Einleitung

Es gibt 2 Ansätze zur statistischen Analyse und Zuordnung von Modellen

- Test einer Hypothese: Sind unsere Daten mit einem theoretischen Modell kompatibel
- Bestimmung von Modell Parametern: Verwendung der Daten, um die freien Parameter eines Modells anzupassen (Fitting).

Der Hypothesentest ist meist eine Vorstufe der Datenanpassung. Bei der Datenanpassung unterscheiden wir 3 Methoden

- Analyse der Momente, z.B. Mittelwerte, Varianz, Median, FWHM, most probable value, 2. und 3. Moment eines Datensatzes. Daraus lassen sich durch einfache Berechnungen Modellparameter und die zugehörigen Fehler ermitteln. → Analyse mit Histogrammen
- Datenanpassung mit der Methode der kleinsten Quadrate (Least Square Fit)
  Iterativer Prozess bei dem die Abweichung eines Modells mit freien Parametern von den Daten minimiert wird, um den besten Wert für die freien Parameter zu bestimmen.
- Likelihood Methode (Likelihood Fit)
  Finde für einen Satz von freien Parametern den wahrscheinlichsten Wert zur Beschreibung der Daten. Die Parameter werden durch Maximieren der Wahrscheinlichkeitsverteilung bestimmt.

Der Hypothesentest und die Analyse der Momente werden meist verwendet um erste Schätzungen für Parameter zu erhalten.

Modellparameter ohne Fehlerabschätzung sind sinnlos.

Hypothesen Test (1)

Treffe aus der Beobachtungen von Zufallsvariablen eine Entscheidung über die Gültigkeit einer Hypothese.
Es wird versucht die Wahrscheinlichkeiten für Fehlentscheidungen durch einen Test zu einem Signifikanzniveau zu kontrollieren, d.h. wie wahrscheinlich ist es, dass eine exakt zutreffende Nullhypothese irrtümlich verworfen werden könnte.

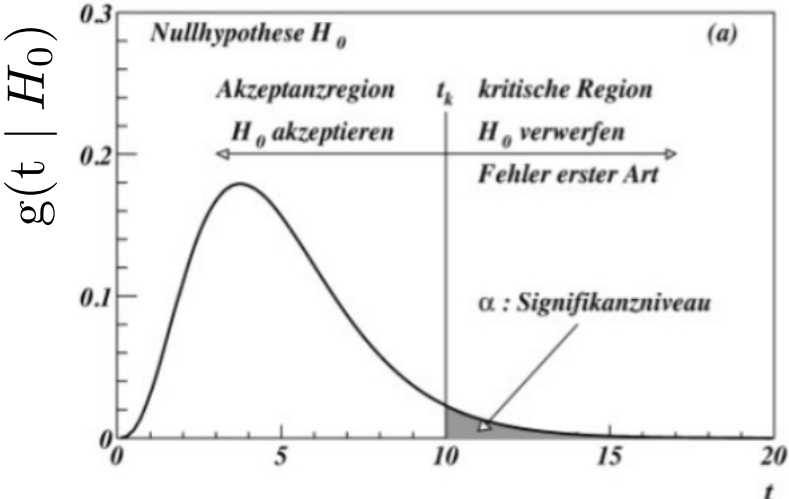
Skizze zur Untersuchung von Hypothesen aufgrund von gemessenen Daten

- Definiere eine Nullhypothese H_0 und Alternativhypothese H_1
- Wähle eine Testgröße oder Teststatistik $t(x_1,x_2,..,x_n)$ mit $x_1,..,x_n$ Stichprobenwerten
- Bestimme Wahrscheinlichkeitsdichte für t unter den Hypothesen $H_i \rightarrow f(t|H_i)$
- Festlegen eines Kriteriums für das Verwerfen/Unterscheiden der Hypothesen
  Es sollen für H_0 und H_1 Wahrscheinlichkeitsdichten $g(t|H_0)$ und $g(t|H_1)$ existieren und berechnet werden können
  $t(x_1,x_2,..,x_n) = t_{cut}$ legt die Entscheidungsgrenzen (kritische Region) fest
  für $t>t_{cut}$ wird H_0 verworfen bzw. H_1 akzeptiert

