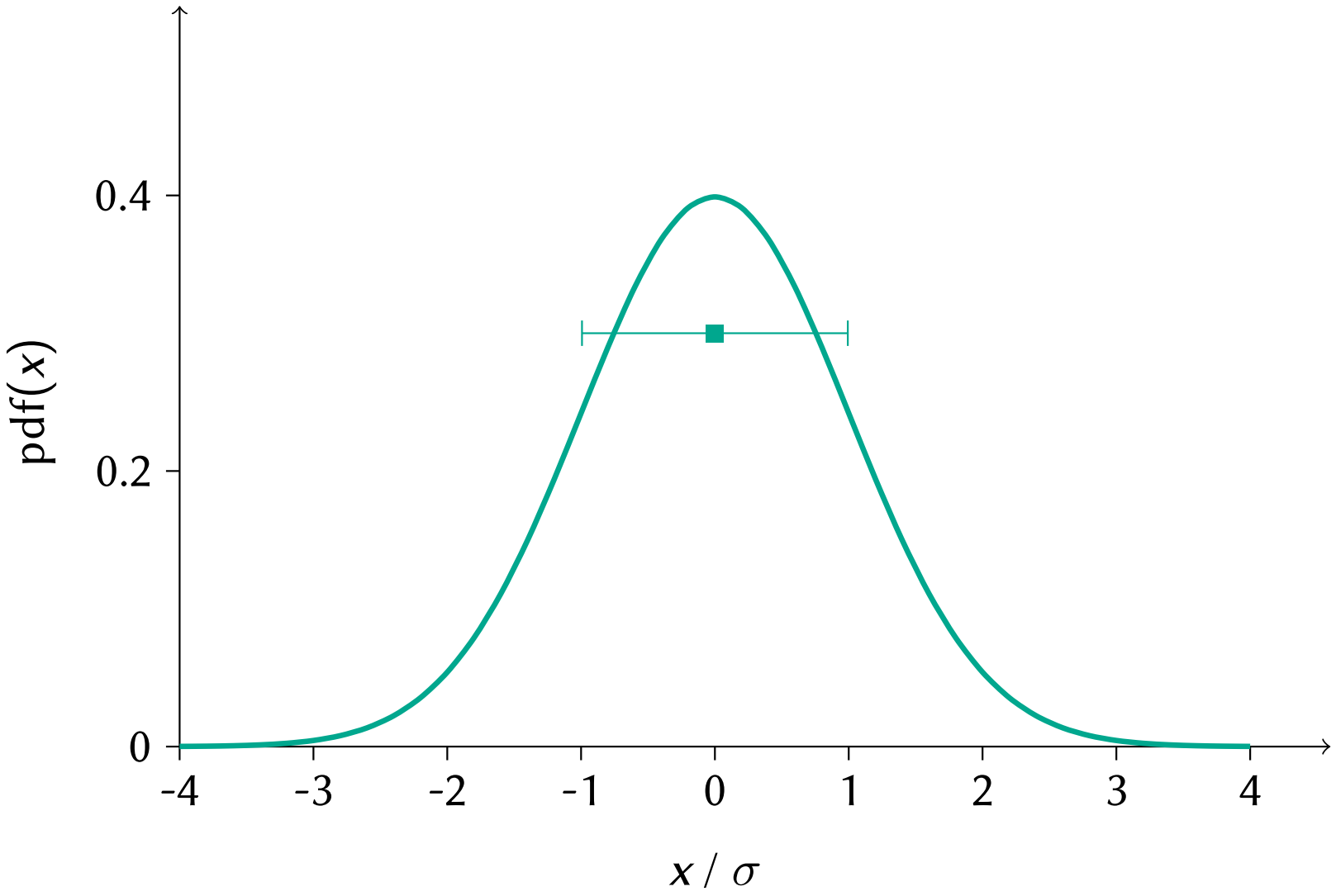
J. P. Schwarz, D. S. Robertson, T. M. Niebauer, und J. E. Faller, *Science*, **1998**, *282*(5397), 2230–2234.

Mittelwert und Standardabweichung

empirischer Mittelwert	**empirische Varianz**
M_X bzw. `MEAN`	V_X bzw. `VAR`
$M_X = \sum_{i=1}^{N} p_i x_i$	$V_X = \frac{N}{N-1} \sum_{i=1}^{N} p_i \left(x_i - M_X\right)^2$
$M_X = \frac{1}{N} \sum_{i=1}^{N} x_i$	$V_X = \frac{1}{N-1} \sum_{i=1}^{N} \left(x_i - M_X\right)^2$
„Schwerpunkt“	„Trägheitsmoment“

empirische Standardabweichung: S_X (bzw. `STDEV`) $= \sqrt{V_X}$

Normalverteilung



Konfidenzniveau und Konfidenzintervall

$$U = (M_U \pm n\, S_U)\,\mathrm{u}$$

Konfidenzintervall $= 2\, n\, S_U$
(In der Physik) üblich: $n = 1$

$n = 1$	$\Rightarrow$	Konfidenzniveau	$\gtrsim 68\,\%$
$n = 2$	$\Rightarrow$		$\gtrsim 95\,\%$
$n = 3$	$\Rightarrow$		$\gtrsim 99\,\%$

Konfidenzniveau und Konfidenzintervall

In der Physik üblich: Standardabweichung auf 2 signifikante Stellen.

$$\left.\begin{array}{rcl} M_U & = & 2.71828183\ldots\,\mathrm{u} \\ S_U & = & 0.00314159\ldots\,\mathrm{u} \end{array}\right\} \Rightarrow \quad U = (2.7183 \pm 0.0031)\,\mathrm{u}$$

Alternative Schreibweise („condensed form“): $U = 2.7183(31)\,\mathrm{u}$
(Wird häufig missverstanden!)

