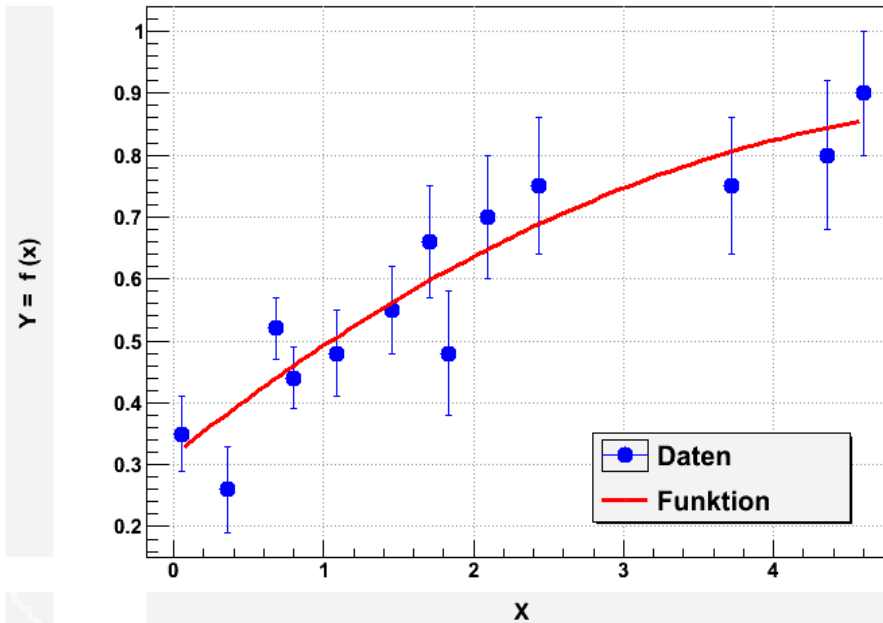


Anpassung von *Modellen* = *parameterabhängige Funktionen*
an *statistische Daten* = *Messwerte*



Messdaten (x_i, y_i) mit
angepasster Funktion $f(x; \mathbf{p})$
 $\mathbf{p}$ ist der Vektor der Parameter der Funktion

Benötigen ein „Abstandsmass“ für Übereinstimmung von (x_i, y_i) und $f(x_i)$

Mathematisch: y_i und $f_i = f(x_i)$ sind Elemente eines Vektorraums: Vektoren $\mathbf{y}$ und $\mathbf{f}$

Abstand = Betrag des Differenzvektors, $d^2 = (\mathbf{y} - \mathbf{f}) \cdot (\mathbf{y} - \mathbf{f})$ (Skalarprodukt)

also: **Finde Parameterwerte, die d^2 (für geeignet definiertes Skalarprodukt) minimieren**

Parameterschätzung

- χ^2 und Likelihood
- Parameterschätzung
(= Funktionsanpassung) mit
der Methode der kleinsten
Fehlerquadrate (χ^2 -Methode)
- Maximum-Likelihood Methode
- Anpassungen mit ROOT
- Fertige Programmpakete

Script: Anpassen von Funktionen an Messdaten

26. September 2010

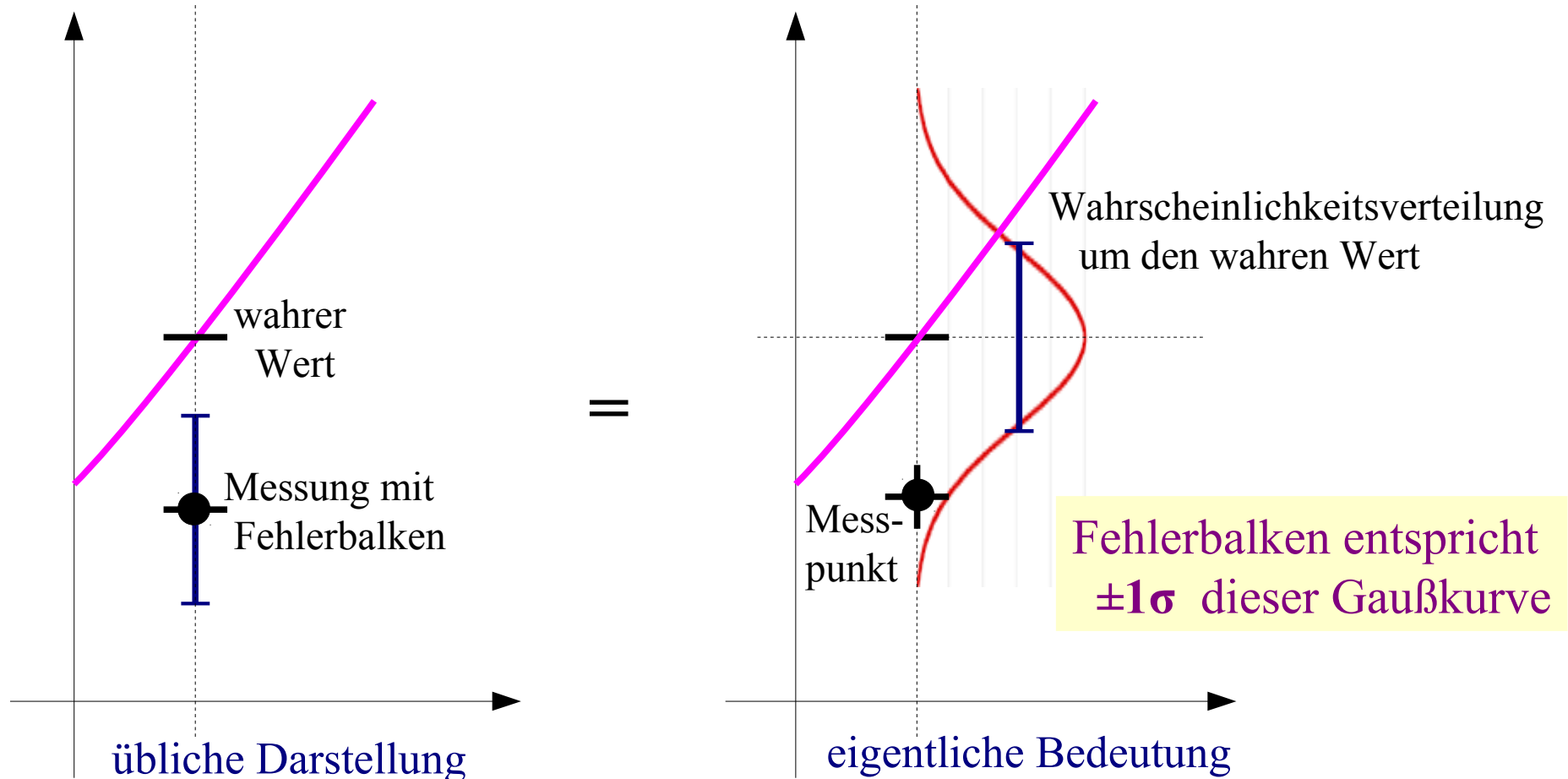
Funktionsanpassung mit der χ^2 -Methode

Zusammenfassung

Der Vergleich von Modellen mit Messungen gehört zu den Standardaufgaben in der Experimentalphysik. Im einfachsten Fall stellt ein Modell eine Funktion dar, die Vorhersagen für Messdaten liefert und die üblicherweise von Modellparametern abhängt. Neben der Überprüfung, ob das Modell die Daten beschreibt, gehört die Bestimmung der Modellparameter zu den typischen Aufgaben. Zur Funktionsanpassung wird häufig die Methode der kleinsten Quadrate verwendet, mit der sich sehr elegant auch korrelierte Unsicherheiten oder Fehler in Abszissenrichtung zusätzlich zu Fehlern in Ordinatenrichtung behandeln lassen. Dieses Script gibt einen kurzen Abriss der Methode und enthält praktische Hinweise zur Funktionsanpassung mit Hilfe numerischer Methoden auf dem Computer.

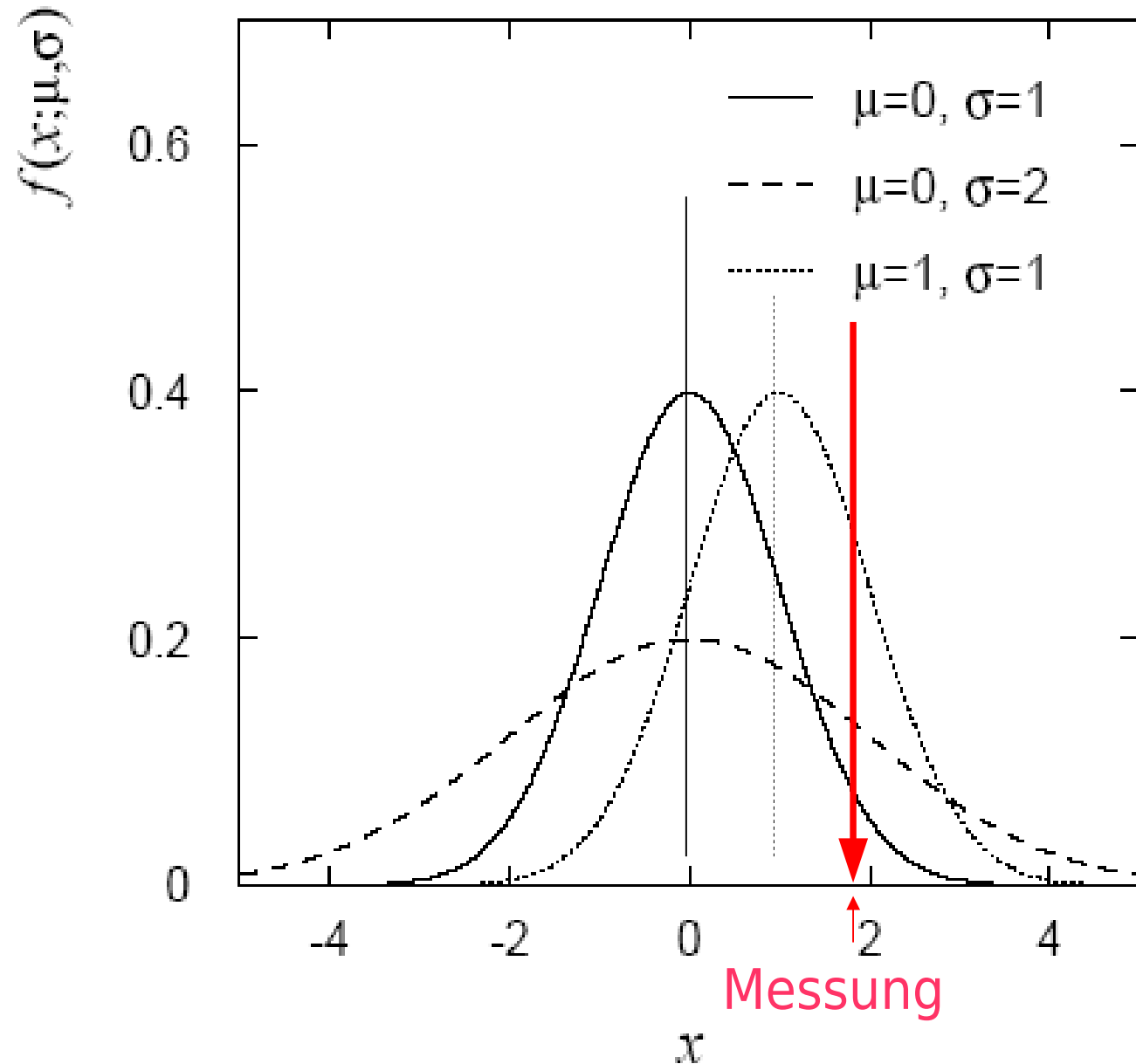
Parameterschätzung

Kleinste Quadrate (χ^2) und Maximum Likelihood

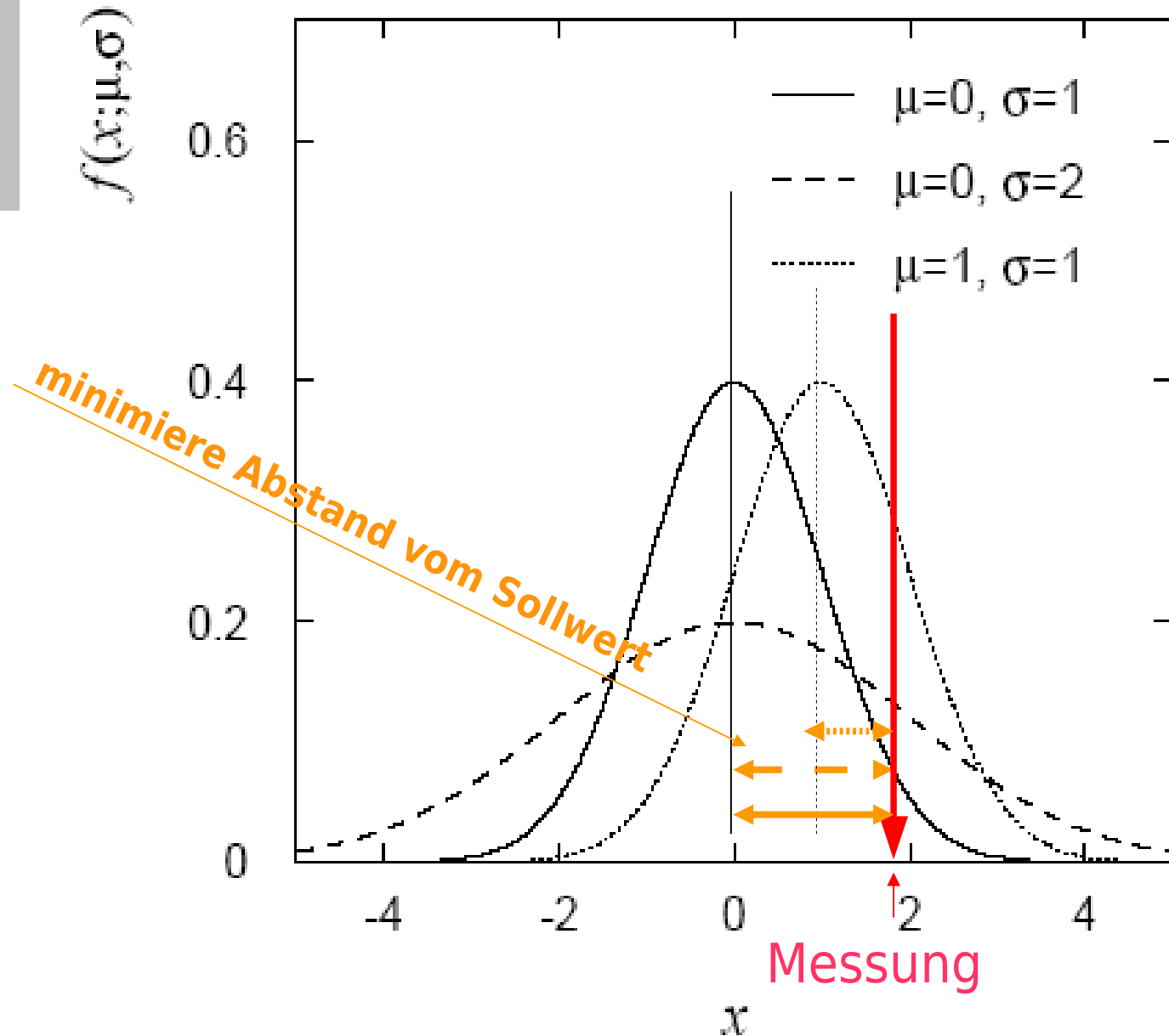


Messung mit Fehlerbalken bedeutet: Beobachtung eines Messergebnisses, das einer Zufallszahl aus einer Verteilungsdichte um den wahren Wert entspricht.

Welche Verteilung passt am besten zu Messung ?