Signifikanzniveau: $\alpha = \int_K g(\vec{t}|H_0)d\vec{t}$

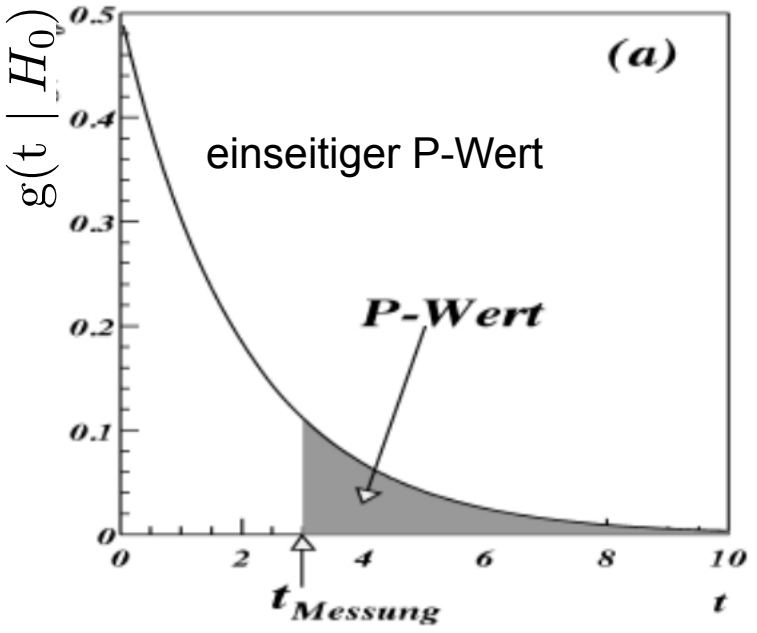
kritische Region: K

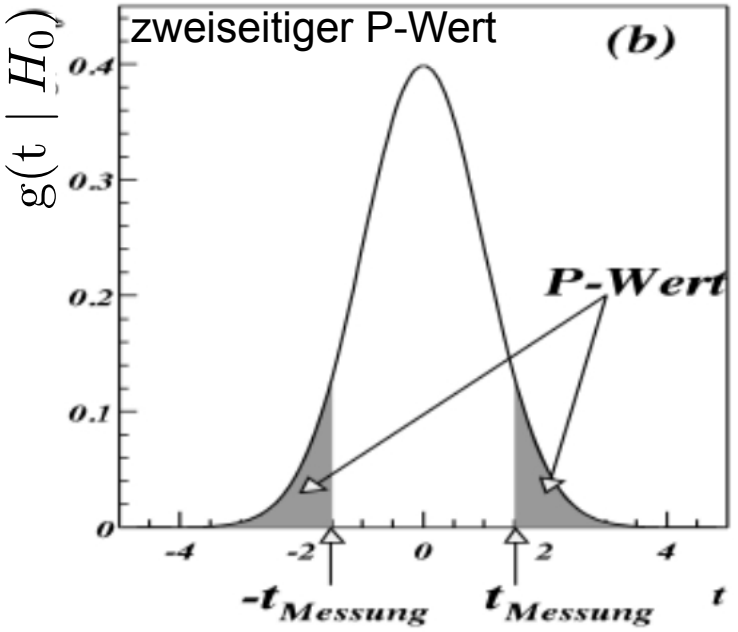
α ist die Wahrscheinlichkeit H_0 zu verwerfen
typische Werte 1% - 5%



Hypothesen Test (2)

P-Wert: Wahrscheinlichkeit die Stichprobe einer Messung zu beobachten, die genauso oder weniger verträglich mit der Nullhypothese H_0 ist wie die aktuelle Messung.





- Bei Übereinstimmung von H_0 mit den Daten ist t=0
- Wenn der P-Wert gleich dem Signifikanzniveau α ist, dann gilt $t_{Messung} = t_{cut}$
- Der P-Wert wird auch beobachtetes Signifikanzniveau genannt
- Eine Einheit im P-Wert wird Vertrauensniveau (confidence level) genannt
- Wenn P-Wert < Signifikanzniveau dann wird H_0 verworfen
- Falsche Aussagen: i) P-Wert ist nicht die Wahrscheinlichkeit das H_0 falsch ist
  ii) Eine Einheit im P-Wert ist nicht die Wahrscheinlichkeit das H_0 wahr ist

Chi-Quadrat-Test

Der Chi-Quadrat-Test ist ein Hypothesentest mit χ^2 verteilter Teststatistik.

Sei $\vec{x} = (x_0, x_1, ...x_i)$ eine Liste von n unabhängigen Messungen mit den Varianzen σ_i^2 und den Modellvorhersagen $\vec{\nu} = (\nu_0, \nu_1, ...\nu_i)$

χ^2 Teststatistik: $\chi^2 = \sum_{i=1}^{n} \frac{(x_i - \nu_i)^2}{\sigma_i^2} = \sum_{i=1}^{n} \frac{(x_i - \nu_i)^2}{\nu_i}$

wenn x_i poissonverteilt ist $\sigma_i^2 = V[x_i] = \nu_i$

Wenn die Hypothese, das die Stichprobe aus der Modellvorhersage stammt, wahr ist, und die Messungen gaussverteilt mit Mittelwert ν_i und Standardabweichung σ_i^2 sind, also $x_i \sim N(\nu_i, \sigma_i^2)$, dann folgt die χ^2 Teststatistik einer χ^2 Wahrscheinlichkeits-dichte-Verteilung mit n Freiheitsgraden mit dem

P-Wert: $P = \int_{\chi^2}^{\infty} f_{\chi^2}(z; n) dz$

Die χ^2 Teststatistik ist ein Mass für die Übereinstimmung zwischen den Messungen und der Modellvorhersage.

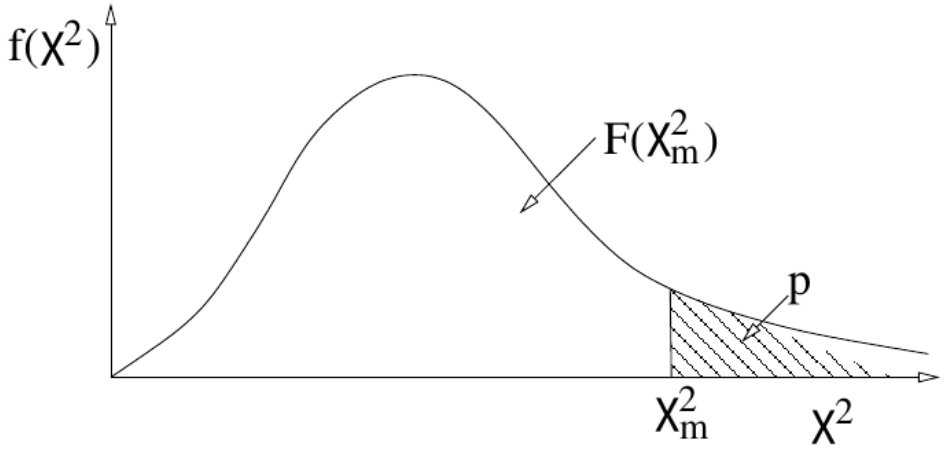
Chi-Quadrat-Test

Die Zuverlässigkeit einer Messung beziehungsweise den Grad der Übereinstimmung mit dem Gauss-Modell erfolgt durch Angabe des Integrals über die χ^2- Verteilung oberhalb des gemessenen χ^2-Wertes χ^2_m. Der P-Wert gibt also die Wahrscheinlichkeit an, dass bei den gemachten Annahmen eine Messung einen schlechteren χ^2-Wert ergibt, also $\chi^2 > \chi^2_m$.