Ausreißertest nach Grubbs

$$G_i = \frac{|x_i - M_X|}{S_X}$$

- $G_i > G_{\text{crit},\,\alpha} \quad \Rightarrow \quad x_i$ ist Ausreißer
- Werte für $G_{\text{crit},\,\alpha}$ siehe Tabelle.

Fehlerfortpflanzung I

Physikalische Größe U bestimmen über

$$U = f(\boldsymbol{x})$$

mit Messgrößen

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \qquad \boldsymbol{x_0} \equiv \begin{pmatrix} M_{x_1} \\ M_{x_2} \\ \vdots \\ M_{x_n} \end{pmatrix} \qquad \boldsymbol{\Delta x} \equiv \begin{pmatrix} S_{x_1} \\ S_{x_2} \\ \vdots \\ S_{x_n} \end{pmatrix}$$

Fehlerfortpflanzung II

$$U \approx f(\boldsymbol{x_0})$$

$$|\Delta U| \approx \sqrt{\sum_{i=1}^{n} (\Delta x_i)^2 \left(\frac{\partial f}{\partial x_i}(\boldsymbol{x_0}) \right)^2}$$

$$|\Delta U| \lesssim \sum_{i=1}^{n} \left| \Delta x_i \frac{\partial f}{\partial x_i}(\boldsymbol{x_0}) \right| \qquad \text{(Minkowski)}$$

Fehler des Mittelwerts bei wiederholter ($\Delta X_i = \Delta X_{\mathrm{sgl}} = \mathrm{const.}$) Messung

$$U = M_X = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \frac{\partial}{\partial X_i} U = \frac{1}{n}$$

$$\begin{aligned}
\Delta U &= \sqrt{\sum_{i=1}^{n}\left(\Delta X_i \frac{\partial U}{\partial X_i}\right)^2} = \sqrt{\sum_{i=1}^{n}\left(\frac{\Delta X_{\mathrm{sgl}}}{n}\right)^2} \\
&= \Delta X_{\mathrm{sgl}}\sqrt{\sum_{i=1}^{n}\frac{1}{n^2}} = \frac{\Delta X_{\mathrm{sgl}}}{\sqrt{n}}
\end{aligned}$$

Mathematica in einer Nussschale

- Vordefinierte Namen immer in CamelCase (z.B. `ListPlot`)
- Funktionsargumente in eckigen Klammern (z.B. `Cos[x]`)
- Listen in geschwungenen Klammern (z.B. `l = {a, b, c}`)
- Gruppierung mit runden Klammern (z.B. `a / (b + c)`)
- Optionale Argumente mit Pfeil (->): z.B.
  `Plot[Sin[x], {x, 0, 2π}, PlotStyle → Thick]`
- Sonderzeichen: $\boxed{\text{Esc}}\langle \textit{Name} \rangle\boxed{\text{Esc}}$
- Definieren von Funktionen: `f[x_, y_] := x^2 + y^2`
- Eingabe auswerten mit $\boxed{\text{Shift}} + \boxed{\text{Enter}}$
- Ausgabe unterdrücken mit schließendem Semikolon
  (z.B. `c = 299792458;`)
- Komentare: `(* ... *)`

Operation	**konventionelle Notation**	*Mathematica* **Input**
Grundrechnungsarten	$+,\ -,\ \cdot,\ /$	`+, -, *, /`
Funktionen	$\sqrt{x},\ \cos(\phi)$	`Sqrt[x], Cos[`ϕ`]`
Definition von Listen	$\vec{x} = (x_1, x_2, \ldots)$	`x = {x1, x2, ...}`
	$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$	`A = {{a11, a12}, {a21, a22}}`
Zugriff auf Listen	x_i	`x[[i]]`
	A_{ij}	`A[[i]][[j]]`
Vektorrechnen	$\lambda\vec{r},\ \vec{a}\cdot\vec{b}$	`λ r, a . b`

Mathematica: Statistik und I/O

```
In[1]:=   dat = {24.56, 23.89, 23.12};
In[2]:=   mean = Mean[dat]
Out[2]:=  23.8567
In[3]:=   stdev = StandardDeviation[dat]
Out[3]:=  0.720578
```

Einlesen aus Datei:

```
In[1]:=   SetDirectory[NotebookDirectory[]]
Out[1]:=  /home/tux/Documents/university/...
In[2]:=   dat = Flatten[Import["⟨datei⟩", "Data"]]
Out[2]:=  {24.56, 23.89, 23.12}
```

Mathematica: Fehlerforpflanzung

```
In[1]:=   area[x1_, x2_] := x1 x2;
          x1 = 297.8;
          Δx1 = 0.5;
          x2 = 210.3;
          Δx2 = 0.5;
In[2]:=   area[x1, x2]
Out[2]:=  62627.3
In[3]:=   ΔA = Sqrt[
           (Δx1 D[area[ξ1,ξ2],ξ1] /. {ξ1→x1,ξ2→x2})^2 +
           (Δx2 D[area[ξ1,ξ2],ξ2] /. {ξ1→x1,ξ2→x2})^2]
Out[3]:=  182.285
```

Regression („fitting“)

Experiment $(x_i, U_i, \Delta U_i)$ Modell $U = f_{\boldsymbol{\alpha}}(x)$

$i = 1 \ldots n$

Stützstellen $\boldsymbol{x} \equiv \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$