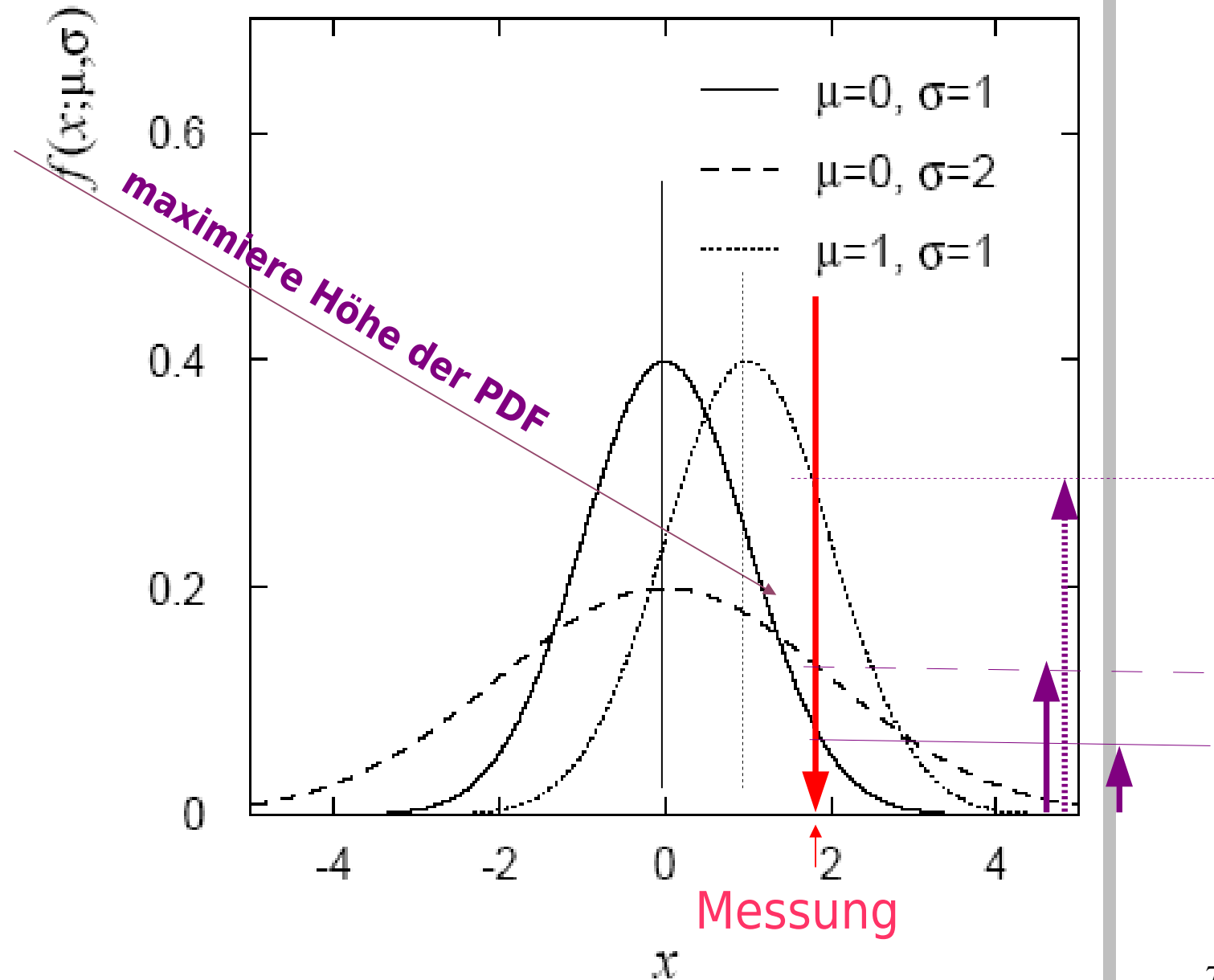


kleinster Abstand
Messung $\leftrightarrow$
Erwartungswert



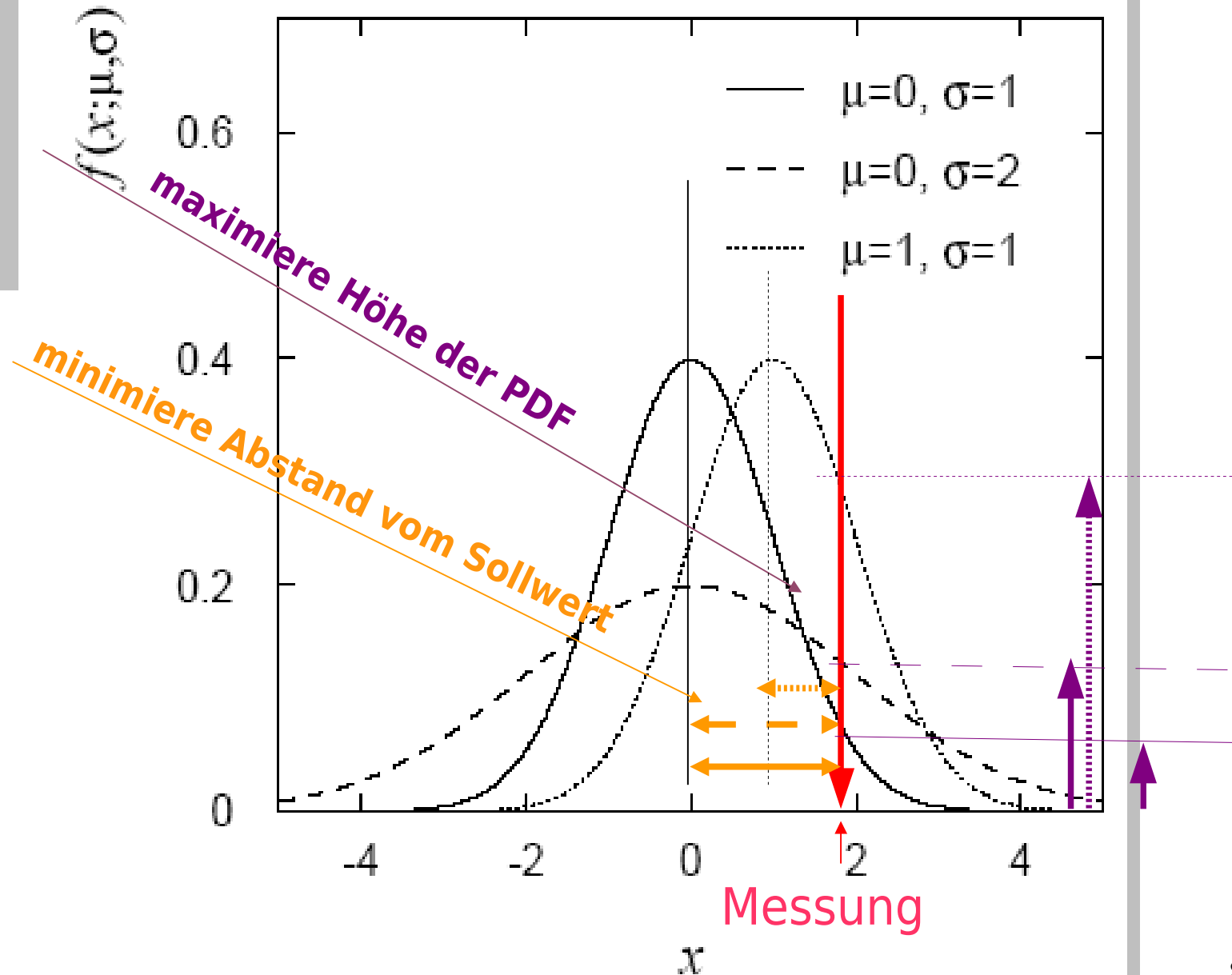
Methode: Maximum Likelihood

Maximale
Wahrscheinlichkeit



2 Methoden: Maximum Likelihood & Kleinste Quadrate

kleinster Abstand
Messung $\leftrightarrow$
Erwartungswert
oder
Maximale
Wahrscheinlichkeit



2 Methoden: Maximum Likelihood & Kleinste Quadrate

kleinster Abstand
Messung $\leftrightarrow$
Erwartungswert
oder
Maximale
Wahrscheinlichkeit

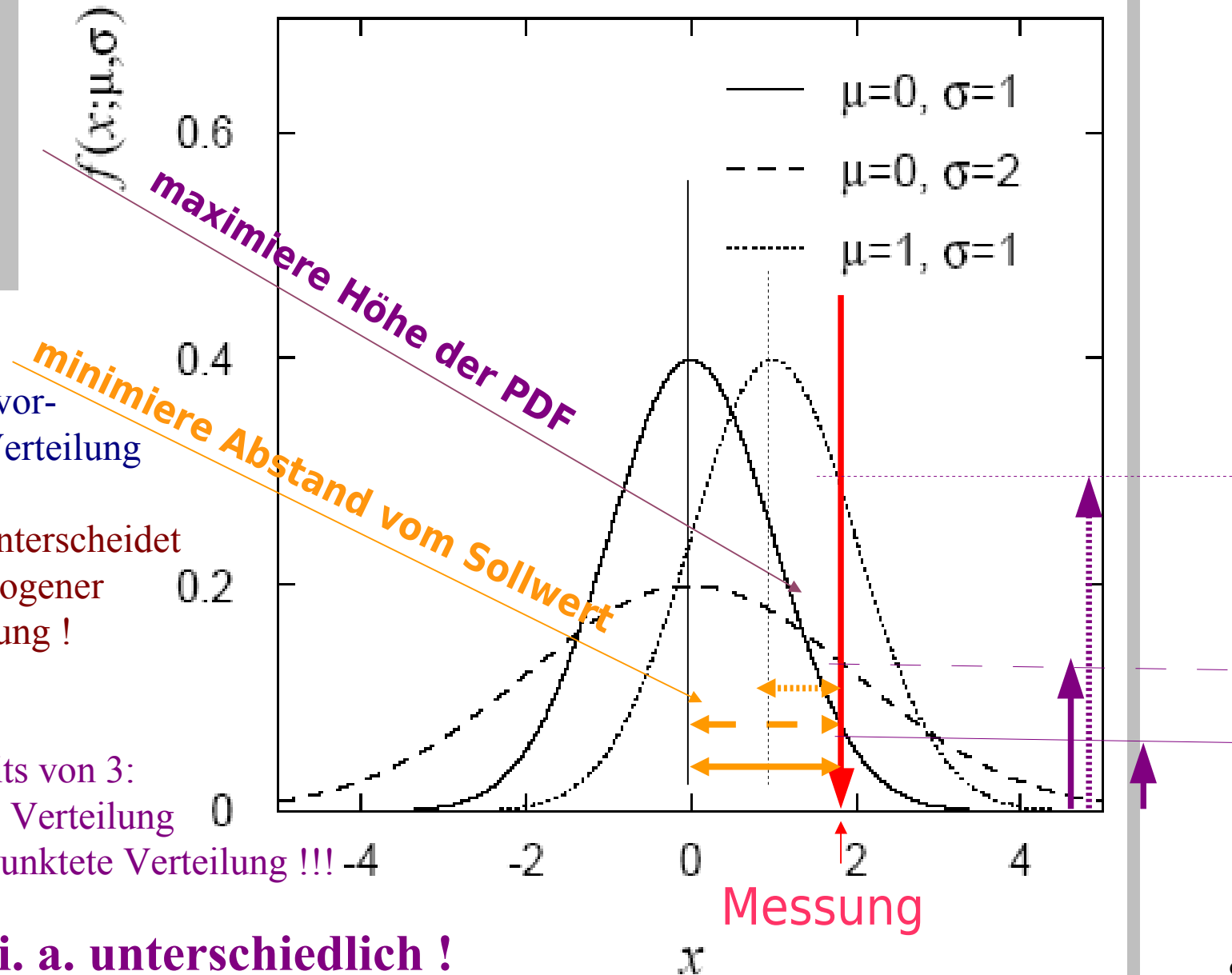
Hier: beide Methoden bevorzugen gepunktete Verteilung

aber: kleinste Quadrate unterscheidet nicht zwischen durchgezogener und gestrichelter Verteilung !

Läge der Messwert jenseits von 3:

likelihood: gestrichelte Verteilung

kleinste Quadrate: gepunktete Verteilung !!!



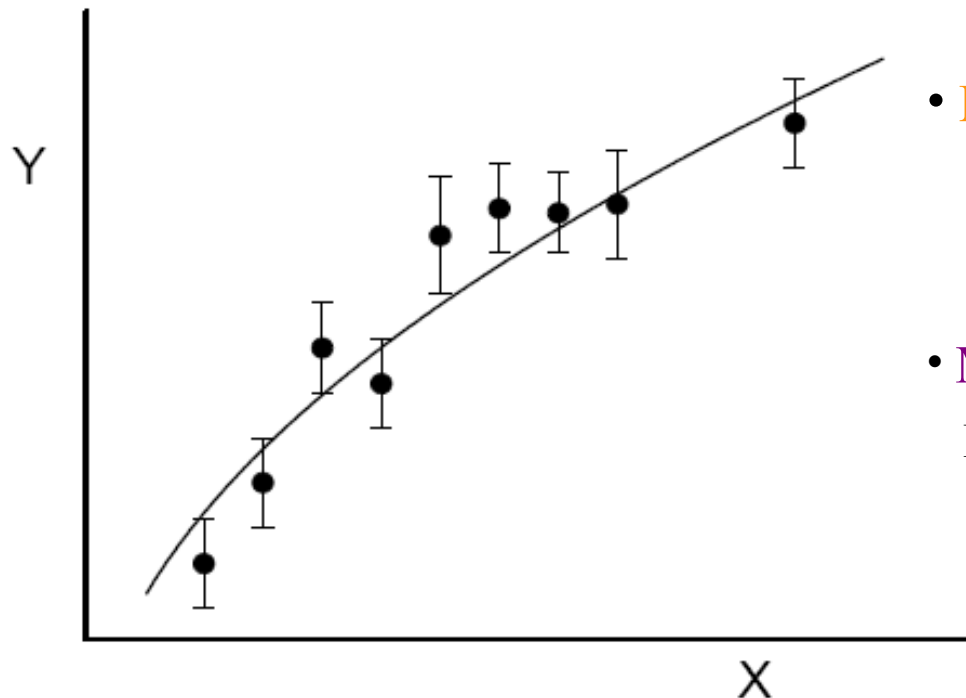
Methoden sind i. a. unterschiedlich !

Methode der kleinsten Quadrate (χ^2 - Methode)

Parameterschätzung

- im allgemeinen:**
- Viele Paare von Messwerten (x_i, y_i) , und Modell $y_i = f(x_i, \{p\})$ mit einer Menge von Parametern $\{p\}$
 - Messwerte sind fehlerbehaftet, d.h. um wahre Werte y_i verteilt, z.B. Gauß-förmig mit „Messfehler“ σ_i :

$$\mathcal{N}(y_i; f(x_i, \{p\}), \sigma_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp \frac{-(y_i - f(x_i, \{p\}))^2}{2\sigma_i^2}$$



- **Methode der kleinste Quadrate:**
Abweichungen der Messwerte y_i vom Erwartungswert $f(x_i; \{p\})$ auf Messfehler normieren, und deren quadratische Summe minimieren
- **Maximum-Likelihood:**
Produkt der Wahrscheinlichkeiten $\prod P_i(y_i | f(x_i, \{p\}))$, maximieren

(bei Gauß-förmigen Fehlern sind die beiden Verfahren äquivalent, s. später)

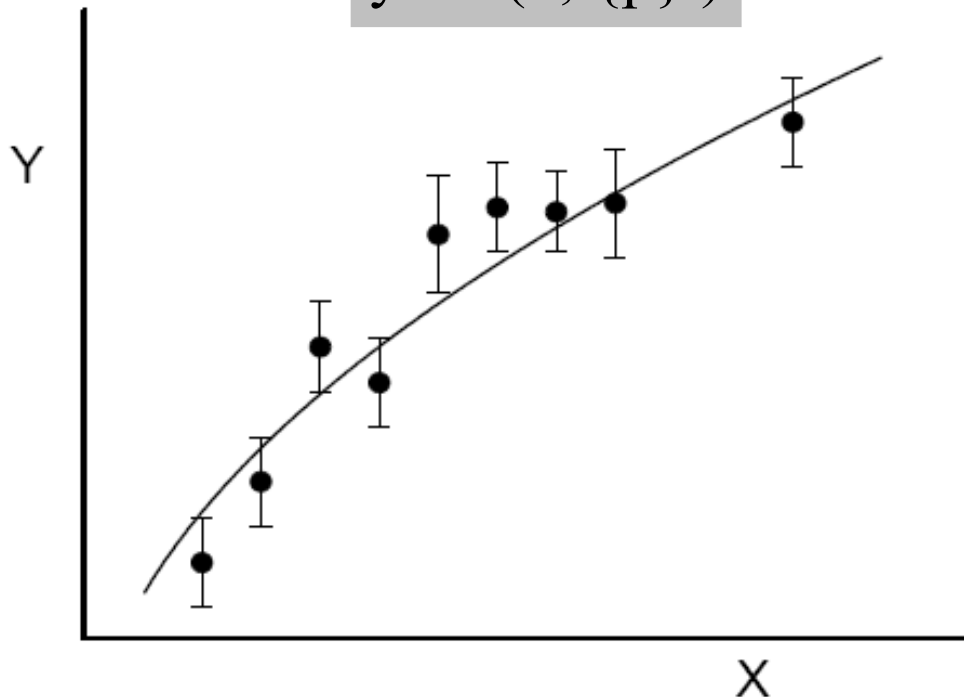
Parameterschätzung mit der Methode der kleinsten Quadrate („ χ^2 -Methode“)

Entwickelt zu Beginn des 19. Jahrhunderts von Legendre, Gauß und Laplace.