Ein gemessener χ^2_m -Wert wird mit einem χ^2 -Wert für ein vorgegebenes Vertrauensniveau α verglichen:

$$p = 1 - F(\chi^2_m) \qquad \alpha = 1 - F(\chi^2_\alpha)$$

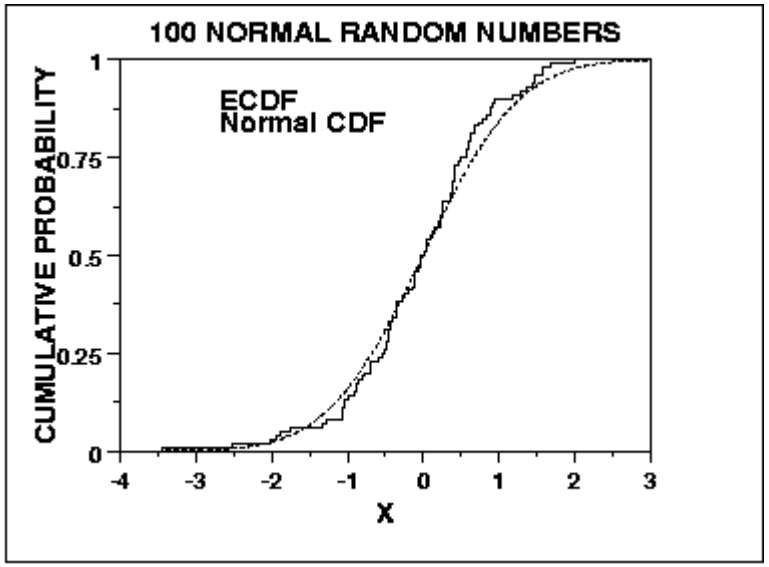
P-Wert für einen gemessenen χ^2-Wert χ^2_m :



In ROOT ist der Chi-Quadrat-Test als TH1::Chi2Test implementiert.
Es wird geprüft, ob zwei Histogramme der selben Verteilung folgen und es wird ein P-Wert ausgegeben.

Kolgomorov Smirnov Test

Dient zur Prüfung, ob ein sample einer gegebenen Verteilung gehorcht.
Sei $x_0, x_1, x_2 ... x_i$ eine geordnete Liste von N Datenpunkten, dann ist $E_N = n(i)/N$ eine empirische Verteilung mit $n(i)$ gleich der Anzahl der Datenpunkte kleiner x_i.
Die kummulative Verteilung ist eine Treppenfunktion. Der Test bestimmt die Abweichung von 2 Verteilungen und ist gut geeignet, um die Form von Verteilungen zu vergleichen.
Details and plot: https://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm



In ROOT gibt es die Methode TH1::KolmogorovTest() zum statistischen Test zweier Histogramme oder für unbinned data TMath::KolmogorovTest

$$d_n = sup|F_n(x) - F_0(x)|$$

Wenn die absolut größte Differenz einen für eine Signifikanz gegebenen Wert überschreitet, wird die Hypothese abgelehnt.

Datenanpassung in ROOT

- Generelle Fragestellung

  Welche Möglichkeiten haben wir einen Satz von n mal gemessenen Werten (Stichprobe) mit Hilfe eines Modells zu beschreiben?

  Zum Beispiel wollen wir die invariante Masse des Zerfalls $D^0 \rightarrow K^+\pi^-$ bestimmen. Wir messen die Vierervektoren (E, P_x, P_y, P_z) von K^+und π^- und bilden

$$m^2 = [(E, P_x, P_y, P_z)_K + (E, P_x, P_y, P_z)_\pi] \cdot [(E, P_x, P_y, P_z)_K + (E, P_x, P_y, P_z)_\pi]$$

  Durch die statistische Natur des Zerfalls und der Messung von Energie und Impuls ist die invariante Masse gaussverteilt mit Parametern μ und σ^2 .

  Das Ziel ist eine Messung von μ und σ^2 und eine Bestimmung des Messfehlers.

  Die Methodik der Parameterbestimmung wird im Physiker Slang als Fit bezeichnet.

- Die Datenanpassung in ROOT erfolgt mit Hilfe von speziellen Programm-Paketen
  - Minuit/Minuit2 wird in ROOT zum Fitten verwendet
  - RooFit → Behandlung von PDFs und Fitting framework (B-Physics @ BaBar)
  - RooStats → Implementiert einheitliche statistische Methoden für LHC Physik

Modelierung einer Messung – erste Schritte

In dem File D0Mass.txt finden Sie wiederholte Messungen der invarianten Masse des D^0 Zerfalls $D^0 \to K^+\pi^-$

Arbeitsvorschlag:

- Schreiben Sie ein ROOT Macro, das die gemessene invariante Masse darstellt.
  Bestimmen Sie die Anzahl der wiederholten Messungen.
  Gibt es ein Model mit dem wir die Messungen beschreiben können.
  Wie lassen sich die Modellparameter bestimmen?