Fit-Parameter $\boldsymbol{\alpha} \equiv \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{pmatrix}$

$N < n$

Stützvektor $\boldsymbol{U} \equiv \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_n \end{pmatrix}$

Residuen-vektor $\boldsymbol{\varepsilon} \equiv \begin{pmatrix} (f_{\boldsymbol{\alpha}}(x_1) - U_1)/\Delta U_1 \\ (f_{\boldsymbol{\alpha}}(x_2) - U_2)/\Delta U_2 \\ \vdots \\ (f_{\boldsymbol{\alpha}}(x_3) - U_3)/\Delta U_3 \end{pmatrix}$

Aufgabe: Finde $\boldsymbol{\alpha}$ sodass $\|\boldsymbol{\varepsilon}\|$ minimal wird.

Lineare Modelle:

- $f_{\boldsymbol{\alpha}}(x) \equiv \sum_{i=1}^{N} \phi_i(x)$

Nicht-lineare Modelle:

- f hängt in nicht linearer Weise von $\boldsymbol{\alpha}$ ab
- i.d.R. nur numerisch lösbar
- Konvergenz nicht garantiert
- vernünftige Anfangswerte setzen!

Qualitätsbewertung

The first thing to do with any data set is to look at it.

- $\chi^2 \equiv \frac{1}{N} \|\varepsilon\|^2$
- Erwartungwert für Normalverteilte Daten: $\chi^2 = 1$
- $\chi^2 \gg 1 \quad \Rightarrow \quad$ Modell beschreibt das Experiment schlecht
- $\chi^2 \ll 1 \quad \Rightarrow \quad$ Experiment war unseriös!

N.H. Bingham und J.M. Fry, *Regression.* Springer, 2010.

Mathematica: LinerModelFit

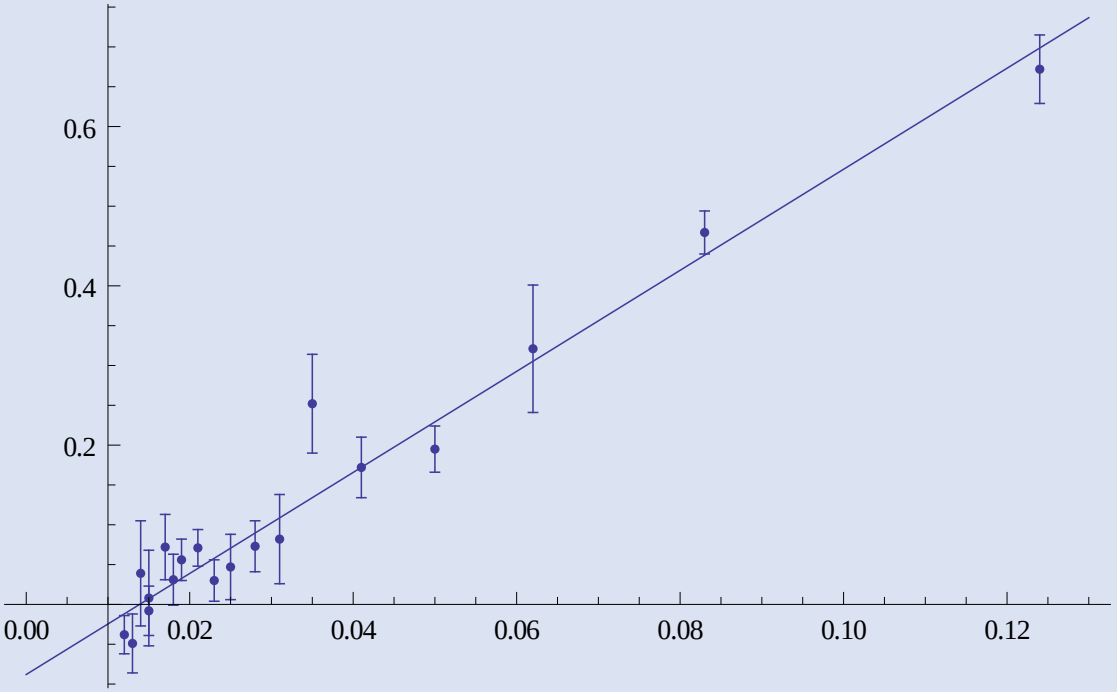
```
In[1]:=   SetDirectory[NotebookDirectory[]];
In[2]:=   data = Import["⟨datei⟩","Data"];
In[3]:=   data = Delete[data,1]; (* 1. Zeile löschen. *)
In[4]:=   xydata = Transpose[Delete[Transpose[data], 3]];
In[5]:=   yerrdata = Transpose[data][[3]];
In[6]:=   fit = LinearModelFit[xydata,x,x,
            Weights→1/yerrdata]
Out[6]:=  FittedModel[⟨funktion⟩]
In[7]:=   fit["ParameterConfidenceIntervals"]
Out[7]:=  {{⟨min1⟩,⟨max1⟩}, ...}
In[8]:=   Needs["ErrorBarPlots`"] (* Gravis nicht vergessen! *)
In[9]:=   p1 = ErrorListPlot[data];
In[10]:=  p2 = Plot[fit[x],{x,⟨min⟩,⟨max⟩}];
```