Ziel: finden der besten Funktion, die Fehler-behaftete Datenpunkte annähert.

zunächst für Geraden (Ausgleichsgerade oder lineare Regression), aber
mit numerischen Minimierungsalgorithmen (z.B. MINUIT in ROOT)
auf beliebige Funktionen und Anzahl Parameter anwendbar !

$$y = f(x; \{p\})$$



Minimiere bzgl. der Parameter $\{p\}$

$$S = \chi^2 = \sum_{i=1}^N \frac{(y_i - f(x_i, \{p\}))^2}{\sigma_i^2}$$

σ_i sind die Varianzen der y_i

S: „Summe der Residuen-Quadrate“

Falls die Fehler korreliert sind,

ersetze $1/\sigma_i^2 \rightarrow \text{cov}^{-1}$ (Inverse der
Kovarianzmatrix)

Parameterschätzung mit kleinsten Quadraten

Aufgabenstellung:

Mehrere Messungen y_i , $i=1, \dots, n$ mit Fehlern σ_i

Ein Modell $y_i = f(x_i; \mathbf{a})$ mit Vektor $\mathbf{a}$ von p Parametern a_j , $p < n$ macht Vorhersagen der Messungen.

Vektoriell geschrieben: $\mathbf{y} = f(\mathbf{x}; \mathbf{a})$

Gesucht ist der beste Schätzwert der Parameter a_j
dazu wird

$$S = \chi^2 = \sum_{i=1}^N \frac{(y_i - f(x_i, \mathbf{a}))^2}{\sigma_i^2} \quad \text{bzgl. } \mathbf{a} \text{ minimiert.}$$

Problem ist analytisch lösbar für lineare Modelle, d.h. $f(x_i, \mathbf{a}) = A(\mathbf{x}) \mathbf{a}$
mit $n \times p$ Matrix $A(\mathbf{x})$, wobei die Elemente $A_{ij}(\mathbf{x})$ Funktionen der x_i sind.

Im allgemeinen: numerische Minimierung von χ^2

Parameterschätzung mit kleinsten Quadraten

Beispiele ($\Delta y_i := (y_i - f(x_i))$):

Identische Fehler:
$$S = \sum_{i=1}^n \Delta y_i^2$$

Unterschiedliche Fehler:
$$S = \sum_{i=1}^n \frac{\Delta y_i^2}{\sigma_i^2}$$

Korrelierte Messungen mit Kovarianzmatrix V :
$$S = \Delta \mathbf{y}^T V^{-1} \Delta \mathbf{y}$$

z.B. Mittelwert von Messungen y_i :
$$S = \sum_{i=1}^n (y_i - \bar{y})^2 \quad \text{minimieren}$$
$$\Rightarrow \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

Lösung des linearen Optimierungsproblems

Lineares Problem $y_i = \sum_j f_j(x_i) \cdot a_j = A_{ij} a_j \Leftrightarrow \mathbf{y} = \mathbf{A} \mathbf{a}$

mit Kovarianzmatrix $\mathbf{V}$

minimiere $S(\mathbf{a}) = (\mathbf{y} - \mathbf{A} \mathbf{a})^T \mathbf{V}^{-1} (\mathbf{y} - \mathbf{A} \mathbf{a})$ bzgl. $\mathbf{a}$, d.h. $\frac{\partial S}{\partial \mathbf{a}} = 0$

mit $\mathbf{V}^{-1} =: \mathbf{W}$ als „Gewichtsmatrix“

Lösung $\hat{\mathbf{a}}$ ist lineare Funktion der Messwerte:

$$\hat{\mathbf{a}} = (\mathbf{A}^T \mathbf{V}^{-1} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{V}^{-1} \mathbf{y}$$

Kovarianzmatrix durch Fehlerfortpflanzung:

$$\mathbf{V}(\hat{\mathbf{a}}) = (\mathbf{A}^T \mathbf{V}^{-1} \mathbf{A})^{-1}$$

Beobachtung: hängt zusammen mit der Inversen der Matrix der 2. Ableitungen:

$$(\mathbf{V}^{-1}(\hat{\mathbf{a}}))_{ij} = \frac{1}{2} \frac{\partial^2 S}{\partial a_i \partial a_j}$$

Lineare Regression als Spezialfall

Anpassung einer Geraden, $f(x) = p_1 + p_2 x$ an Messdaten mit unkorrelierten Fehlern:

$$A = \begin{pmatrix} 1 & x_1 \\ \dots & \dots \\ 1 & x_N \end{pmatrix} \quad W = \begin{pmatrix} \frac{1}{\sigma_1^2} & 0 & \dots & 0 & 0 \\ 0 & \dots & \frac{1}{\sigma_i^2} & \dots & 0 \\ 0 & 0 & \dots & 0 & \frac{1}{\sigma_N^2} \end{pmatrix}$$

Durch Einsetzen in die Lösungsformeln von vorhin erhält man mit den Abkürzungen

$$\begin{aligned} S_1 &= \sum_{i=1}^N \frac{1}{\sigma_i^2}, & S_x &= \sum_{i=1}^N \frac{x_i}{\sigma_i^2} = \bar{x} S_1, & S_y &= \sum_{i=1}^N \frac{y_i}{\sigma_i^2} = \bar{y} S_1, \\ S_{xx} &= \sum_{i=1}^N \frac{x_i^2}{\sigma_i^2} = \overline{x^2} S_1, & S_{xy} &= \sum_{i=1}^N \frac{x_i y_i}{\sigma_i^2} = \overline{xy} S_1, & D &= S_1 S_{xx} - S_x^2 \end{aligned}$$

die **Lösung**

$$\begin{aligned} \hat{p}_1 &= \frac{S_{xx} S_y - S_x S_{xy}}{D}, & \sigma_{p_1}^2 &= \frac{S_{xx}}{D}, \\ \hat{p}_2 &= \frac{S_1 S_{xy} - S_x S_y}{D}, & \sigma_{p_2}^2 &= \frac{S_1}{D}, & V_{12} &= \frac{-S_x}{D}. \end{aligned}$$

Lineare Regression: eine Anmerkung

Das Kovarianzelement V_{12} verschwindet für $S_x=0$,

d.h. wenn der Erwartungswert der Abszissenwerte 0 ist

Dies lässt sich durch Änderung der Parametrisierung erreichen:

$$\text{setze } x' = x - \bar{x} \quad \rightarrow \quad f'(x) = p'_1 + p'_2 (x - \bar{x})$$

Man erhält dann diese einfacheren, unkorrelierten Lösungen für die Parameter

$$\begin{aligned} \hat{p}'_1 &= \frac{S_y}{S_1} = \bar{y}, & \sigma_{p'_1}^2 &= \frac{1}{S_1}, \\ \hat{p}'_2 &= \frac{S_{x'y}}{S_{x'x'}} = \frac{\overline{x'y}}{\overline{x'^2}}, & \sigma_{p'_2}^2 &= \frac{1}{S_{x'x'}} \end{aligned}$$

Beispiel: Mittelwert korrelierter Messungen

zwei Messungen der gleichen Größe m_0 mit unabhängigen ($u_{1,2}$) und gemeinsamen Fehlern (c): $m_{1,2} \pm \sigma_{1,2}$; $\sigma_{1,2} = u_{1,2} \oplus c$

Inverse Kovarianzmatrix:
$$V^{-1} = \begin{pmatrix} \sigma_1^2 & c^2 \\ c^2 & \sigma_2^2 \end{pmatrix}^{-1} = \frac{1}{\sigma_1^2 \sigma_2^2 - c^4} \begin{pmatrix} \sigma_2^2 & -c^2 \\ -c^2 & \sigma_1^2 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sigma_1^2 - \frac{c^4}{\sigma_2^2}} & \frac{-c^2}{\sigma_1^2 \sigma_2^2 - c^4} \\ \frac{-c^2}{\sigma_1^2 \sigma_2^2 - c^4} & \frac{1}{\sigma_2^2 - \frac{c^4}{\sigma_1^2}} \end{pmatrix}$$

$$\chi^2(m_0) = \sum_{i,j} (m_i - m_0) V_{ij}^{-1} (m_j - m_0) = \Delta \vec{m}^T V^{-1} \Delta \vec{m}$$

χ^2 -Minimierung:
$$\frac{\partial}{\partial m_0} \chi^2 = -\vec{1}^T V^{-1} \Delta \vec{m} - \Delta \vec{m}^T V^{-1} \vec{1} = -2 \sum_{i,j} V_{ij}^{-1} (m_j - m_0) = 0$$

$$\Rightarrow m_0 = \frac{\sum_{i,j} V_{ij}^{-1} m_j}{\sum_{i,j} V_{ij}^{-1}} = \frac{\frac{1}{\sigma_1^2} m_1 + \frac{1}{\sigma_2^2} m_2}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}} \quad \text{mit} \quad \sigma_i^2 = \sigma_i^2 - \sigma_{c,i}^2$$

$1/\sigma^2$ - gewichtete
Summe der Einzel-
Messungen

gilt *allgemein* in n Dimensionen

Zusammenhang von S und χ^2

$$S = \chi^2 = \sum_{i=1}^N \frac{(y_i - f(x_i, \{p\}))^2}{\sigma_i^2}$$

N Messungen
k Parameter

S, die „gewichtete Summe der Residuenquadrate“, folgt bei Gauß-förmig verteilten Fehlern σ_i einer χ^2 -Verteilung mit $n_f = N - k$ Freiheitsgraden.

Erwartungswert: $\langle \chi^2 \rangle = n_f$ oder $\langle \chi^2 / n_f \rangle = 1$

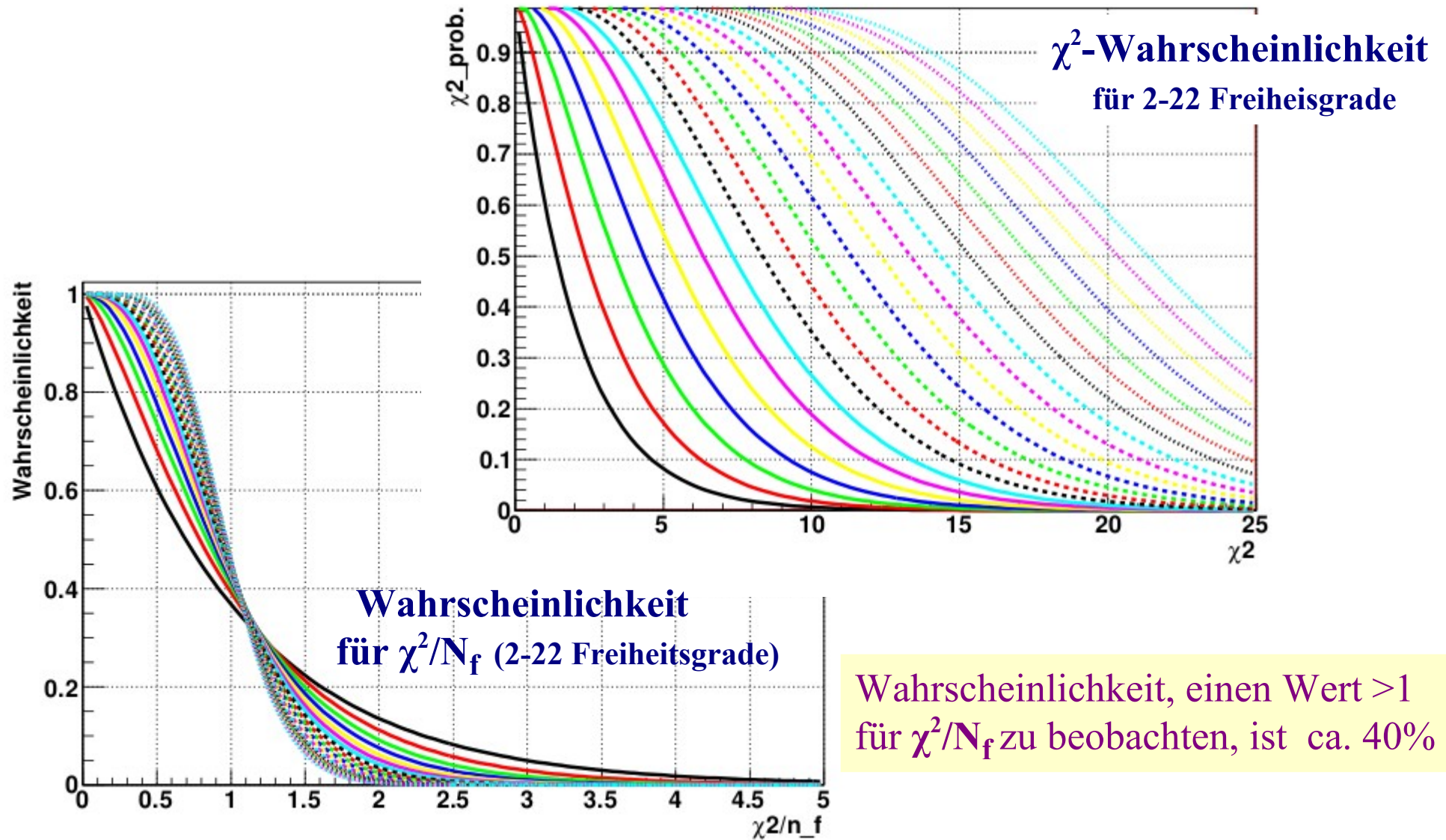
Die **χ^2 -Wahrscheinlichkeit**

$$\chi^2_{\text{prob}} = \int_{\chi_{\min}}^{\infty} \chi^2(s; n_f) ds = 1. - \int_0^{\chi_{\min}} \chi^2(s; n_f) ds$$

dient zur Quantifizierung der Qualität einer Anpassung

Aussage, mit welcher Wahrscheinlichkeit ein größerer Wert von χ^2 am Minimum als der tatsächlich beobachtete zu erwarten wäre.