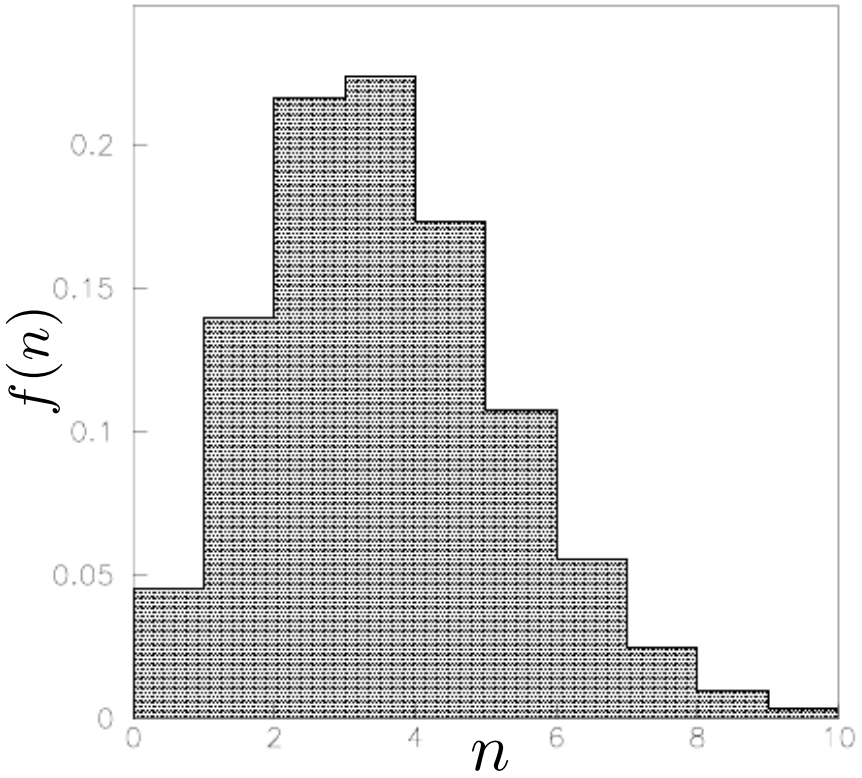
readD0Signal.cc

Maximum Likelihood - Methode

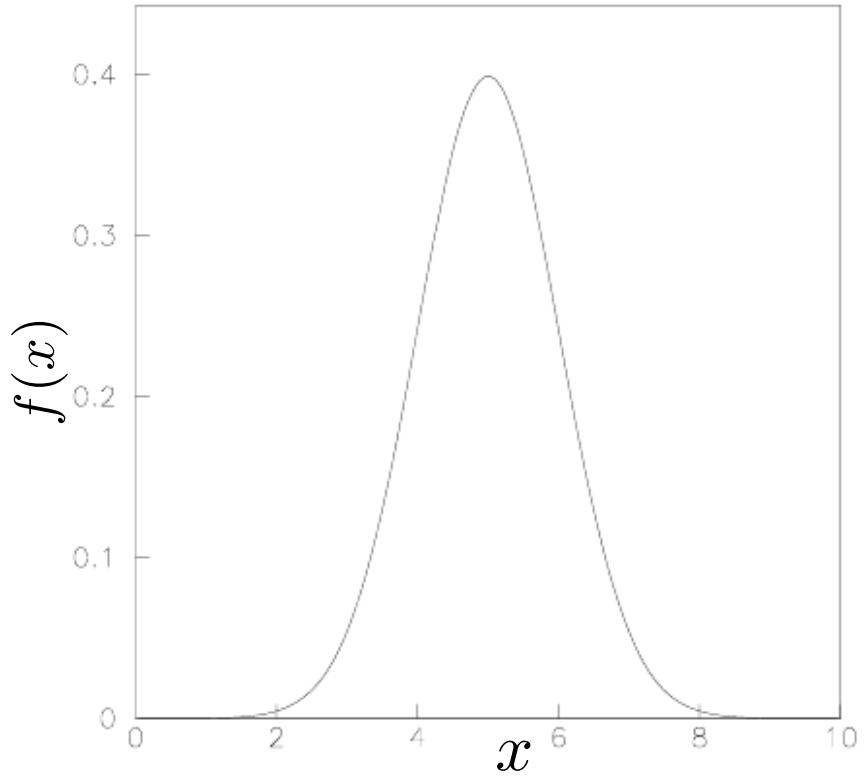
- Wahrscheinlichkeitsdichte (Probability Density Function (PDF))

Experimente werden häufig durch eine Zufallsvariable x oder mehrere $\vec{x} = (x_0, x_1, ...x_n)$ beschrieben. Die Variablen nehmen diskrete oder kontinuierliche Werte an.
Die Wahrscheinlichkeit $\Delta p(x)$ für das Auftreten von x in einem Intervall Δx wird mit der Wahrscheinlichkeitsdichte $f(x)$ beschrieben: $\Delta p(x) = f(x)\Delta x$

Diskrete Zufallsfallsvariable



Kontinuierliche Zufallsfallsvariable



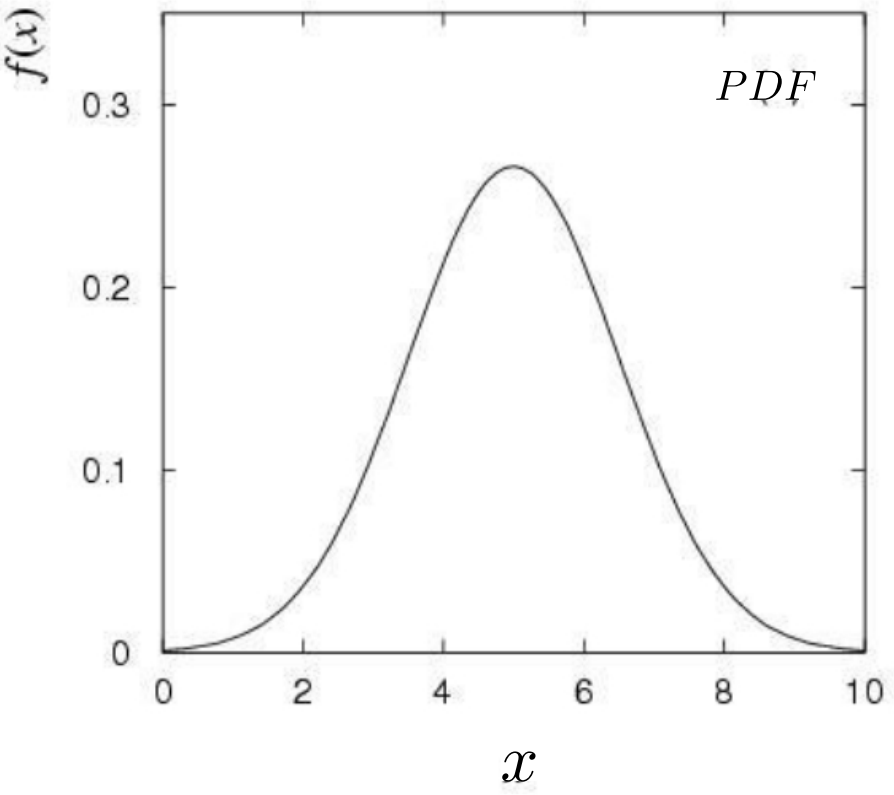
Nichtnegativ: $f(x) \geq 0 \ \forall \ x \in \mathbb{R}$ Normierung: $\int_{-\infty}^{\infty} f(x)dx = 1$