Mathematica: LinerModelFit

In[11]:=

```
Show[{p1, p2}]
```

Out[11]:=



Mathematica: NonlinearModelFit

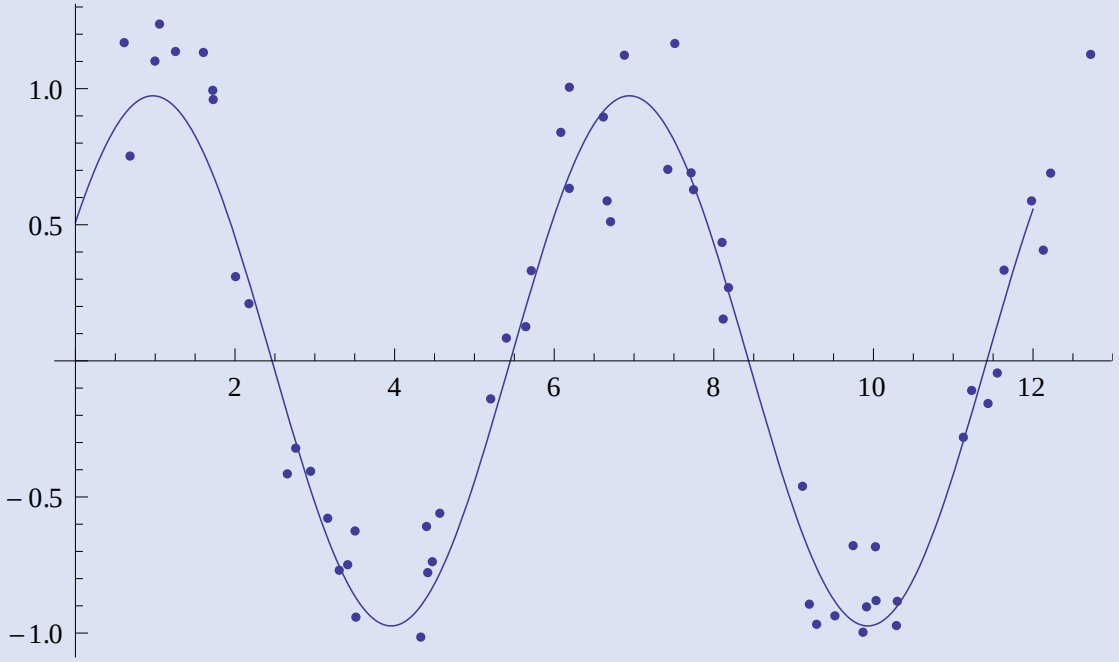
Sei data eine $2 \times n$ Liste von Messdaten, an die eine periodische Funktion zu fitten ist.

```
In[1]:=   fit = NonlinearModelFit[data,
            A Sin[ω (t+ϕ)],{A,ω,ϕ},t]
In[2]:=   p1 = ListPlot[data];
          p2 = Plot[fit[t], {t,⟨min⟩,⟨max⟩}];
```

Mathematica: NonlinerModelFit

In[3]:= `Show[{p1, p2}]`

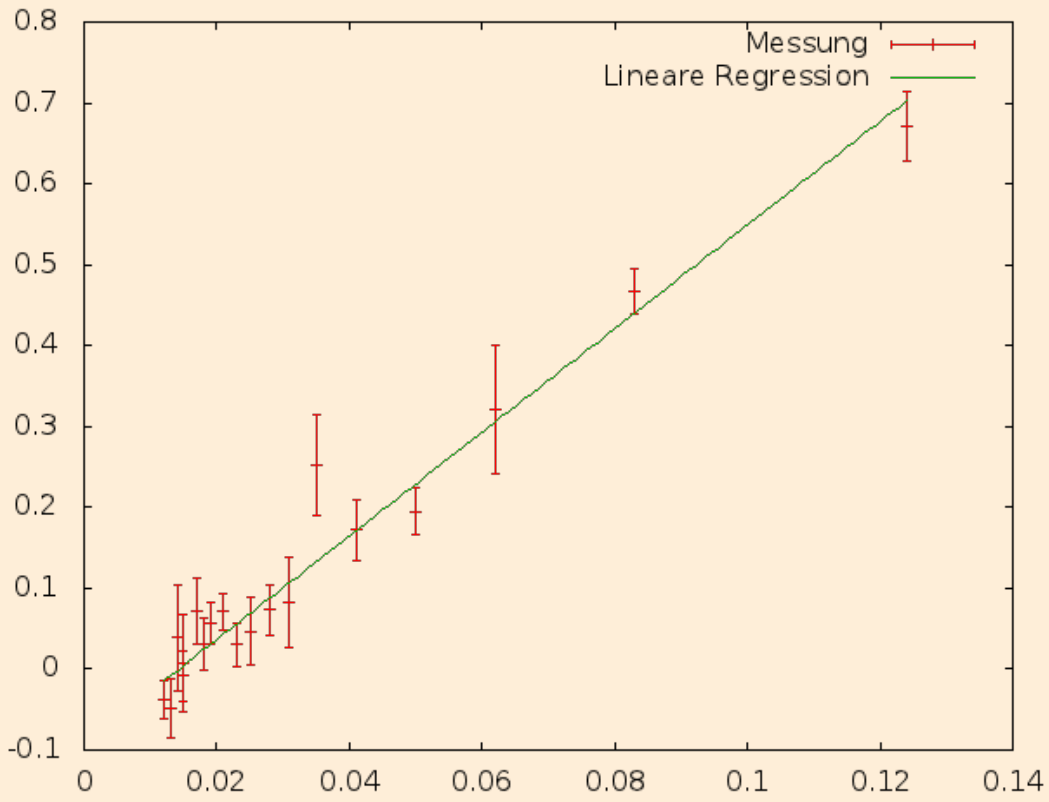
Out[3]:=



gnuplot: fit (Gerade)

```
f(x) = A + B * x
A = 1.0
B = 1.0
fit f(x) "⟨datei⟩" using 1:2:3 via A, B
plot "⟨datei⟩" with yerrorbars title "Messung", ...
             ... f(x) with lines title "Lineare Regression"
```

gnuplot: fit (Gerade)