Zusammenhang von S und χ^2 : χ^2 -Wahrscheinlichkeit



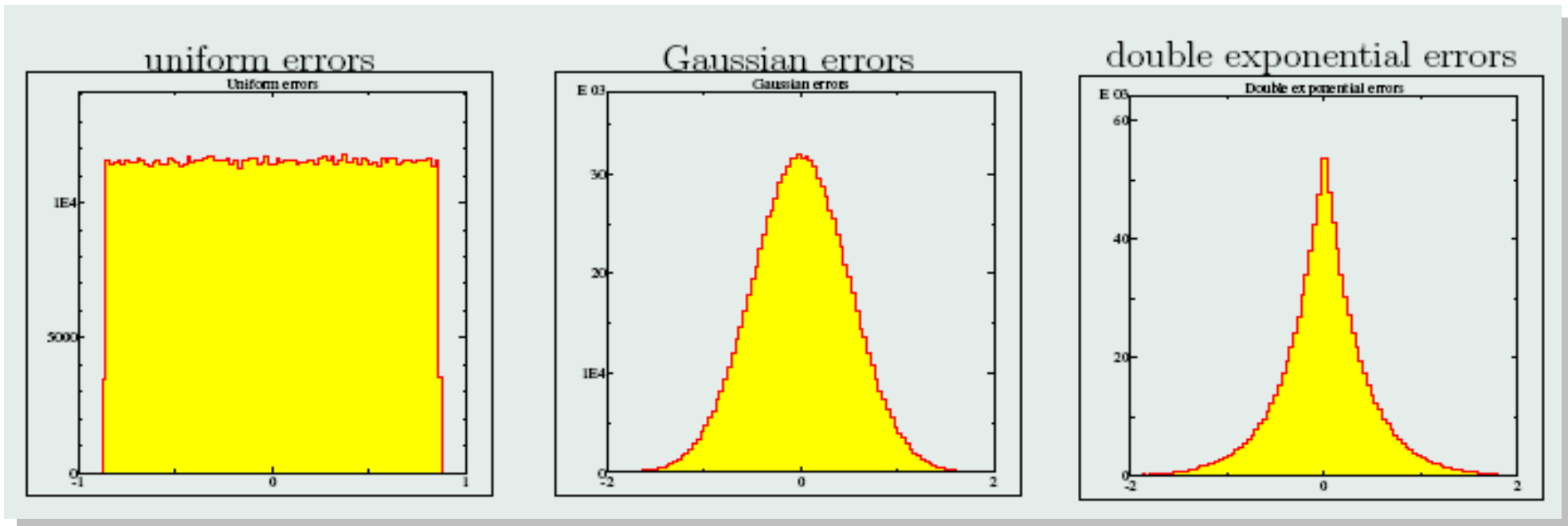
Wahrscheinlichkeit, einen Wert >1 für χ^2/N_f zu beobachten, ist ca. 40%

s. Skript `chi2PDF.C`

Abhängigkeit von der Messfehler-Verteilung

Anpassung einer Geraden an 20 Datenpunkte ($\text{ndf}=20-2=18$)

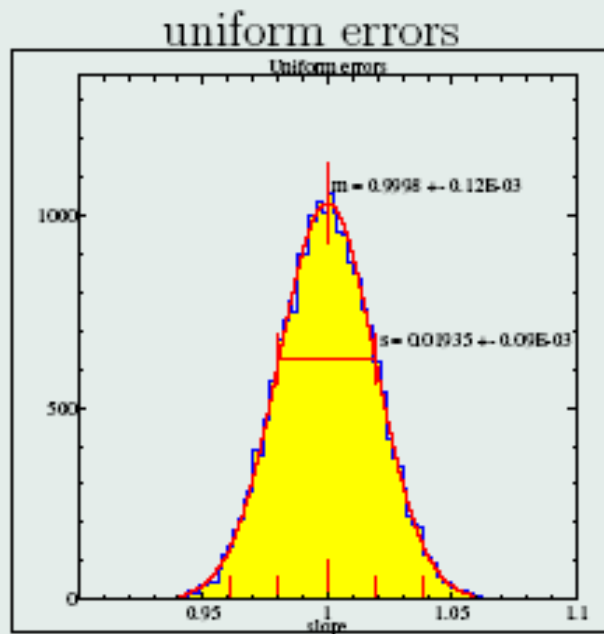
Betrachten 3 verschiedene Verteilungen der Einzelmessungen,
(alle mit Mittelwert 0 und Standardabweichung=0.5)



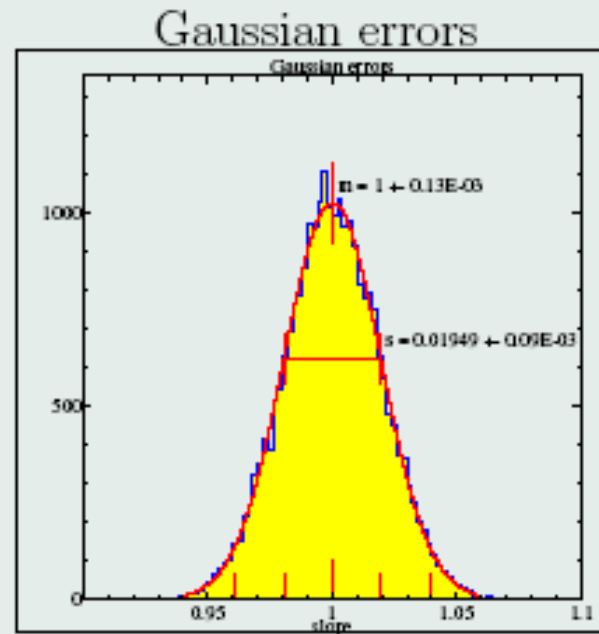
Parameterschätzung: Abhängigkeit von der Messfehler-Verteilung

25000 Monte-Carlo-Tests:

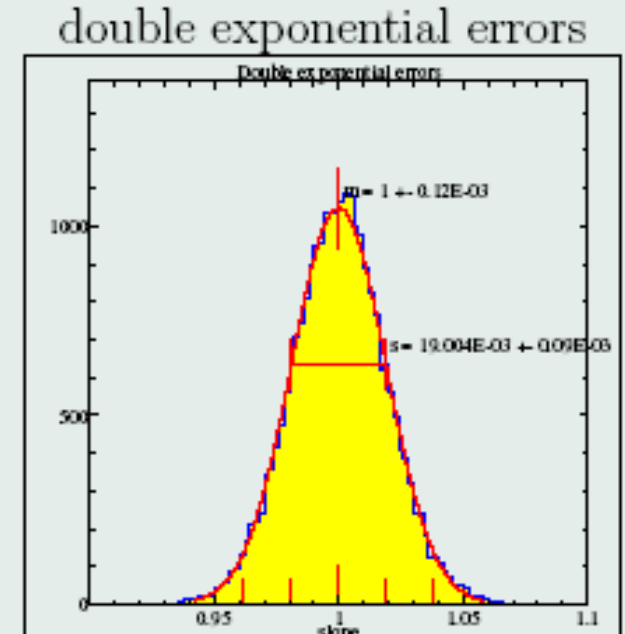
Alle Parameter-Verteilungen sind Gaußisch, die Breite kompatibel zur Erwartung aus Fehlerfortpflanzung (für beide Parameter)



$$\sigma = 0.0194$$



$$\sigma = 0.0195$$

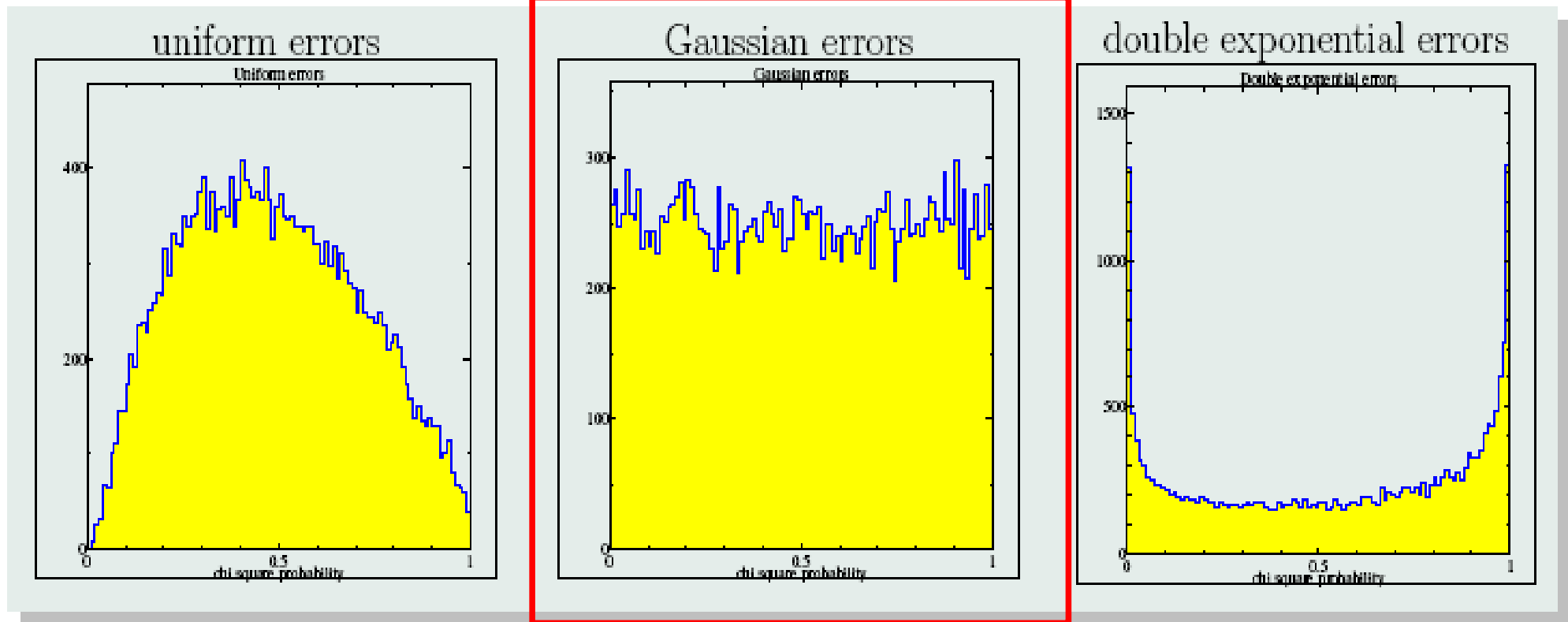


$$\sigma = 0.0190$$

Methode der kleinsten Quadrate funktioniert, wenn Mittelwert und Standardabweichung der Daten bekannt sind (genaue Form der Verteilung unerheblich!)

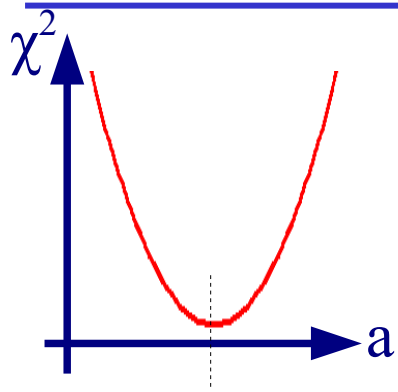
χ^2 -Wahrscheinlichkeit: Abhängigkeit von der Messfehler-Verteilung

Berechnen jeweils die χ^2 -Wahrscheinlichkeit, d.h. $\chi^2_{\text{prob}} = \int_S^\infty \chi^2(s, ndf) ds$



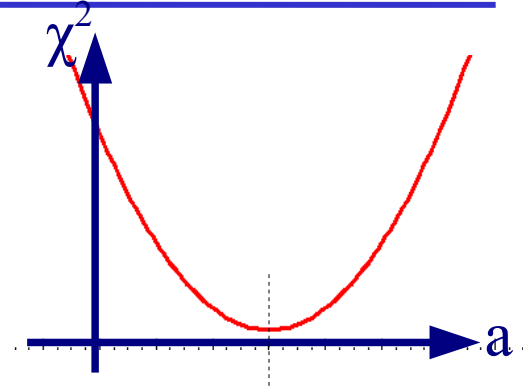
Bei Gauß'schen Messfehlern gilt **$\chi^2\text{-prob} = \text{const.}$** , d.h. S folgt einer χ^2 -Verteilung
andernfalls ergeben sich andere Verteilungen !

Bestimmung der Parameterfehler



scharfes Minimum: große Krümmung

Je schärfer das Minimum von $\chi^2(\mathbf{a})$,
desto kleiner die Parameterfehler:



flaches Minimum: kleine Krümmung

→ Fehler der Parameter a_i sind umgekehrt proportional zur Krümmung
von $\chi^2(\mathbf{a})$ am Minimum

$$\frac{1}{\sigma_{\hat{a}}^2} = \frac{1}{2} \left. \frac{\partial^2 \chi^2(a)}{\partial a^2} \right|_{\hat{a}} \quad \text{bzw.} \quad (V_{\hat{\mathbf{a}}}^{-1})_{ij} = \frac{1}{2} \left. \frac{\partial^2 \chi^2(\mathbf{a})}{\partial a_i \partial a_j} \right|_{\hat{a}_i \hat{a}_j}$$

bei mehreren Parametern .

Beweis nicht ganz einfach, s. auch später bei Likelihood-Methode

Fehler auf die Parameter ohne Datenfehler ? NEIN!

Manche Programme zur Anpassung (QTIplot, Origin usw.) geben bei χ^2 -Anpassungen Parameterfehler aus, ohne dass Fehler auf die Datenpunkte angegeben wurden. **Wie geht das?**

Annahmen:

- Modell beschreibt die Daten perfekt: $\chi^2 = n_f$ statt $\langle \chi^2 \rangle = n_f$
- Alle Datenpunkte haben den gleichen Fehler: $\sigma_i = \sigma$ (nur selten so !)

$$S' = \sum_{i=1}^N (y_i - f(x_i; \hat{p}))^2 =: n_f \cdot \sigma^2 \text{ d.h. setze alle } \sigma_i = \sqrt{\frac{S'}{n_f}}$$

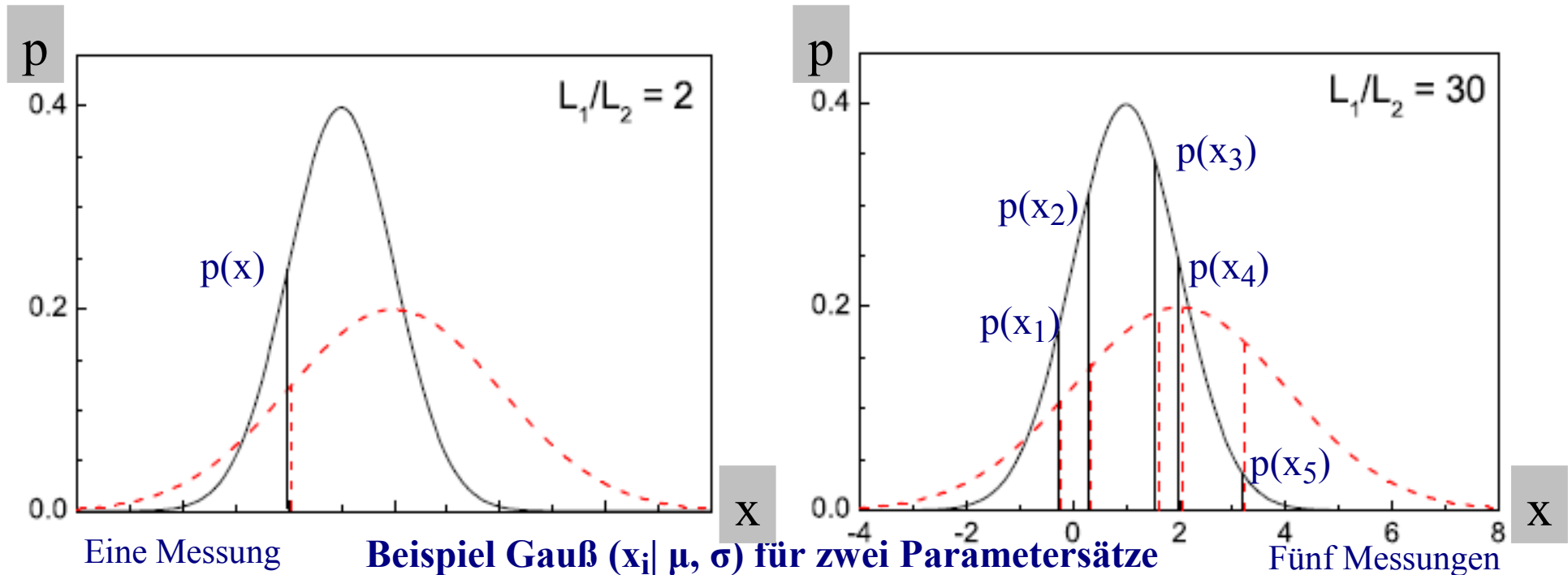
dann $\chi^2 = \sum_{i=1}^N \frac{(y_i - f(x_i; p))^2}{S'/n_f}$ setzen und Parameterfehler bestimmen

Verlust der χ^2 -Wahrscheinlichkeit, angenommenes Fehlermodell oft falsch!

Maximum Likelihood – Methode

Maximum Likelihood Parameterschätzung

Mehrere unabhängige Messgrößen x_i , $i=1, \dots, n$ einer Größe folgen einer Verteilungsdichte $P(\mathbf{x}, \mathbf{a})$ mit Parametern $\mathbf{a}_i$.
Als pdf ist $p(\mathbf{x}; \mathbf{a})$ positiv und normiert (bzgl. $\mathbf{x}$, aber nicht bzgl. $\mathbf{a}$!)