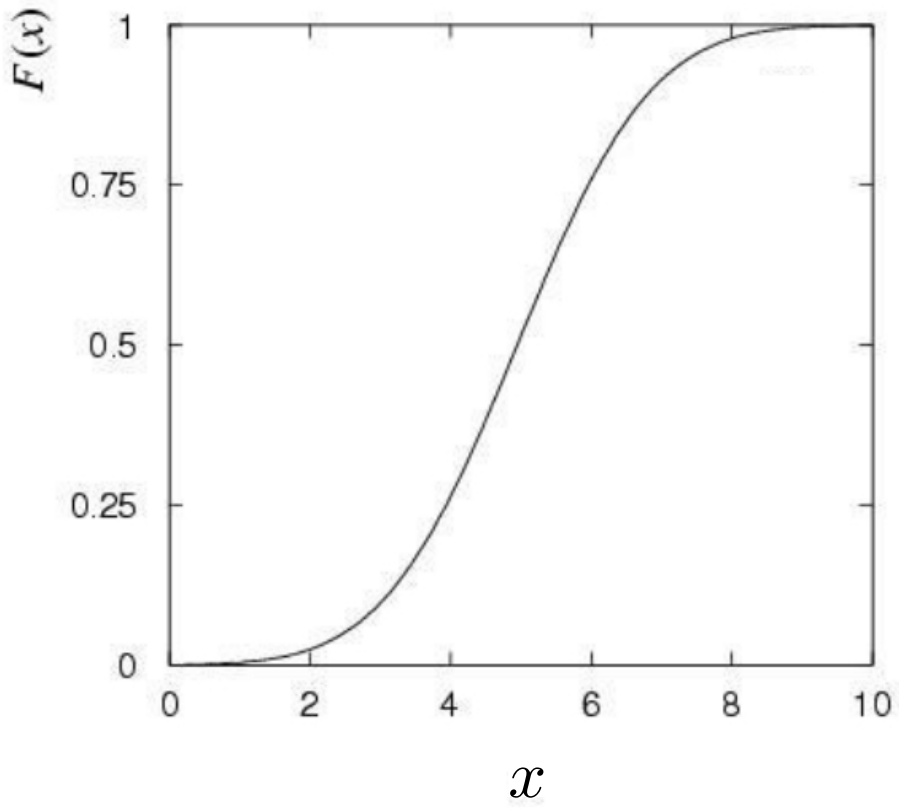
Maximum Likelihood - Methode

- Kummulative Wahrscheinlichkeitsverteilung

Sei $f(x)$ eine Wahrscheinlichkeitsdichteverteilung, dann wird mit

$$\int_{-\infty}^{x} f(x')dx' \equiv F(x)$$ die kummulative Wahrscheinlichkeitsverteilung beschrieben.





- (Vollständiger) Katalog von PDFs und deren Eigenschaften

http://staff.fysik.su.se/~walck/suf9601.pdf

Maximum Likelihood - Methode

Allgemeine Methode zur Bestimmung von optimalen Parametern aus Stichproben für beliebige Wahrscheinlichkeitsverteilungen.

- Maximum-Likelihood-Prinzip

Betrachten wir eine Stichprobe $x_1, x_2, ..., x_n$, die aus n Messungen besteht. Jedes x_i kann dabei für einen ganzen Satz von Variablen stehen.
Die Einzelmessungen x_i gehorchen dabei Verteilungen mit der Wahrscheinlichkeitsdichte $f(x_i)$, die von einem Parametersatz $\theta = \theta_1, \theta_2, ..., \theta_k$ abhängen, geschrieben $f(x|\theta)$

Die Gesamtwahrscheinlichkeit für das Auftreten einer Stichprobe ist das Produkt der Einzelwahrscheinlichkeiten der einzelnen Elemente der Stichprobe.

$$L(x_1, ..., x_n) = \prod_{i=1}^{n} f(x_i|\theta) \quad \text{mit} \quad \int_\Omega L(x_1, ..., x_n) dx_1 ... dx_n = 1$$

wobei Ω der Definitionsbereich der Stichprobe $x_1, x_2, ..., x_n$ ist.
L heißt Likelihood-Funktion.

Der optimale Parametersatz $\hat{\theta}$ ist der, der die Likelihood-Funktion L maximiert.

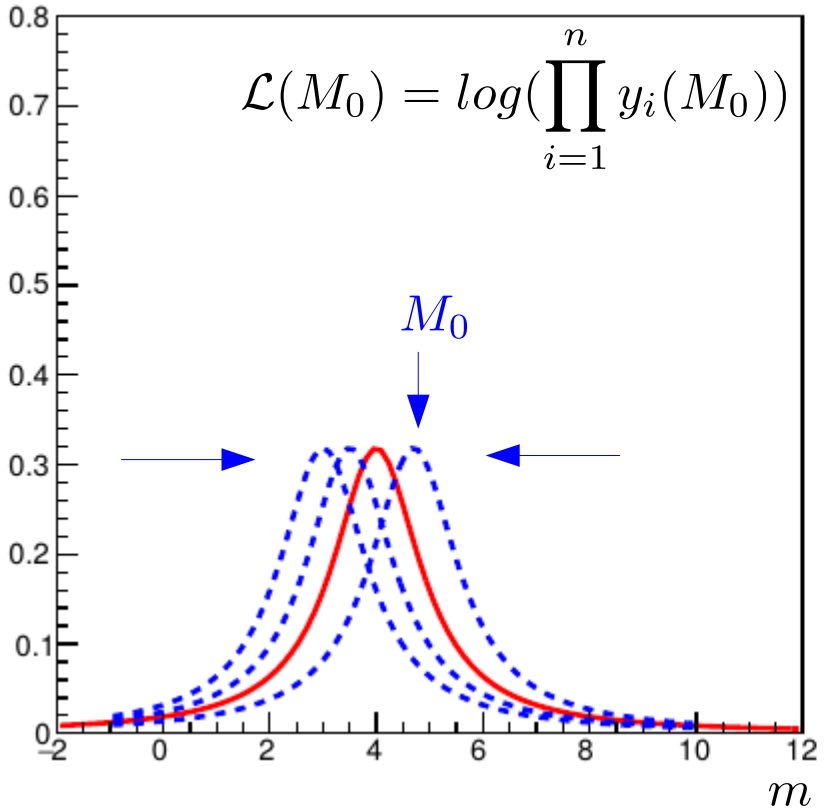
Aus numerischen Gründen wird der Logarithmus der Likelihood-Funktion betrachtet

$$\mathcal{L}(x_1, ..., x_n|\theta) = log(L(x_1, ..., x_n|\theta)) = \sum_{i=1}^{n} log(f(x_i|\theta)) \quad \text{und} \quad -\mathcal{L} \text{ minimiert.}$$