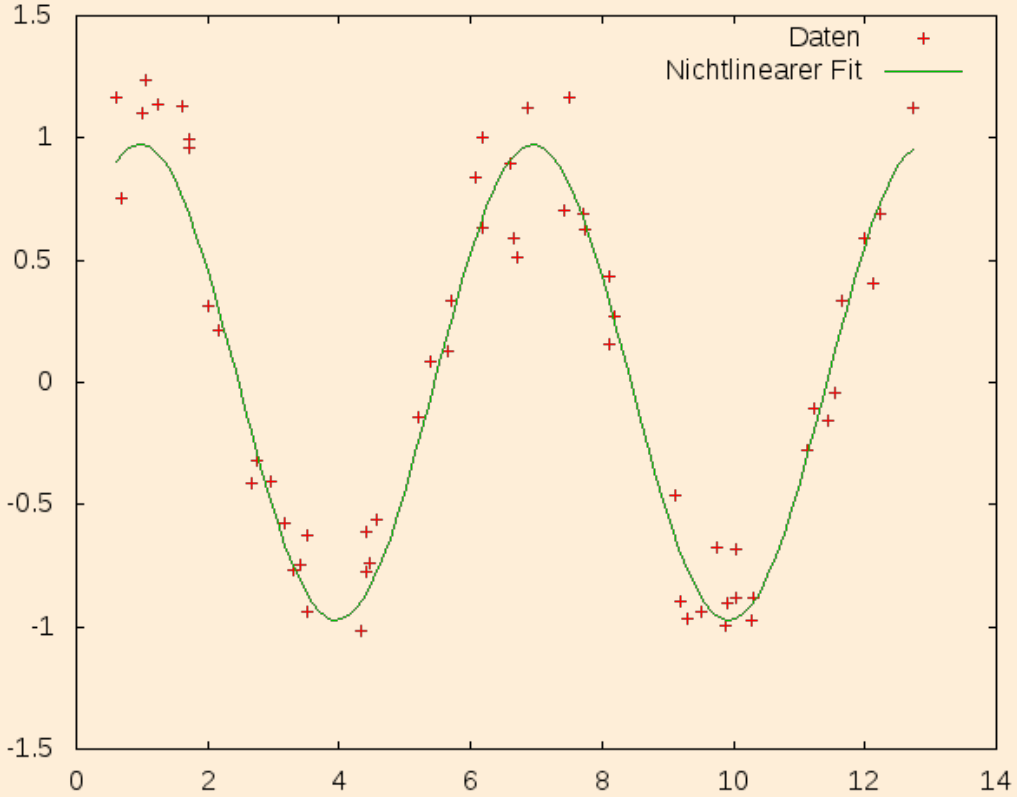
Ausführlicher Bericht auf `stdout` und in `fit.log`

gnuplot: fit (beliebig, nicht-linear)

Beliebig komplexe Funktionen lassen sich mit *gnuplot* genauso einfach fitten wie Geraden!

```
f(x) = A * sin(B * x + C)
A = 1.0
B = 1.0
C = 0.5
fit f(x) "⟨datei⟩" via A, B, C
plot "⟨datei⟩" title "Messdaten", ...
                    ... plot f(x) title "Nichtlinearer Fit"
```

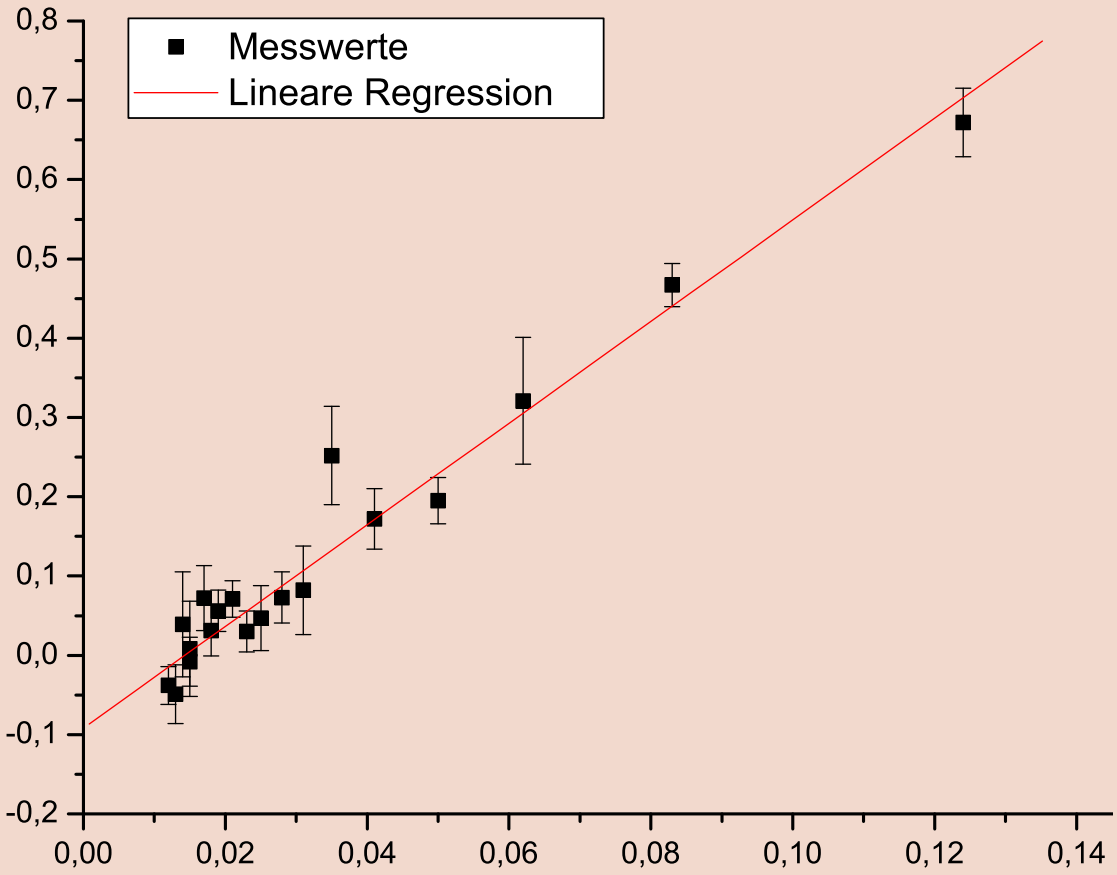
gnuplot: fit (beliebig, nicht-linear)