„Likelihood“

ist das Produkt der Wahrscheinlichkeiten $p(x_i | \mathbf{a}_i)$ aller Messungen

Maximum Likelihood- Prinzip

Likelihood-Funktion:

Produkt der Werte der Wahrscheinlichkeitsdichte, P_i , für
n unabhängige Messungen x_i :

$$\mathcal{L}(\mathbf{a}) = P(x_1|\mathbf{a}) \cdot P(x_2|\mathbf{a}) \cdot \dots \cdot P(x_n|\mathbf{a}) = \prod_{i=1}^n P(x_i|\mathbf{a})$$

hängt nur noch von den Parametern $\mathbf{a}$ ab !

Maximum-Likelihood-Prinzip:

Der beste Schätzwert für den Parametervektor $\hat{\mathbf{a}}$ ist der, der die Likelihood-Funktion $\mathcal{L}(\mathbf{a})$ maximiert

Maximum Likelihood in der Praxis

Technische und theoretische Gründe:

Minimiere den negativen Logarithmus der Likelihood-Funktion:

$$F(\mathbf{a}) = -\ln \mathcal{L}(\mathbf{a}) = -\sum_{i=1}^n \ln P(x_i | \mathbf{a})$$

Likelihood-Gleichung definiert den Schätzwert $\hat{\mathbf{a}} : \frac{dF(\mathbf{a})}{da_j} = 0$

Kombination (unkorrelierter !) Messungen oder Experimente einfach:

$$\mathcal{L}(\mathbf{a}) = \mathcal{L}_1(\mathbf{a}) \cdot \mathcal{L}_2(\mathbf{a}) \quad \text{bzw.} \quad F(\mathbf{a}) = F_1(\mathbf{a}) + F_2(\mathbf{a})$$

multipliziere
Likelihood-Funktionen

addiere
log-Likelihood-Funktionen

Maximum Likelihood: Fehlerabschätzung

$F(a)$ näherungsweise quadratisch um das Minimum;

$\pm 1\text{-}\sigma\text{-Intervall}$ aus $\Delta F = \pm 0.5$

Erste Ableitung näherungsweise linear, $=0$ am Minimum

Zweite Ableitung $\sim$ konstant;

Varianz = $1/\text{Krümmung}$

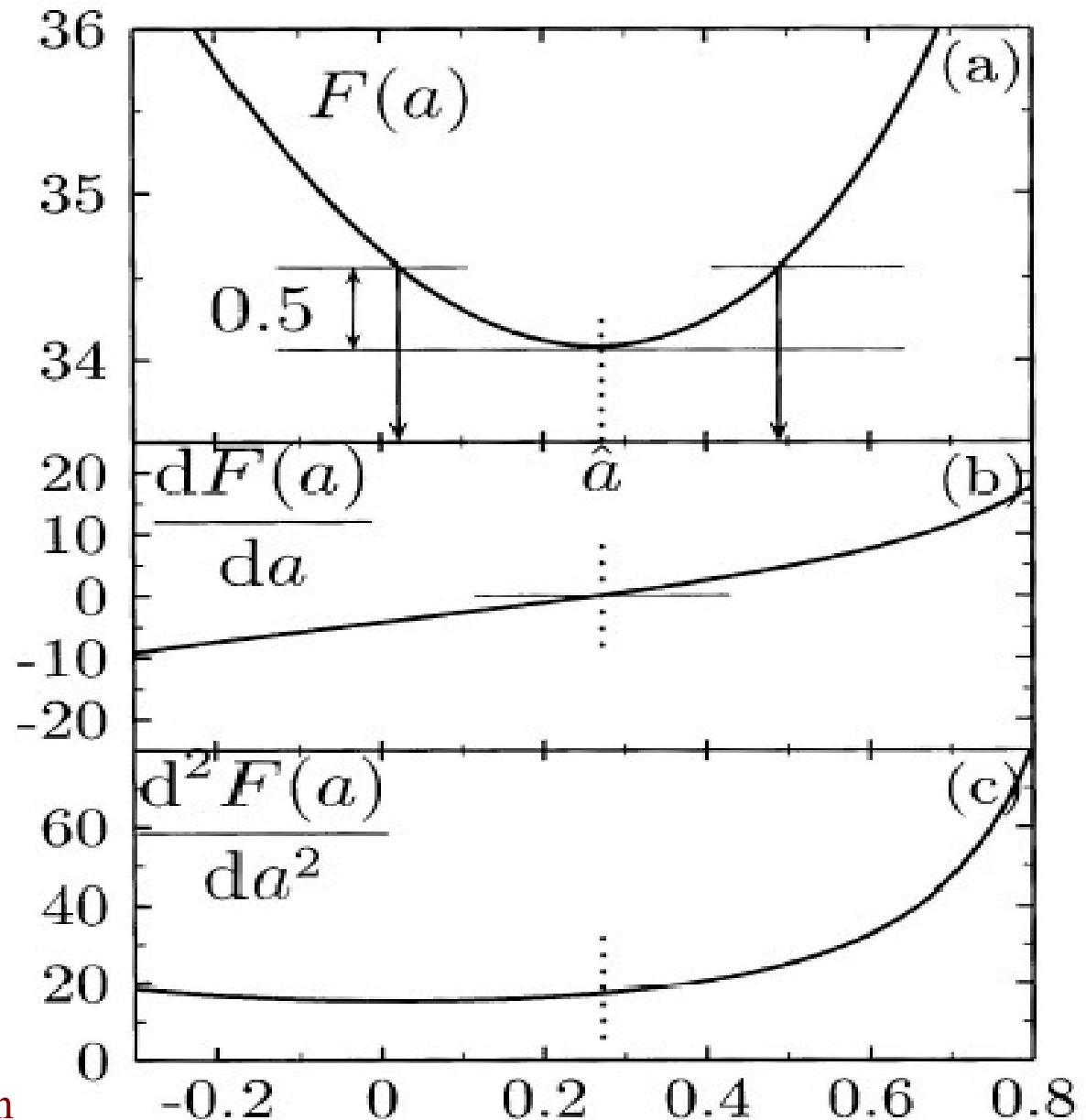
$$1/\sigma^2 = \partial^2 F / \partial a^2$$

bei mehreren Parametern a_i :

$$(\text{cov}^{-1})_{ij} = \partial^2 F / \partial a_i \partial a_j$$

Nur für Parabel-förmigen Verlauf von $F(a)$
sind die beiden Fehlerdefinitionen äquivalent

Mathematisch exakt: die oben angegebenen
Fehlerabschätzungen sind Untergrenzen bei der lnL-Methode



Maximum Likelihood: Fehlerabschätzung (2)

Plausibilitätserklärung (kein Beweis, Stichwort „Cramer-Rao-Frechet Grenze“)

nur ein Parameter a , betrachten Taylor-Entwicklung von $F(a)$ um Minimum:

$$F(a) = F(\hat{a}) + \frac{1}{2}(a - \hat{a})^2 \frac{\partial^2 F}{\partial a^2} + \dots \quad \text{näherungsweise parabelförmig}$$

Likelihood = $\exp(-F(a))$ als Verteilungsdichte in a auffassen:

$$\mathcal{L}(a) = \exp(-F(a)) \simeq A \exp \left(-\frac{1}{2}(a - \hat{a})^2 \frac{\partial^2 F(a)}{\partial a^2} \bigg|_{a=\hat{a}} \right)$$

ist Gauß-Verteilung (mit Normierungsfaktor A)

und Standardabweichung gegeben durch $\frac{1}{\sigma^2} = \frac{\partial^2 F(a)}{\partial a^2} \bigg|_{a=\hat{a}}$

weiter gilt mit dieser Beziehung für σ : $F(a) - F(\hat{a}) \simeq \frac{1}{2} \frac{(a - \hat{a})^2}{\sigma^2}$

$$\text{d. h. } |a - \hat{a}| = \sigma \text{ wenn } F(a) - F(\hat{a}) = \frac{1}{2}$$

Maximum Likelihood: Fehlerabschätzung (3)

zur Fehlerbestimmung aus $F(a)$:

$$F(a) - F(\hat{a})_{min} = \frac{1}{2} \quad \Leftrightarrow \quad |a - \hat{a}| = 1\sigma$$

$$F(a) - F(\hat{a})_{min} = \frac{n^2}{2} \quad \Leftrightarrow \quad |a - \hat{a}| = n\sigma$$

Log-Likelihood-Differenz bestimmt Fehler.

Vorteil: invariant unter Variablentransformation $a \rightarrow a'(a)$:

$$F(a) - F(\hat{a}) = F(a'(a)) - F(a'(\hat{a}))$$

wichtig, wenn $-\ln L$ nicht parabelförmig in der Nähe des Minimums:

Angabe eines asymmetrischen Fehlerintervalls

Bei mehreren Parametern a_j : zur Fehlerbestimmung von a_i muss die Likelihood bzgl. aller anderen Parameter a_j ; $j \neq i$ minimiert werden („**Profile-Likelihood**“)
(MINOS-Verfahren in MINUIT)

Zusammenhang $-\ln L$ und χ^2

Für **Gauß-förmig** um $f(x_i; \mathbf{a})$ verteilte Messungen y_i ist die χ^2 Methode äquivalent zur $-\ln L$ -Methode :

$$P(y_i|\vec{a}) = \frac{1}{\sqrt{2\pi}\sigma_i} \cdot \exp\left\{-\frac{(y_i - f(x_i, \vec{a}))^2}{2\sigma_i^2}\right\}$$

$$-\ln L(\vec{y}|\vec{a}) = \underbrace{\frac{1}{2} \sum_i \left(\frac{y_i - f(x_i, \vec{a})}{\sigma_i} \right)^2}_{\chi^2} + \underbrace{\sum_i \ln(\sqrt{2\pi}\sigma_i)}_{\text{const. bzgl. } \mathbf{a}}$$

Minimieren von $-\ln L \leftrightarrow$ Minimieren von χ^2

$$\Delta(-\ln L) = \frac{1}{2} \Delta\chi^2$$

$$\partial^2(-\ln L) / \partial a_i \partial a_j = \frac{1}{2} \partial^2 \chi^2 / \partial a_i \partial a_j$$

Fehlerbestimmung:

	$ \Delta(-\ln L) $	$\Delta\chi^2$
1σ	0.5	1
2σ	2.0	4
3σ	4.5	9
$n\sigma$	$n^2/2$	n^2

Bei **anderen als Gauß-förmigen** Fehlerverteilungen ist χ^2 eine eigenständige Methode;

- bei unbekannter Fehlerverteilung haben wir keine bessere
- χ^2 ist optimal für die Anpassung von Linearkombinationen von Fit-Funktionen

Beispiel: Mittelwert von normal-verteilten Größen

$$f(x|\mu) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left[-\frac{(x - \mu)^2}{2\sigma^2} \right],$$

$$L(\mu) = \prod_{i=1}^N \frac{1}{\sqrt{2\pi}\sigma} \exp \left[-\frac{(x_i - \mu)^2}{2\sigma^2} \right],$$

$$\ln L(\mu) = -N \left[\frac{1}{2} \ln 2\pi + \ln \sigma \right] - \sum_{i=1}^N \frac{(x_i - \mu)^2}{2\sigma^2}.$$

Nullsetzen der
1. Ableitung:

$$\sum_{i=1}^N \frac{(x_i - \hat{\mu})}{\sigma^2} = 0$$

$$\Rightarrow \sum_{i=1}^N x_i = N\hat{\mu}$$

Bekannte Formel für den Mittelwert !

2. Ableitung zur Fehlerbestimmung

$$\frac{1}{\sigma_{\mu}^2} = \left. \frac{\partial^2 (-\ln \mathcal{L})}{\partial \mu^2} \right|_{\mu=\hat{\mu}} = \sum_{i=1}^N \frac{1}{\sigma^2}$$

$$\Rightarrow \sigma_{\hat{\mu}} = \frac{\sigma}{\sqrt{N}}$$

**Der Fehler des Mittelwertes ist die
Breite der Verteilung dividiert durch $\sqrt{N}$**

Maximum Likelihood Fit des Parameters p einer Binomialverteilung

Erinnerung:

Wahrscheinlichkeit für die Beobachtung eines bestimmten Ereignisses ist p .

Wahrscheinlichkeit, bei n Versuchen k solcher Ereignisse zu beobachten

ist gegeben durch die **Binomialverteilung** [$P(k) \sim p^k (1-p)^{n-k}$]

Eine Münze wird N mal geworfen.

Was kann man über die Wahrscheinlichkeit p ,
dass „Kopf“ geworfen wird, sagen?

0 Versuche: Wir wissen nichts, p flach in $[0,1]$.

1. Versuch: Kopf $L_1(p) = p$, $p(\text{Zahl})=0$

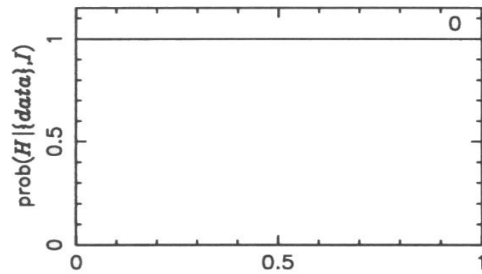
2. Versuch: Kopf $L_2(p) = L_1 p = p^2$

3. Versuch: Zahl $L_3(p) = L_2 (1-p) = p^2 (1-p) = p^3 - p^2$

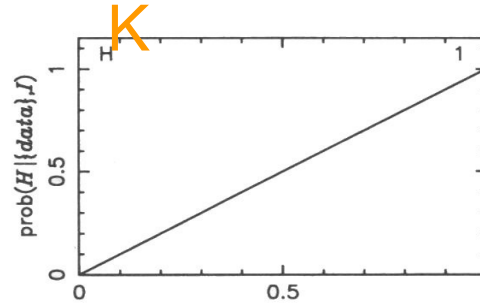
3. Versuch: Zahl $L_4(p) = L_3 (1-p) = p^2 (1-p)^2 = p^4 - 2p^3 + p^2$

...

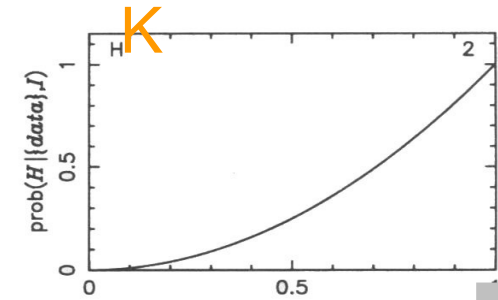
n . Versuch: p wird immer genauer, im Grenzfall $n \rightarrow \infty$ Gauß-förmig verteilt.