Maximum Likelihood - Methode

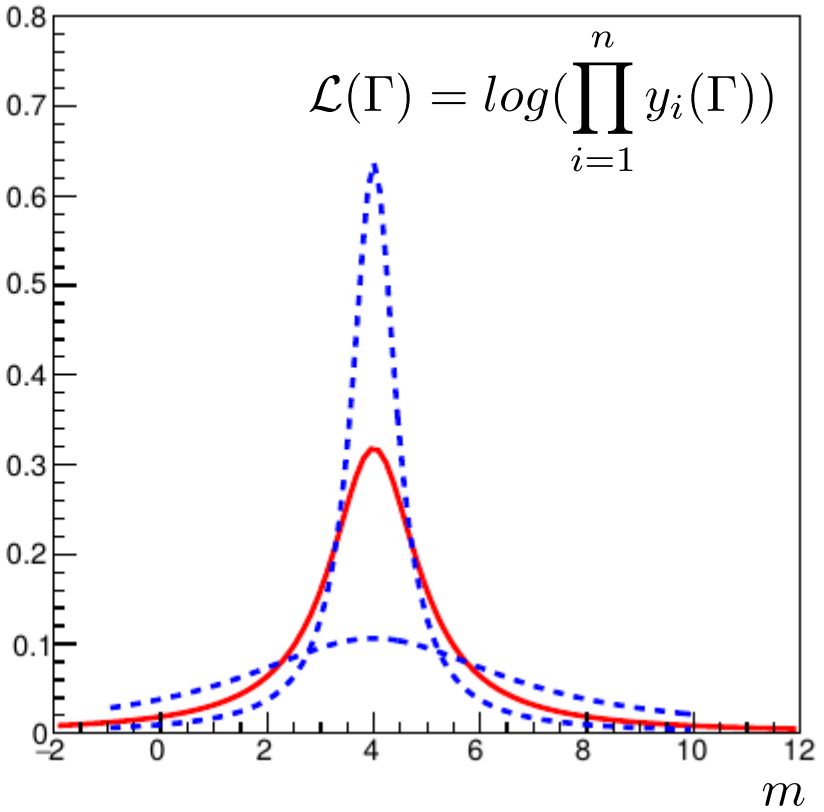
- Anschauliches Beispiel: Anpassung einer Breit-Wigner Resonanz

$$y_i(M_0, \Gamma) = \frac{1}{2\pi} \frac{\Gamma}{(m_i - M_0)^2 + (\Gamma/2)^2}$$



Finde die Position der Kurve so das das Produkt der y_i maximal ist.

Der beste Wert für M_0 ist dort wo die meisten Werte von m_i zu finden sind



Finde die Breite der Kurve so das das Produkt der y_i maximal ist.

Der optimale Wert für Γ hängt von der Breite der m_i Verteilung ab.

Maximum Likelihood - Methode

- Beispiel zur Parametersatz Bestimmung mit dem Maximum-Likelihood-Prinzip

Betrachten wir eine Stichprobe $x_1, x_2, ..., x_n$, die aus n normalverteilten Messungen besteht. Wie groß sind der Mittelwert μ und die Varianz σ^2 der Messgröße x ?

Wahrscheinlichkeitsdichte: $$f(x, |\theta_1, \theta_2) = \frac{1}{\sqrt{\theta_2}\sqrt{2\pi}} e^{-\frac{(x_i-\theta_1)^2}{2\theta_2}}$$

Likelihood-Funktion: $$L(x_1, ..., x_n) = \prod_{i=1}^{n} f(x_i|\theta_1, \theta_2) = \theta_2^{-n/2}(2\pi)^{-n/2} e^{-\frac{1}{2\theta_2}\sum_{i=1}^{n}(x_i-\theta_1)^2}$$

Log-Likelihood: $$\mathcal{L}(x_1, ..., x_n|\theta_1, \theta_2) = -\frac{n}{2} log\theta_2 - \frac{n}{2} log(2\pi) - \sum_{i=1}^{n} \frac{(x_i - \theta_1)^2}{\theta_2}$$

Maximieren: $$\frac{\partial \mathcal{L}(x_1, ..., x_n|\theta_1, \theta_2)}{\partial \theta_1} = 0 \qquad \sum_{i=1}^{n} x_i - n\theta_1 = 0$$

$$\frac{\partial \mathcal{L}(x_1, ..., x_n|\theta_1, \theta_2)}{\partial \theta_2} = 0 \qquad n\theta_2 + \sum_{i=1}^{n} (x_i - \theta_1)^2 = 0$$

Optimale Parametersatz $\hat{\theta}$: $$\hat{\theta_1} = \hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \hat{\theta_2} = \hat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n} (x_i - \theta_1)^2$$

Maximum Likelihood - Methode

- Eigenschaften der Maximum-Likelihood-Methode

Was bedeutet optimaler Parametersatz?

- Likelihood Schätzung der Parameter $\hat{\theta} = \theta_1, \theta_2, ..., \theta_k$ ist invariant gegen Parameter-transformation.

$$\theta \to \phi \quad \text{mit} \quad \hat{\phi} = \phi(\hat{\theta})$$

- Konsistenz: für große n geht der Schätzwert in den wahren Wert über

$$\lim_{n\to\infty} \hat{\theta} = \theta_{true}$$

- Effizient: die Schätzwerte haben für große n minimale Varianz

- Kommentare zur Maximum-Likelihood-Methode

- Man braucht kein Histogramm und da die Wahrscheinlichkeit eventweise bestimmt wird, benötigt man auch keine hohe Ereignisdichte
- Durch Variablentransformation kann ein Datensatz für verschiedene Variable benutzt werden.
- Funktionen mit impliziten Definitionen lassen sich auch leicht verwenden.
- Parameter mit erlaubten/verbotenen Bereichen lassen sich einfach verwenden
- Untergrund ist problematisch/schwierig zu behandeln → kombinierte Likelihood
- Rechenintensiv
- Ist unsere Funktion die richtige Beschreibung? Was bedeutet hier Maximal? → Simulation

Likelihood Ratio

- Bestimmung der Fit Qualität mit dem Likelihood Verhältnis

Betrachten wir eine Stichprobe , die aus n Messungen besteht und die die Likelihood $\mathcal{L}(x_1, ..., x_n|\theta)$ mit N Parametern $\theta = \theta_1, \theta_2, ..., \theta_N$ hat. Wir definieren die Statistik

$$t_\theta = -2 \cdot ln \frac{L(x|\theta)}{L(x|\hat{\theta})}$$

mit dem maximum likelihood Parametersatz $\hat{\theta}$. Der Wert von t_θ ist ein Maß für die Übereinstimmung der Beschreibung der Daten mit dem Parametersatz θ .