Origin: Linearer Fit

- Menü: *Datei* → *Import* → *ASCII Import…*
- Ggf. Dezimaltrennzeichen richtig einstellen.
- Spaltentypen auf `X`, `Y` und `yEr`± stellen.
- Daten markieren und im Menü: *Analyse* → *Anpassen* → *Fit Linear*
- In *Origin 7* steht das Ergebnis rechts unten, *Origin 8* fügt eine Tabelle ein.
- Anpassen der Datenreihen- und Achsenbeschriftung durch Doppelklick.

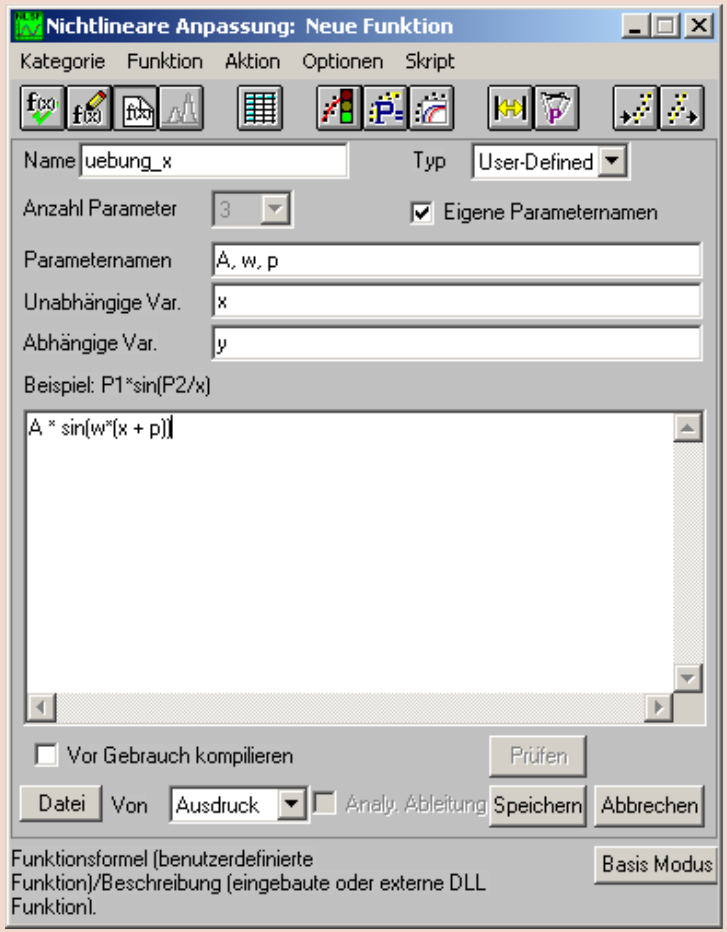
Origin: Linearer Fit

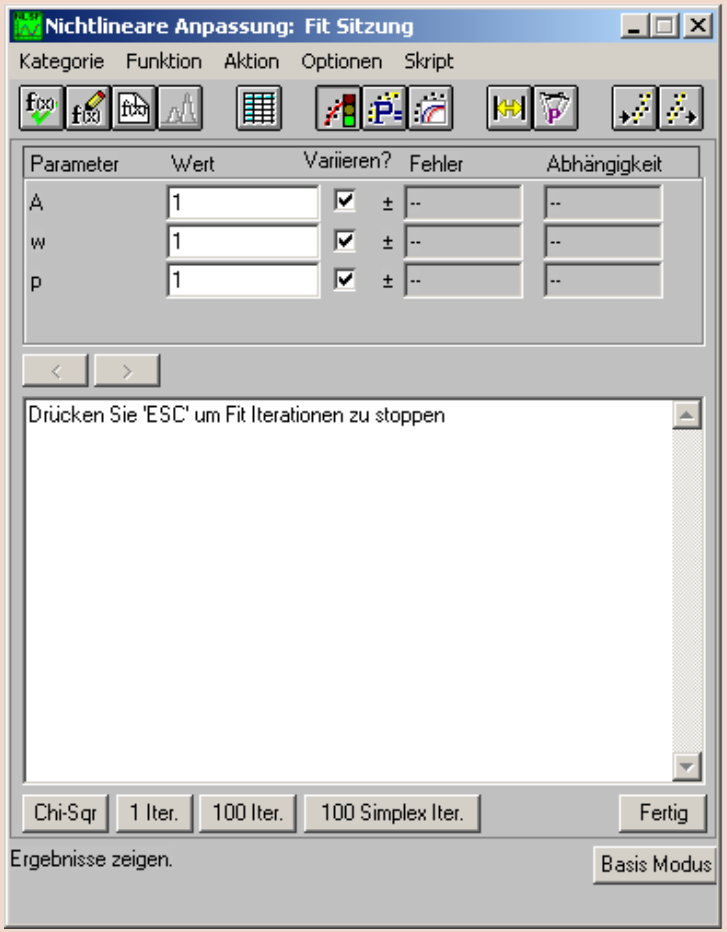


Origin: Nichtlinearer Fit

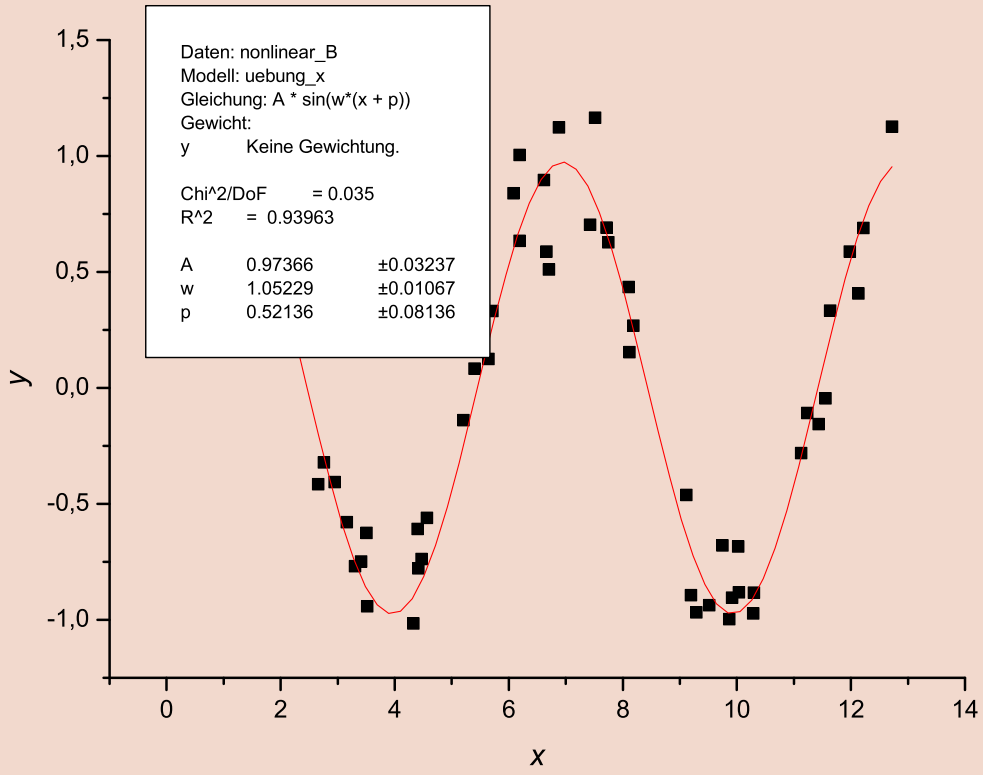
- Importieren wie zuvor.
- *Nichtlinearen Fit Assistenten* mit Ctrl + Y starten und abarbeiten. (Funktioniert häufig nicht wie gewollt.)
- Startwerte für Parameter setzen!

Origin: Nichtlinearer Fit (Origin 7)





Origin: Nichtlinearer Fit



Infinitesimalrechnung mit empirischen Daten

- Infinitesimalrechnungen sind auf diskreten Domänen nicht definiert
- falls möglich: Gefittete Funktion verwenden
- notfalls Polynome niedrigen Grades verwenden

Mathematica: NIntegrate

Sei `data` eine $2 \times n$ Liste von Messdaten, deren Fläche auf die x-Achse zu bestimmen ist. Ein funktionaler Zusammenhang sei nicht bekannt.

In[1]:= `f[x_] := Interpolation[data][x];`

In[2]:= `NIntegrate[pdf[x], {x,`$\langle min \rangle$`,`$\langle max \rangle$`}]`

Out[2]:= 42.

Origin

- Importieren wie zuvor.
- Im Menü: *Analysis* $\rightarrow$ *Integrieren*
- In *Origin 7* werden rechts unten die Fläche und einige weitere Informationen angezeigt.

Mathematica: D

Sei `data` eine $2 \times n$ Liste von Messdaten ohne bekannten funktionalen Zusammenhang. Gesucht ist die Steigung der Kurve.