$L_0(p_{\text{Kopf}})$

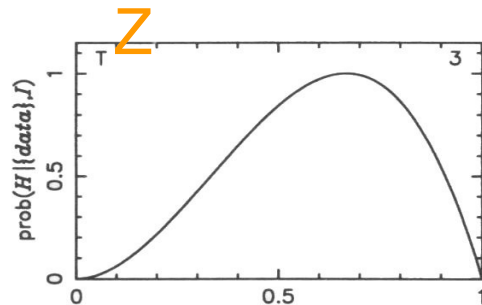


$L_1(p_{\text{Kopf}})$

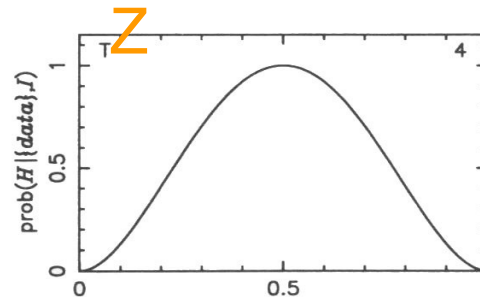


$L_2(p_{\text{Kopf}})$

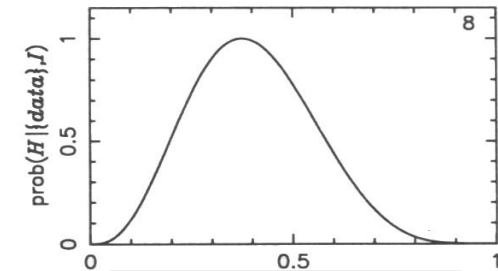
→ p



$L_3(p_{\text{Kopf}})$

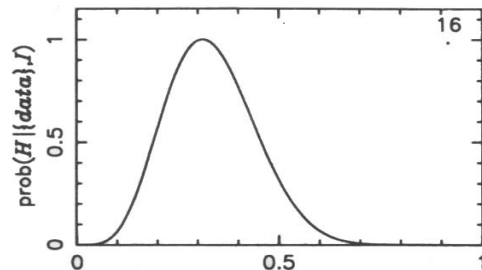


$L_4(p_{\text{Kopf}})$

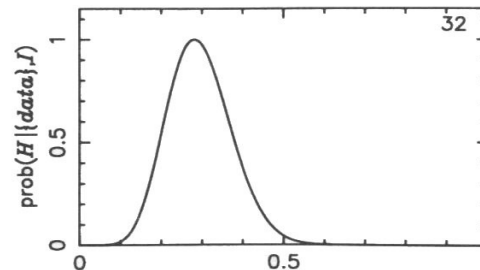


$L_8(p_{\text{Kopf}})$

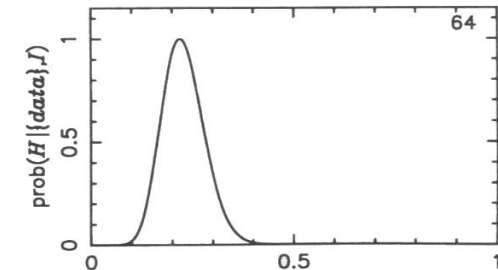
→ p



$L_{16}(p_{\text{Kopf}})$

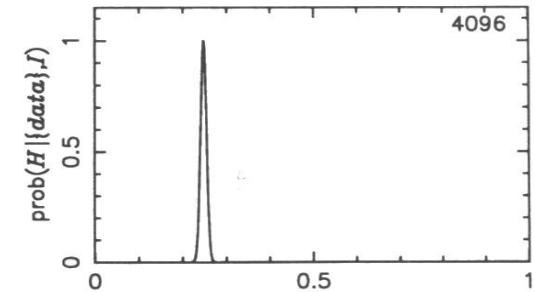
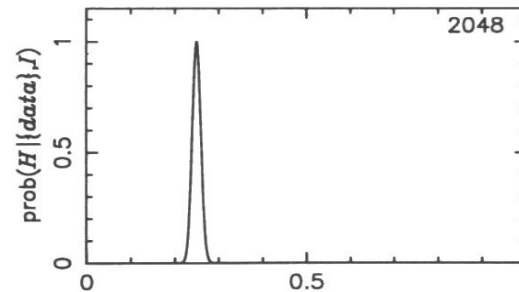
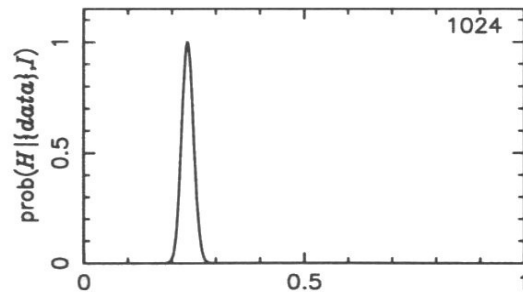
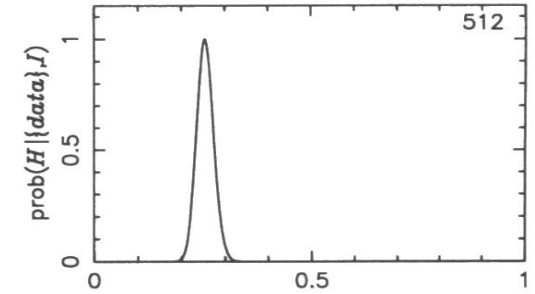
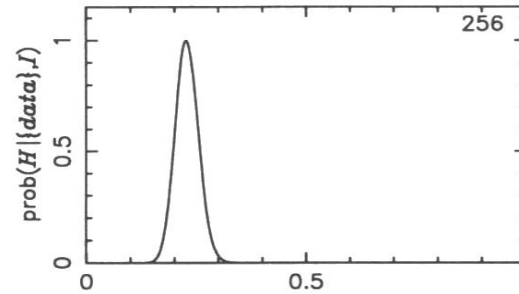
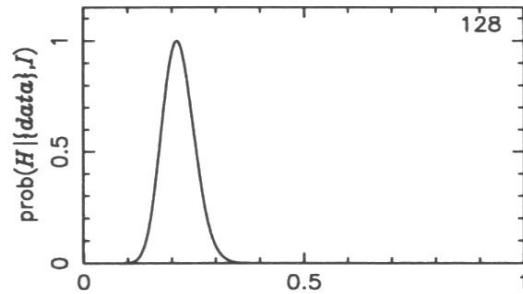


$L_{32}(p_{\text{Kopf}})$



$L_{64}(p_{\text{Kopf}})$

→ p



Fehler wird mit $1/\sqrt{n}$ kleiner

Münze ist falsch !

Maximum Likelihood-Binomialfit

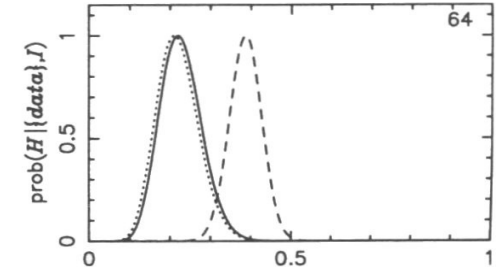
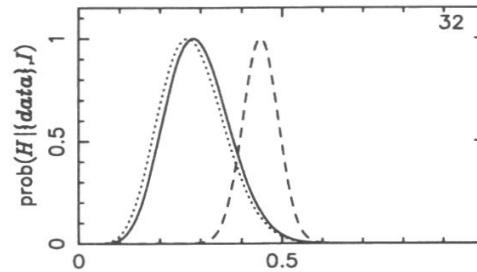
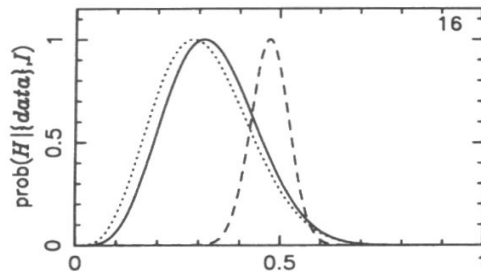
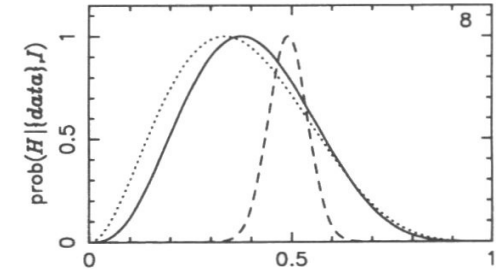
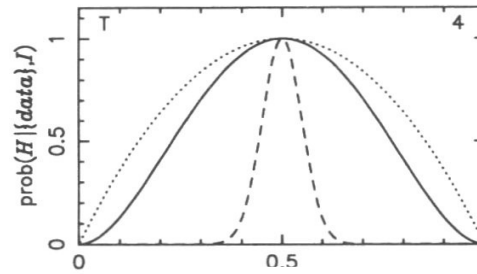
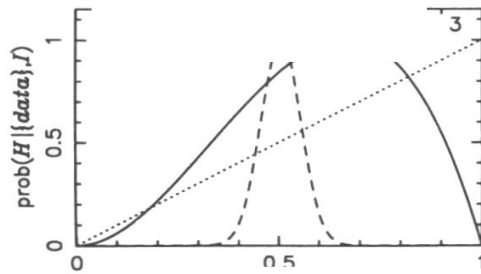
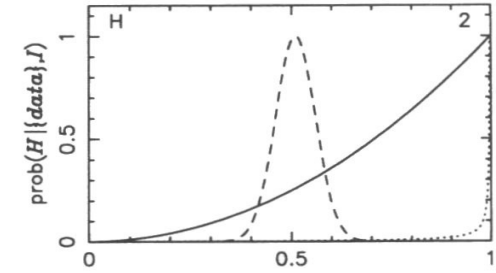
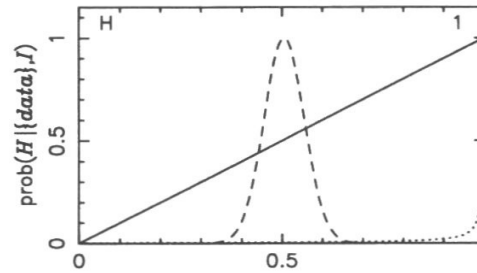
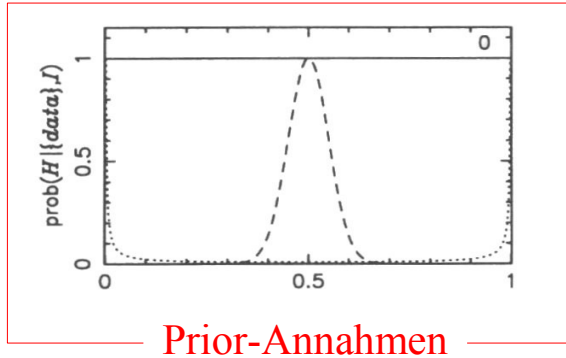
mit verschiedenen a priori-Annahmen

Verschiedene a priori-Annahmen:

1. Wir wissen nichts p flach in $[0,1]$
2. Die Münze ist wahrscheinlich ok $\rightarrow$ Gauß um 0.5
3. Die Münze ist wahrscheinlich gefälscht,
ich weiß aber nicht, in welche Richtung.

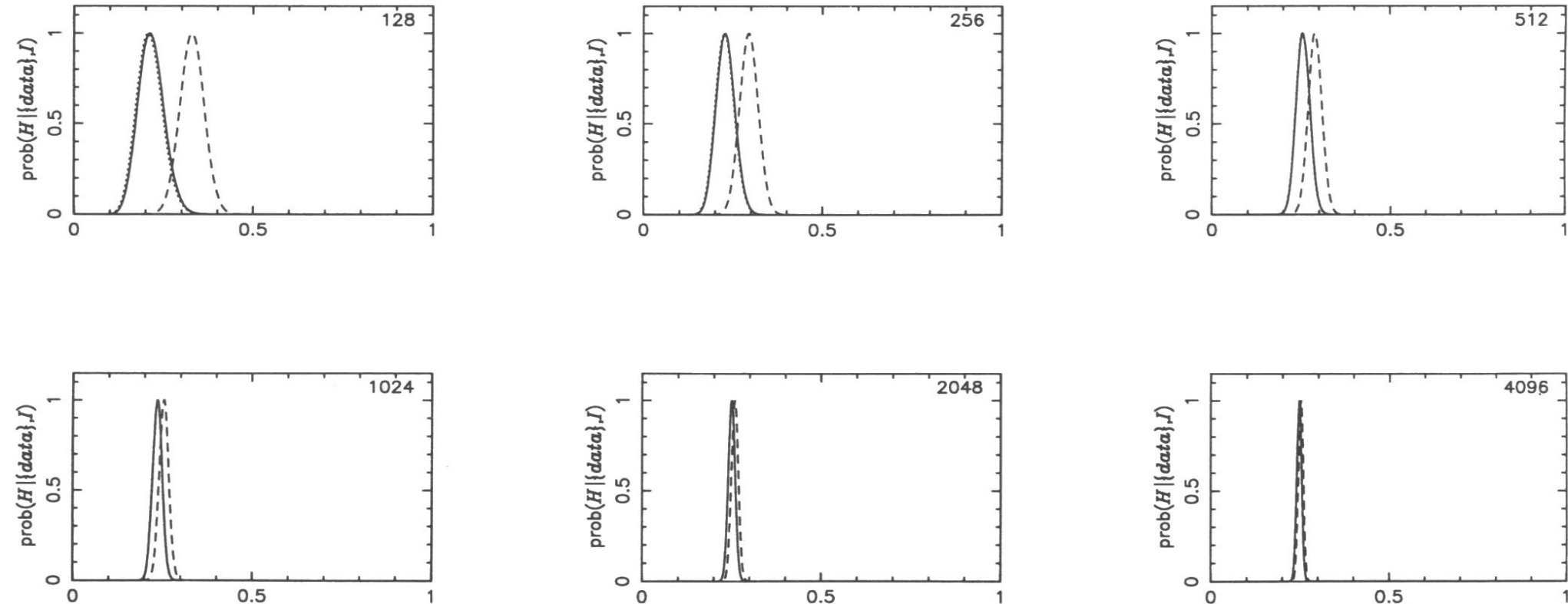
Maximum Likelihood-Binomialfit mit verschiedenen a priori Annahmen

Maximiere $L(x_i | p) * f_{\text{prior}}(p)$



Maximum Likelihood-Binomialfit mit verschiedenen a priori

Annahmen



Fehler wird mit $1/\sqrt{n}$ kleiner

Münze ist (auch jetzt noch) falsch !

Einfluss des Priors bei hoher Statistik vernachlässigbar

Maximum Likelihood vs. Kleinste Quadrate

Vergleich:

	Maximum - Likelihood	Kleinste Quadrate
Voraussetzung	PDF exakt bekannt	Mittelwert und Varianz bekannt
Methode	Höhe der PDF	Abweichung vom Mittelwert der PDF
Effizienz	maximal	maximal bei linearen Problemen
Komplexität	aufwändig, meist nicht-linear	oft linear und exakt lösbar
Robustheit	nein - PDF muss exakt bekannt sein	nein („Ausreißer“)
korrelierte Datenfehler	u.U. kompliziert	einfach über Kovarianzmatrix
Güte der Anpassung	nein	ja: χ^2 -Wahrscheinlichkeit
Bias (dt. Verzerrung)	i.A. ja	nein bei linearen Problemen (wenn Fehler der Daten unverzerrt sind)
Spezialfall	identisch bei Gauß-förmigen Fehlern	

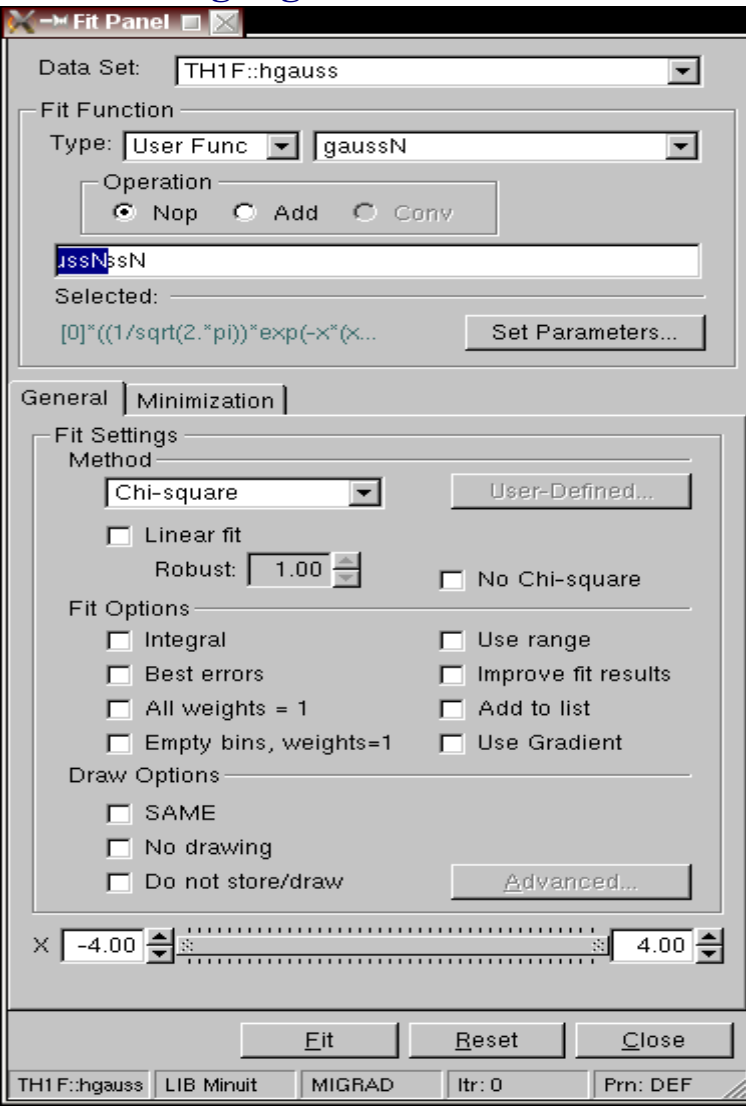
Parameteranpassung mit Root

Parameteranpassung mit ROOT

ROOT enthält einige Minimierungsalgorithmen, u.a. das aus der FORTRAN-Zeit stammende (und nach C++ umgeschriebene) **MINUIT**, entstanden am CERN