- Gute Übereinstimmung bedeutet $\theta \approx \hat{\theta}$ und t_θ ist klein
- Grosse Werte für t_θ bedeuten geringe Übereinstimmung zwischen Daten und der durch θ ausgedrückten Hypothese

Die Wahrscheinlichkeit für Übereinstimmung mit $\hat{\theta}$ wird durch den p-Wert quantifiziert

Beobachteter Wert für eine Hypothese

$$p_\theta = \int_{t_{\theta,obs}}^{\infty} f(t_\theta|\theta) dt_\theta$$

PDF von t_θ , muss bekannt sein

Nehmen wir an, der Parametersatz $\theta = \theta_1, \theta_2, ..., \theta_N$ kann durch einen anderen Parametersatz $\mu = \mu_1, \mu_2, ..., \mu_M$ bestimmt werden, wobei $M < N$
Im maximum likelihood fit wird $\theta_i = \theta(x_i|\mu)$ benutzt und wir definieren die Statistik q_θ

Likelihood Ratio

- Bestimmung der Fit Qualität mit dem Likelihood Verhältnis

mit $$q_\theta = -2 \cdot ln \frac{L(x|\theta(\hat{\mu}))}{L(x|\hat{\theta})}$$ ← Fit M Parameter ← Fit N Parameter

Dies testet die Übereinstimmung der funktionalen Form der Hypothese $\theta(x|\mu)$. Um die Wahrscheinlichkeit für Übereinstimmung mit $\hat{\theta}$ zu erhalten, also q_θ zu bestimmen, muss die PDF $f(q_\theta|\theta)$ bekannt sein.

q_θ ist nicht Wahrscheinlichkeit dafür, dass das Modell richtig ist.

- Wilks' Theorem

Für den wahren Parametersatz $\theta = \theta_1, \theta_2, ..., \theta_N$ im Limit grosser Datensätze gehorchen t_θ und q_θ einer χ^2 Verteilung.

Für $$t_\theta = -2 \cdot ln \frac{L(x|\theta)}{L(x|\hat{\theta})}$$ ist $$f(t_\theta|\theta) \approx \chi^2_N$$ verteilt

Und für $$q_\theta = -2 \cdot ln \frac{L(x|\theta(\hat{\mu}))}{L(x|\hat{\theta})}$$ ist $$f(q_\theta|\theta) \approx \chi^2_{N-M}$$ verteilt

Dies gilt unabhängig vom Wert von μ. Wir können also p-Werte berechnen ohne Werte für μ zu kennen.

http://www.hep.fsu.edu/~harry/statistics/Cowan_2013.pdf

Likelihood Ratio

- Anwendung des Wilks' Theorems auf die Bestimmung der Signifikanz einer Messung.

Die Signifikanz einer Messung kann man mit dem Wilks' Theorem ableiten

Wir haben eine Verteilung mit einer Likelihood als Null Hypothese, H_0 , typischer Weise eine Anpassung an Untergrund Daten. Eine weitere Verteilung mit einer Likelihood als Hypothese H_1, die Signal mit Untergrund beschreibt.

Wir bestimmen $t_H = -2 \cdot ln \frac{L(x|H_1)}{L(x|H_0)}$ und interpretieren diesen Wert als χ^2 Verteilung

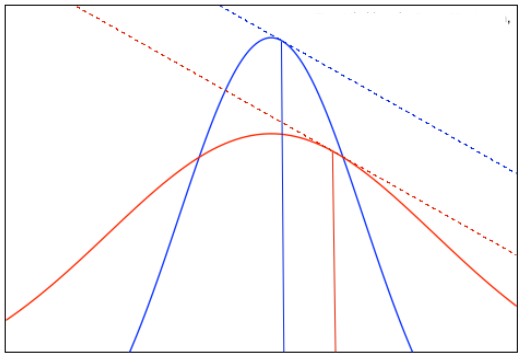
Und wir können daraus einen p-Wert bestimmen, der angibt ob Signal und Untergrund mit der Untergrund Hypothese verträglich ist.

$$p_H = \int_{t_{H,obs}}^{\infty} f(t_H|H)dt_H$$

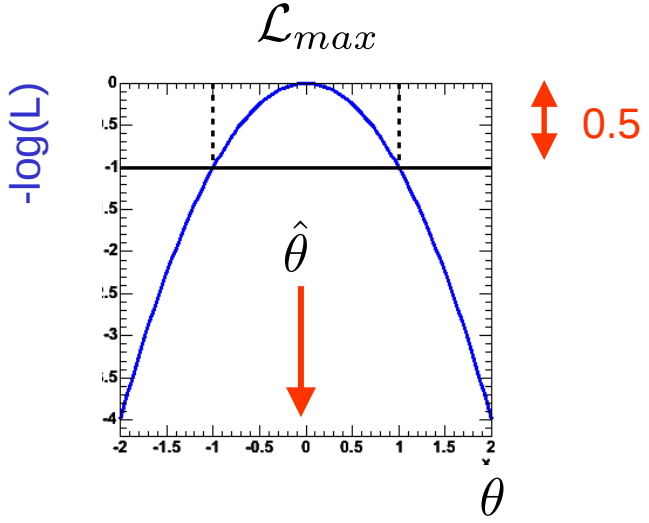
χ^2 Verteilung

Das geht nur unter der Annahme das beide Hypothesen funktional verknüpft sind.