Hahnebüchen: Interpolationspolynom differenzieren

```
In[1]:=   ip = InterpolatingPolynomial[data, x];
In[2]:=   ipf[ξ_] := ip /. x→ξ
In[3]:=   dip = D[ip, x];
In[4]:=   dipf[ξ_] := dip /. x→ξ
```

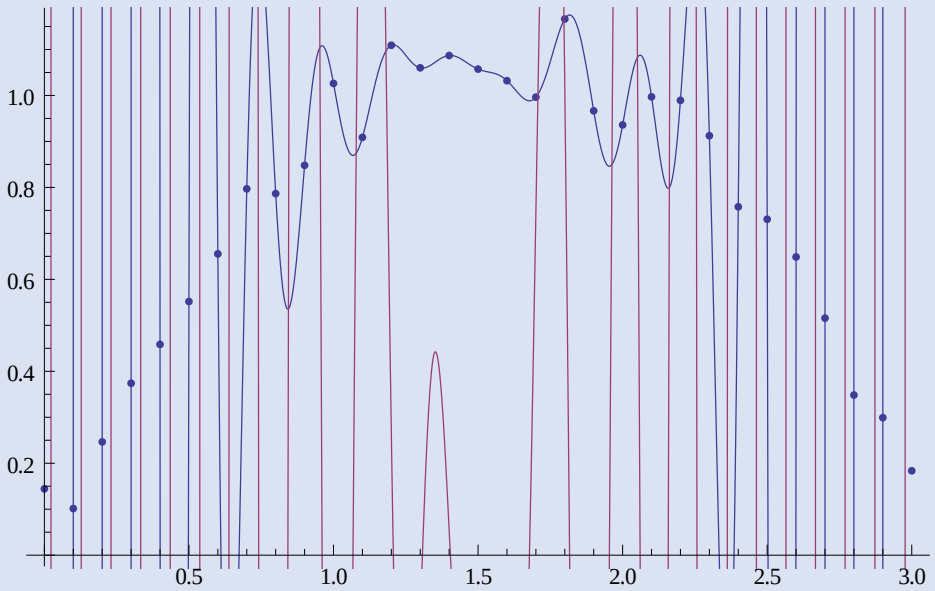
Mathematica: D

In[5]:=
```
p1 = ListPlot[data];
```

In[6]:=
```
p2 = Plot[{ipf[x],dipf[x]}, {x,0,3}];
```

In[7]:=
```
Show[{p1,p2}]
```

Out[7]:=



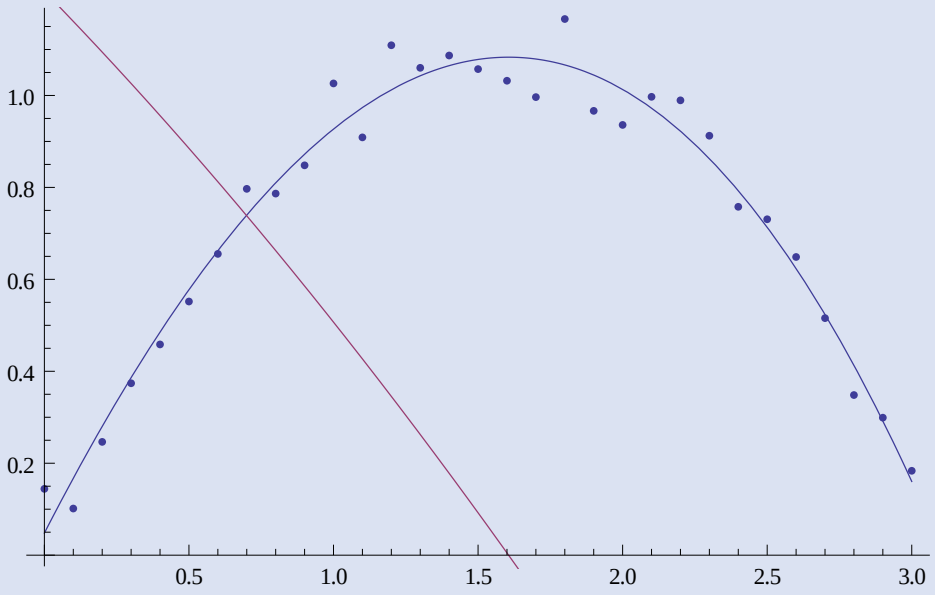
Mathematica: D

Vernünftiger: Polynom dritten Grades fitten und differenzieren

```
In[1]:=   ff = LinearModelFit[data,{x,x^2,x^3},x];
In[2]:=   df = D[ff[ξ], ξ];
In[3]:=   dff[x_] := df /. ξ→x
```

Mathematica: D

```
In[4]:=   p1 = ListPlot[data];
In[5]:=   p2 = Plot[{ff[x],dff[x]}, {x,0,3}];
In[6]:=   [{p1,p2}]
Out[6]:=
```



Origin

- Importieren wie zuvor.
- Im Menü: *Analysis* $\rightarrow$ *Differenzieren*

Software

	Mathemathica	*Origin*	*gnuplot*	*LaTeX*
Freie Software	nein	nein	ja	ja
Preis	*a)*	*a)*	kostenlos	kostenlos
Bezugsquelle	*a)*	*a)*	*b)*	*c)*
MS Windows	ja	ja	ja	ja
Mac OS	ja	nein	ja	ja
GNU/Linux	ja	nein	ja	ja

a) Für KIT-Studenten kostenlos im Softwareshop des Campus Süd erhältlich: `https://rzunika.asknet.de/`

b) `http://www.gnuplot.info/`

c) `http://www.latex-project.org/`

Unterlagen, Literatur und weitere Hilfe

- P.K. Janert, *Gnuplot in Action: Understanding Data with Graphs.* Manning, 2009.
- J. Schneider, *Kurzeinfuhrung in gnuplot.* 2004, `http://www.staff.uni-mainz.de/schneidj/kurse/ss06/gnuplot.pdf`.
- Präsentation und Kursunterlagen + Musterlösungen online: `http://klammler.eu/teaching.html`

Danksagungen

- Der Praktikumsleitung für ihr Entgegenkommen und die gute Kooperation.
- Den Kollegen aus der Fachschaft für ihren Rat und ihre Hilfe.
- Euch für die Aufmerksamkeit.