- gut getestet und anerkannt

Standard in der Teilchenphysik



Auswahl von Minimierern über die Klasse **TvirtualFitter** (s. später)

In vielen Fällen reicht die von ROOT per GUI zur Verfügung gestellte Funktionalität:

- $-2 \log L$ (!) und χ^2 -Anpassungen an Histogramme
- χ^2 -Anpassungen in Klasse TGraphErrors
- Vordefinierte Funktionen: Polynome bis zum 9. Grad, Gauss, Exponential- und Landauverteilung

↖ χ^2 -Wert

```
FCN=82.016 FROM MIGRAD  STATUS=CONVERGED  12 CALLS1 3 TOTAL
      EDM=9.82506e-16  STRATEGY=1  ERROR MATRIX ACCURATE
EXT PARAMETER                                STEP    FIRST
NO. NAME   VALUE       ERROR    SIZE    DERIVATIVE
1  p0      1.54084e+02  3.51831e+00  1.56525e-02  1.25994e-08
```

Text-Ausgabe von MINUIT

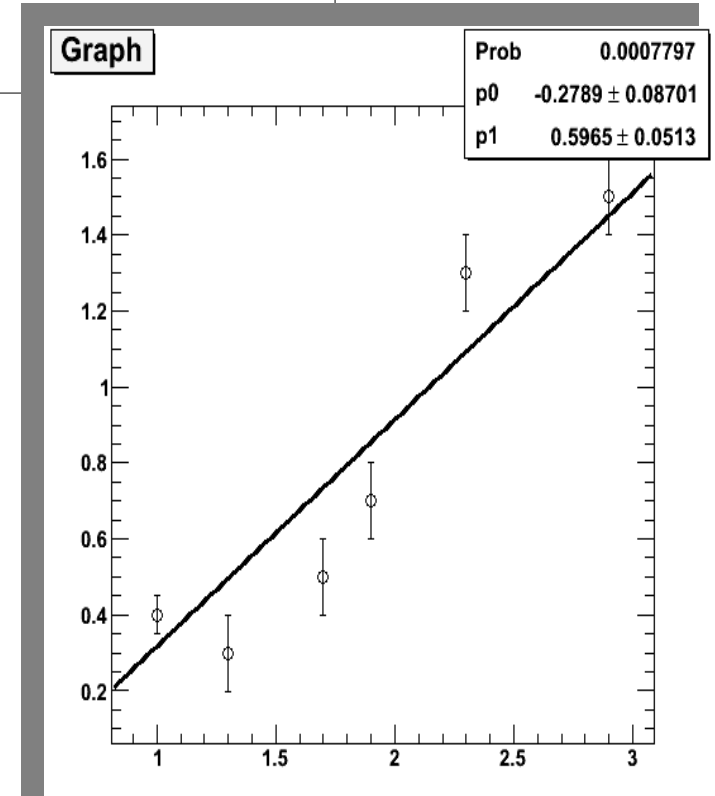
Anpassen mit ROOT GUI

Root-Klassen **TH1**, **TH2** und **TH3** sowie **TGraph** und **TgraphErrors** enthalten eine Methode **.Fit()** zur Funktionsanpassung
(TF1 bzw. TF2 u. TF3)

In der graphischen Darstellung kann durch Rechtsklick das **FitPanel** aktiviert werden, um Anpassungen interaktiv vorzunehmen.

```
root[0]TGraphErrors *gr=new TGraphErrors("ExampleData.txt");  
root[1]gr->Draw("AP");          // Achsen und Daten  
root[2]gr->Fit("pol1");          // Gerade
```

Ein einfaches Beispiel



Parameteranpassung mit ROOT

Man kann auch eigene Fit-Funktionen aus dem Repertoire von ROOT definieren:

```
Root > TF1 *myfit = new TF1("myfit","[0]*sin(x) + [1]*exp(-[2]*x)", 0, 2);  
// set parameters  
Root > myfit->SetParameters(1,0.05,0.2);  
// Fitten  
Root > hist->Fit("myfit");
```

... oder beliebige Funktionen selbst schreiben:

```
Double_t myfunc (Double_t *x, Double_t *par)  
{  
    // IHR CODE !!!  
}  
Root > TF1 *myf = new TF1("myf",myfunc, <min>, <max>, <npar>);  
// set parameters  
Root > myf->SetParameters(<startv_1>, ... , <startv_n>);  
Root > hist->Fit("myf");
```

Im allgemeinsten Fall kann man auch die χ^2 - oder $-2\ln L$ - Funktion selbst vorgeben:

Die Methode von TVirtualFitter

`void SetFCN(void (*fcn)(Int_t &, Double_t *, Double_t &, Double_t *, Int_t))`
erlaubt das Setzen der zu minimierenden Funktion (mit Namen „fcn“)

`void fcn(Int_t &npar, Double_t *gin, Double_t &f, Double_t *par, Int_t iflag)`

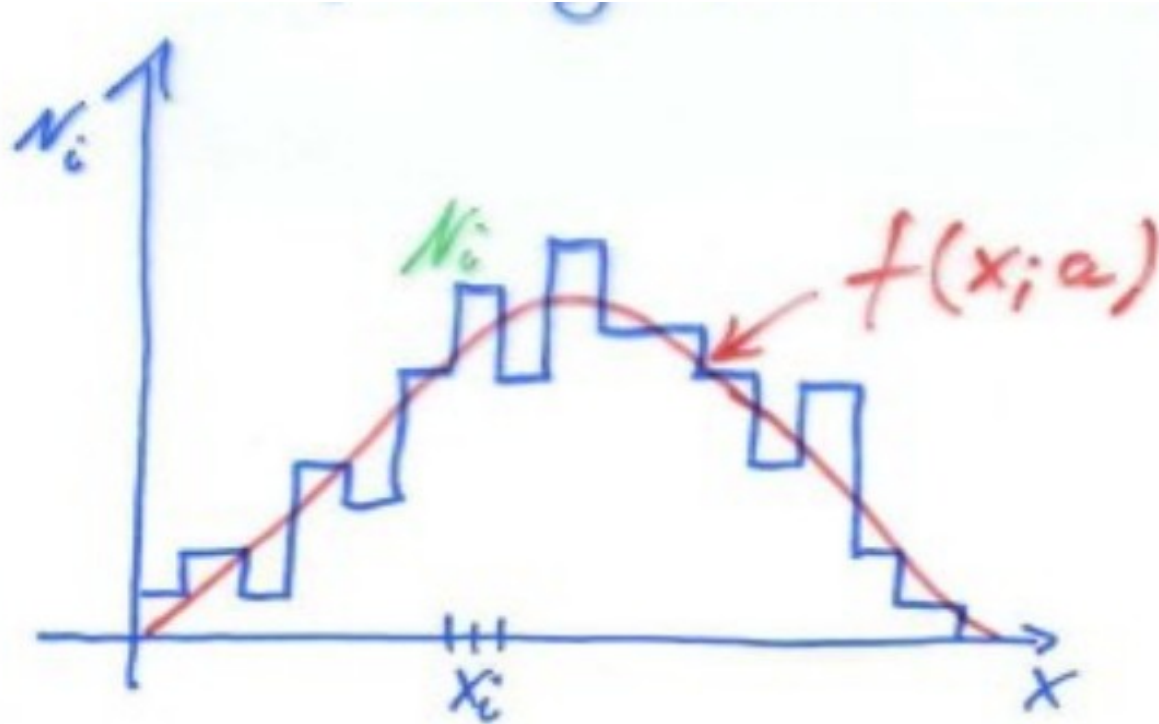
f ist dabei der Rückgabewert, im Array par übergibt Root die Parameter

Parameteranpassung mit ROOT

Ein Code-Fragment zur
Berechnung von χ^2
mit Kovarianz-Matrix

```
// fcn function for fit with covariance matrix
double arr_poi; // global array pointer
TMatrix *icovp; // global matrix pointer
int nval = 3; // global for number of measurements
int main()
{
    arr_poi = new double[3];
    TMatrix cov(3,3);
    cov(0,0) = ... , cov(2,2) = ...;
    // Invert matrix
    TMatrix icov = cov.Invert();
    icovp = &icov;
    ...;
}
void fcn(Int_t &npar, Double_t *gin, Double_t &f, Double_t *par, Int_t iflag)
{
    ...
    for (int i; i < nval; i++) {
        for (int j; j < nval; j++) {
            chi += (arr_poi[i] - fit_func(i)) *
                (arr_poi[j] - fit_func(j)) * (*icovp)(i,j) ;
        }
    }
    ...
}
```

Anpassen von Funktionen an Histogramme



Bininhalte N_i

$$\sigma_{N_i} = \sqrt{N_i} \quad (\text{Poisson})$$

$$N_i^t = \int_{x_i - b/2}^{x_i + b/2} f(x; a) dx$$

$$\simeq b \cdot f(x_i; a)$$

b : Binbreite

1. χ^2 -Methode:
$$\chi^2 = \sum_i \left(\frac{N_i - N_i^t}{\sqrt{N_i}} \right)^2$$

Probleme: *

- Bins mit $N_i=0$, d.h Fehler 0 Abhilfe: setze $\sigma_i = \sqrt{N_i^t}$
- N_i folgen Binomial- oder Poisson-Verteilung, χ^2 -Methode nimmt Gauß-Fehler an: Methode nur gut für „große“ N_i

Anpassen an Histogramme: „Binned Likelihood-Fit“

2. Binned Likelihood-Fit

$$\mathcal{L}_i = \text{Poisson}(N_i; N_i^t) = \frac{N_i^{t N_i}}{N_i!} \exp(-N_i^t)$$

$$-\ln \mathcal{L}_i = -N_i \ln N_i^t + N_i^t + \ln N_i$$

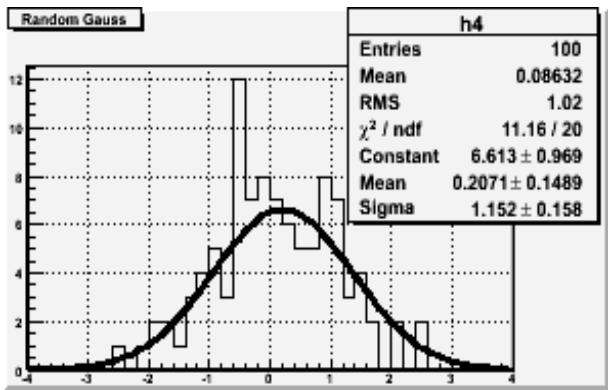
hängt nicht von
der Funktion ab

$$-\ln \mathcal{L} = \sum_i -\ln \mathcal{L}_i \quad \text{minimieren bzgl. } a$$

$\Rightarrow \hat{a} : \text{Schätzung für Parameter } a$

χ^2 – und Likelihood-Anpassung für Histogramme in ROOT
implementiert (χ^2 ist default!)

Achtung: ROOT verwendet $-2 \ln L$



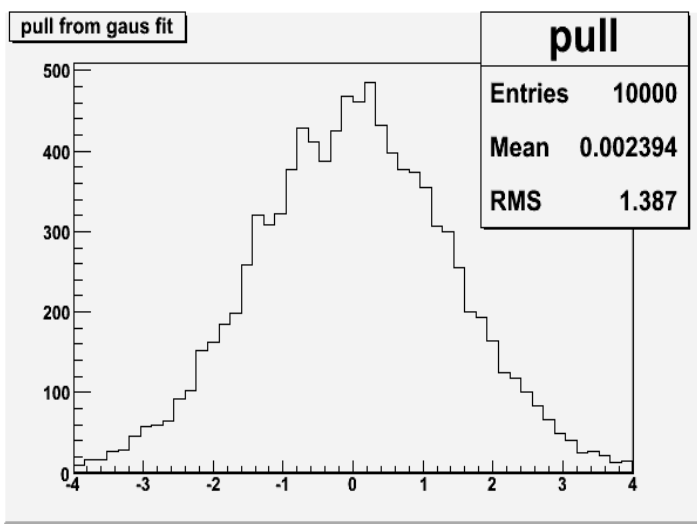
10'000 Histogramme dieser Art, jeweils
Gauß(x;μ,σ) anpassen mit der χ^2 -Methode

Test: Pull $p = \frac{(\text{fitted mean} - \mu)}{\text{error}}$

muss standard-normalverteilt sein -
ist es aber in diesem Fall nicht !

Grund: angenommene Fehler in jedem
Bin sind $\sqrt{n_i}$ - falls n_i nach unten fluktuiert,
wird auch der Fehler kleiner angenommen, die
Folge ist einer verzerrte Parameterschätzung
(betrifft hier vor allem den Parameter σ !)

Beispiel: pull.c



Abhilfe: -Log L – Anpassung:

Mean	0.0020
RMS	1.007

Behandlung von Fehlern in x- und y-Richtung

1. Ableitung nutzen, um x-Fehler in y-Fehler umzurechnen und Quadratisch zu den y-Fehlern addieren
Iteratives Verfahren:

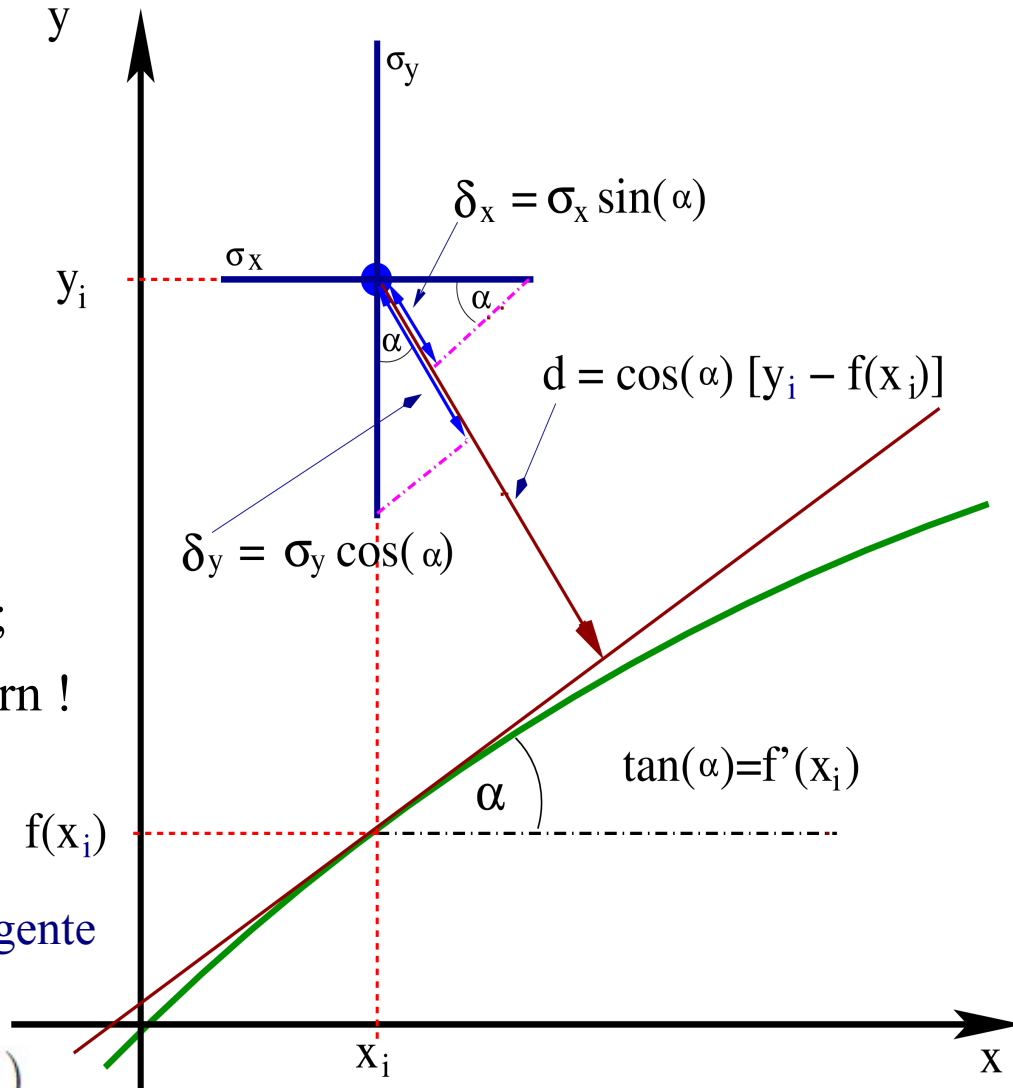
1. Anpassung ohne x-Fehler
2. $f'(x_i)$ bilden und neue Fehler bestimmen:
$$\sigma_i^2 = \sigma_{y_i}^2 + f'(x_i) \cdot \sigma_{x_i}^2$$
3. Schritt analog 2 zur Kontrolle wiederholen;
 χ^2 am Minimum darf sich nicht stark ändern !

Geometrische Interpretation:

Mimimierung des auf projizierte Messfehler normierten Abstands d der Punkte von der Tangente an die Funktion:

$$\chi^2 = d^2 / (\cos(\alpha) \sigma_y^2 + \sin(\alpha) \sigma_x^2)$$

Implementierung: Methode Fit() der Root-Klasse TGraphErrors



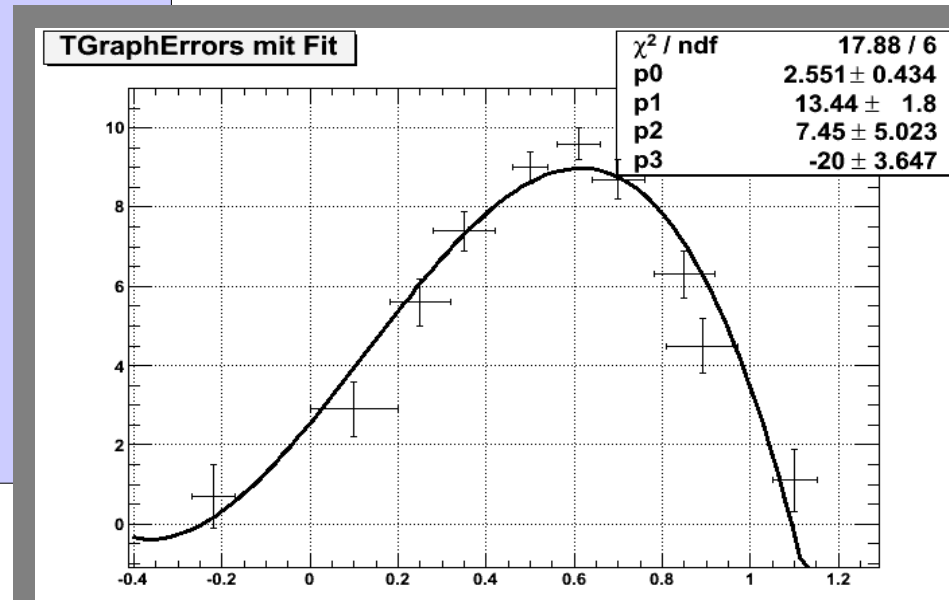
Parameteranpassung mit ROOT Klasse TGraphErrors

```
void TGraphFit() {  
    //Draw a graph with error bars and fit a function to it.  
    //set global options  
    gStyle->SetOptFit(111); //superimpose fit results  
    // make nice Canvas  
    TCanvas *c1 = new TCanvas("c1","Daten",200,10,700,500);  
    c1->SetGrid();  
    //define some data points  
    const Int_t n = 10;  
    Float_t x[n] = {-0.22, 0.1, 0.25, 0.35, 0.5, 0.61, 0.7, 0.85, 0.89, 1.1};  
    Float_t y[n] = {0.7, 2.9, 5.6, 7.4, 9., 9.6, 8.7, 6.3, 4.5, 1.1};  
    Float_t ey[n] = {.8,.7,.6,.5,.4,.4,.5,.6,.7,.8};  
    Float_t ex[n] = {.05,.1,.07,.07,.04,.05,.06,.07,.08,.05};  
    // copy data to TGraphErrors object  
    TGraphErrors *gr = new TGraphErrors(n,x,y,ex,ey);  
    gr->SetTitle("TGraphErrors mit Fit");  
    gr->Draw("AP");  
    // now perform a fit(with errors in x and y!)  
    gr->Fit("pol3");  
    c1->Update();  
}
```

Script TGraphFit.C

Anpassung mittels
Root Macro:

> root TGraphFit.C



Behandlung von Fehlern in x- und y-Richtung (2)

Allgemein bei korrelierten Fehlern: Kovarianzmatrizen C^x und C^y
im zweiten Schritt x-Fehler zur Kovarianzmatrix addieren:

$$C_{i,j} = C_{i,j}^y + C_{i,j}^x \cdot f'(x_i) \cdot f'(x_j)$$

Geschrieben mit dem Residuenvektor $\Delta_i(p_1, p_2, \dots) = y_i - f(x_i; p_1, p_2, \dots)$
und dem Vektor der 1. Ableitungen, $\vec{f}'$
ergibt sich

$$\chi^2(p_1, p_2, \dots) = \vec{\Delta}(p_1, p_2, \dots)^T \left(\mathbf{C}^y + \vec{f}'^T \mathbf{C}^x \vec{f}' \right)^{-1} \vec{\Delta}(p_1, p_2, \dots)$$

Implementiert im Programm RooFiLab, basierend auf Root

abschließende Bemerkungen

Hier konnte nicht alles angesprochen werden;
es gibt noch viele weitere Fragestellungen ...

- **Anpassen mit Nebenbedingungen:**
 - Parameter innerhalb von Grenzen, $a < \lambda_i < b$
 - Einschränkungen an Parameter, z.B. durch Funktionen $e_j(\lambda_1, \dots, \lambda_p) = 0$
- Wie funktioniert **numerische Optimierung** / Minimierung ?
- Weitere **praktische Beispiele** ?
- Wie man es nicht machen sollte: **beliebte Fehler**.

... (einige) Antworten in Teil 2 im nächsten Semester:
Moderne Methoden der Datenanalyse

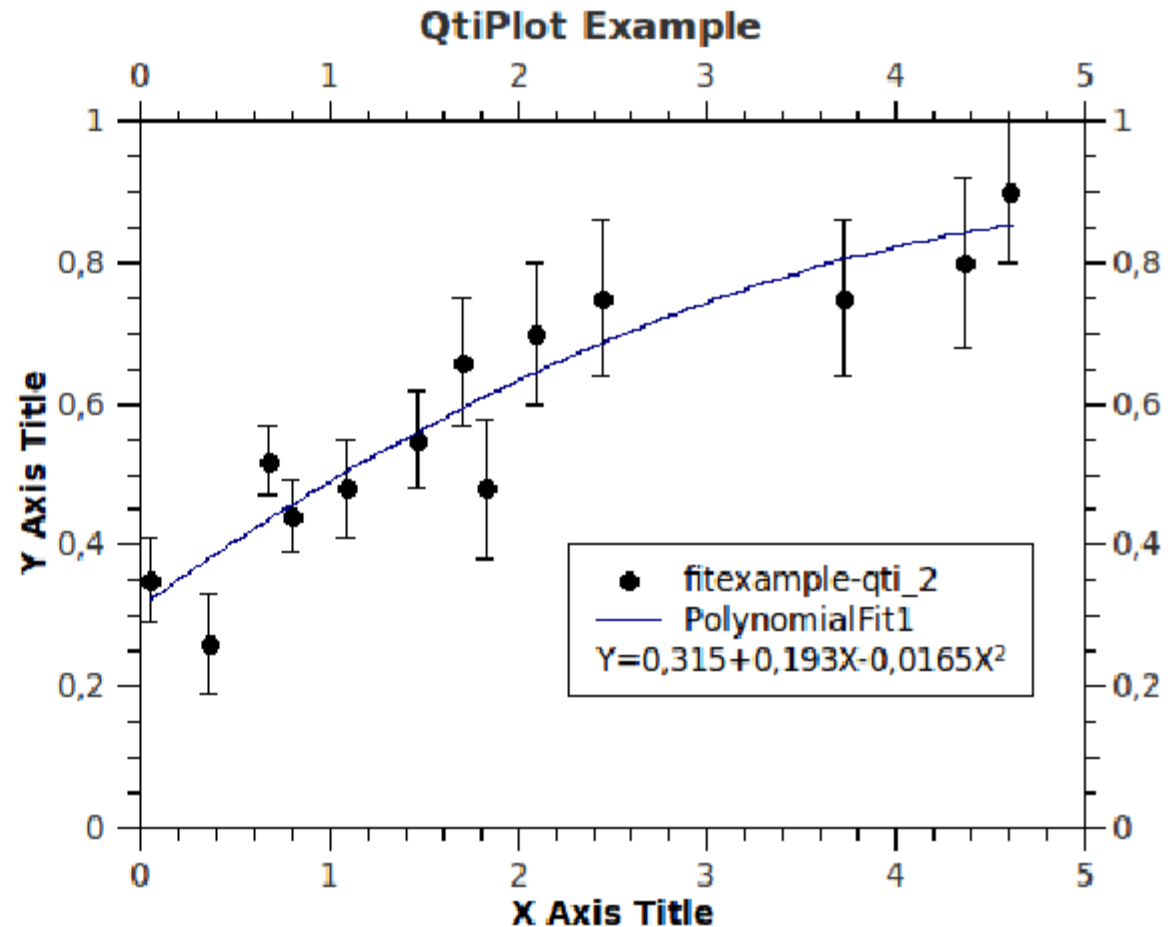
Fertige Programmpakete

(freie) Programme zur Parameteranpassung

qtiplot

<http://wiki.ubuntuusers.de/qtiplot>

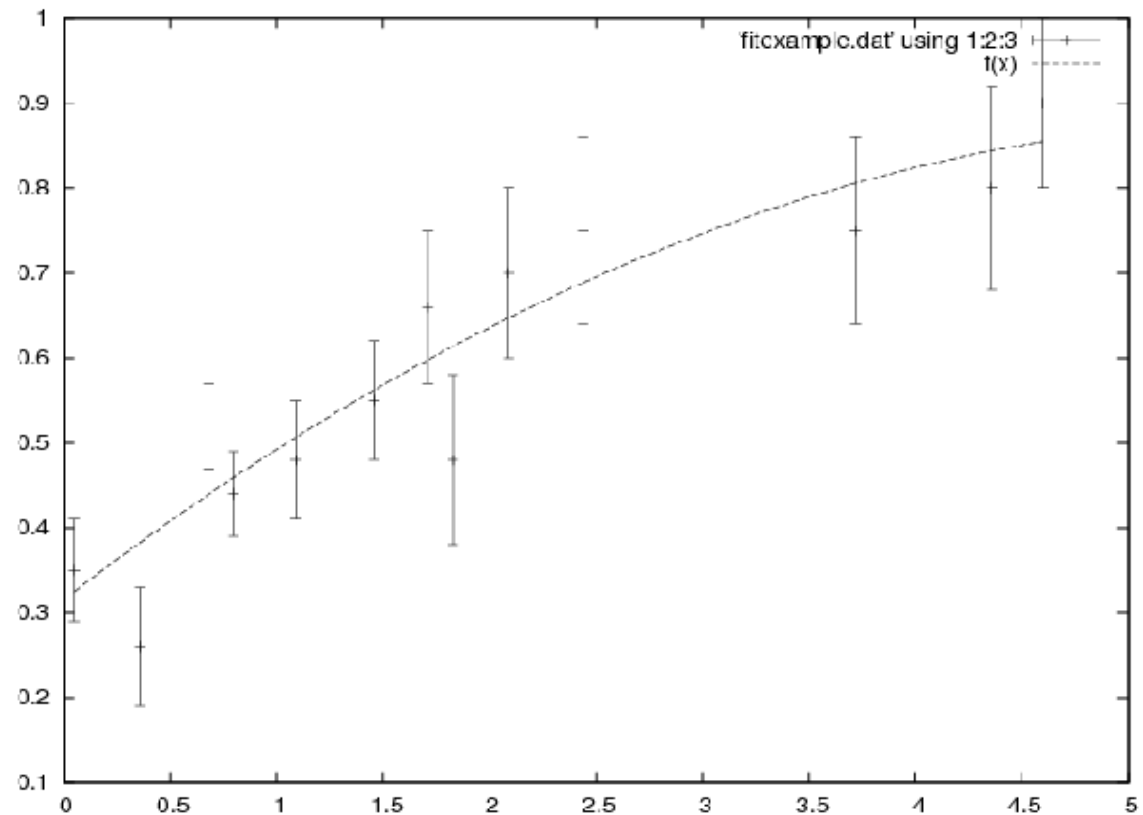
#	x	y	dy
0	0,05	0,35	0,06
1	0,36	0,26	0,07
2	0,68	0,52	0,05
3	0,80	0,44	0,05
4	1,09	0,48	0,07
5	1,46	0,55	0,07
6	1,71	0,66	0,09
7	1,83	0,48	0,10
8	2,44	0,75	0,11
9	2,09	0,70	0,10
10	3,72	0,75	0,11
11	4,36	0,80	0,12
12	4,60	0,90	0,10



(freie) Programme zur Parameteranpassung

gnuplot

```
# ----- data -----  
# x      y      ey  
.05      0.35    0.06  
0.36     0.26    0.07  
0.68     0.52    0.05  
0.80     0.44    0.05  
1.09     0.48    0.07  
1.46     0.55    0.07  
1.71     0.66    0.09  
1.83     0.48    0.1  
2.44     0.75    0.11  
2.09     0.70    0.1  
3.72     0.75    0.11  
4.36     0.80    0.12  
4.60     0.90    0.1
```



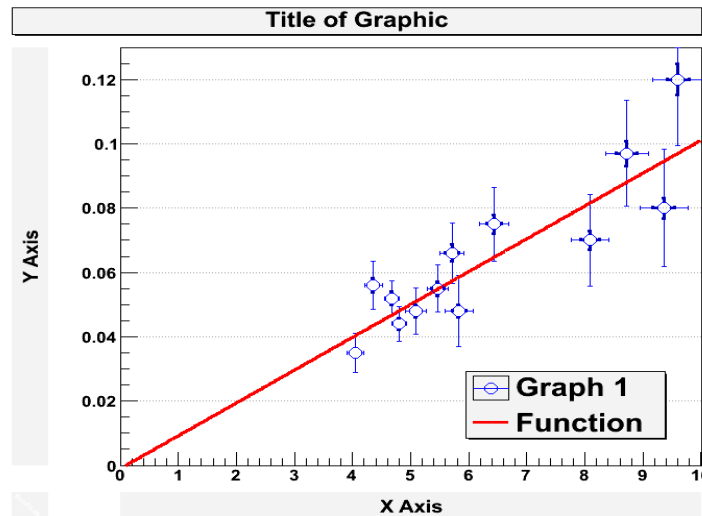
(freie) Programme zur Parameteranpassung

RooFiLab

<http://www-ekp.physik.uni-karlsruhe.de/~quast/RooFiLab>

Eigenentwicklung (HiWi Thomas Müller)

- Fehler in x- und y-Richtung
- korrelierte Fehler
- geführte, grafische Oberfläche, fit-by-eye und automatisierte Anpassung mit Skript (angelehnt an gnuplot)
- weitgehende Editierbarkeit der grafischen Ausgabe mittels ROOT-GUI



The screenshot shows the "SPECIFY INITIAL VALUES" window of the RooFiLab GUI. The window has a title bar with "IMPORT DATA" and "SPECIFY INITIAL VALUES". The "Selection" section contains a "Graph:" dropdown menu set to "graph2 (Graph 1)" and a "Function:" text box containing $f(x) = m * x + b$. The "Specify initial Values" section has input fields for "m" (0.0102124) and "b" (-0.00100647). The "Information on the Fit-Function" section shows "Chi^2:" (10.7874) and "Degrees of freedom:" (11). At the bottom, there are three buttons: "RUN ROOT-FIT", "FINISH ANALYSIS", and "HELP".