Maximum Likelihood - Methode



- Variance der Maximum Likelihood Parameter
  - Im Minimum von $\mathcal{L}$ ist $\frac{d\mathcal{L}(\vec{x}|\vec{\theta})}{d\vec{\theta}} = 0$
  - Variance läßt sich aus der 2ten Ableitung am Minimum bestimmen



$$\mathcal{L}(\theta \pm \sigma) = \mathcal{L}_{max} - \frac{1}{2}$$

Der Parameterfehler ist durch die Änderung der log Likelihood am Minimum um 0.5 gegeben.

Taylor-Entwicklung:

$$\ln L(p) = \ln L(\hat{p}) + \left.\frac{d \ln L}{dp}\right|_{p=\hat{p}} (p - \hat{p}) + \tfrac{1}{2} \left.\frac{d^2 \ln L}{d^2 p}\right|_{p=\hat{p}} (p - \hat{p})^2$$

$$= \ln L_{\max} + \left.\frac{d^2 \ln L}{d^2 p}\right|_{p=\hat{p}} \frac{(p - \hat{p})^2}{2}$$

$$= \ln L_{\max} + \frac{(p - \hat{p})^2}{2\hat{\sigma}_p^2} \quad \Rightarrow \quad \ln L(p \pm \sigma) = \ln L_{\max} - \frac{1}{2}$$

$$\hat{\sigma}(p)^2 = \hat{V}(p) = \left(\frac{d^2 \ln L}{d^2 p}\right)^{-1}$$

Methode der kleinsten Quadrate

Methode (LS Methode = least square method) zur Bestimmung der optimalen Parameter von Funktionen an normalverteilte Messwerte.

- LS-Prinzip

  Betrachten wir eine Stichprobe mit n Messwerten y_i und der parametrisierten Beschreibung η_i der Messwerte. σ_i sind die Messfehler und x_i sind die abhängigen Werte. Die erwartete Abhängigkeit von der Wahrscheinlichkeitsdichte ist $\eta_i = f(x_i|\theta)$.
  Der Parametersatz $\theta = \theta_1, \theta_2, ..., \theta_k$ soll so bestimmt werden, das die Daten optimal beschrieben werden,

  also $$S = \sum_{i=1}^{n} \frac{(y_i - \eta_i)^2}{\sigma_i^2} = \sum_{i=1}^{n} \frac{(y_i - f(x_i|\theta))^2}{\sigma_i^2}$$ minimal wird.

  Für den Fall korrelierter Messwerte muss die Kovarianzmatrix der y_i Werte mit verwendet werden!
  S folgt einer χ^2- Verteilung mit $(n-k)$ Freiheitsgraden, d.h. Anzahl der Messungen minus der Anzahl der freien Parameter.

- LS-Methode und Maximierung der Likelihood Funktion sind für den Fall normalverteilter Messwerte äquivalent

Methode der kleinsten Quadrate

- Beispiel LS-Methode

Häufig ist die Anpassungsfunktion $f(x|\theta)$ linear in $\theta = \theta_1, \theta_2, ..., \theta_k$.

$$f(x|\theta) = \theta_1 f_1(x) + ... + \theta_k f_k(x)$$

Für die Hypothese, das die Messwerte auf einer Geraden liegen mit zu bestimmendem θ_1 und θ_2 , ($f_1(x) = 1$, $f_2(x) = x$) ist

$$f(x|\theta) = \theta_1 + \theta_2 x$$

Der LS-Ansatz ist dann

$$S = \sum_{i=1}^{n} \frac{(y_i - \eta_i)^2}{\sigma_i^2} = \sum_{i=1}^{n} \frac{(y_i - \theta_1 - x_i\theta_2)^2}{\sigma_i^2}$$

$$\frac{\partial S}{\partial \theta_1} = \sum_{i=1}^{n} \frac{-2(y_i - \theta_1 - x_i\theta_2)}{\sigma_i^2} = 0 \qquad \text{und}$$

$$\frac{\partial S}{\partial \theta_2} = \sum_{i=1}^{n} \frac{-2x_i(y_i - \theta_1 - x_i\theta_2)}{\sigma_i^2} = 0$$

erlauben die Bestimmung von θ_1 und θ_2 .

Allgemein:
Für lineare Anpassungsfunktionen lassen sich die Lösungen durch Matrix Inversion finden.

Vergleich der Methoden

Reference:
L. Lyons
Statistics for Nucler and Particle Physics
ISBN: 0-521-37934-2

Table 4.2. *Comparison of various techniques for parameter determination*

	Moments	Maximum likelihood	Least squares
How easy?	Very, provided suitable moments can be found	Normalisation and maximisation can be messy	Needs minimisation
Efficiency	Not very	Usually most efficient	Sometimes equivalent to max. like.
Input data	Individual events	Individual events	Histograms
Estimate of goodness of fit	Rather messy to obtain	Very difficult	Easy
Constraints among parameters	Cannot be imposed	Easy	Can be imposed
Are n-dimensional problems difficult?	Not if suitable moments can be found	Normalisation and maximisation get messier	Problem of which distributions to choose
Weighted events	Easy	Can be used	Easy
Overflow	Excluded	Included	Included
Background subtraction	Easy	Can be troublesome	Easy
Error estimate	From spread of individual values	$\left(-\frac{\partial^2 \ell}{\partial p_i \partial p_j}\right)^{-\frac{1}{2}}$	$\left(\frac{1}{2}\frac{\partial^2 S}{\partial p_i \partial p_j}\right)^{-\frac{1}{2}}$

MINUIT – Programmpaket zur Datenanpassung

Sowohl die Likelihood-Methode als auch die LS-Methode führen zu einem Minimierungsproblem von Funktionen. In den 1970iger Jahren hat F. James (CERN) das fortranbasierende Programmpaket MINUIT entwickelt, das das Minimierungsproblem löst. Der Nutzer stellt dabei eine zu minimierende Funktion $F(X, P)$ im Parameterraum $P = (p_1, ...p_n)$ zur Verfügung. Mit Hilfe von Funktionen kann auf den Minimierungsprozeß Einfluß genommen werden.

Das Programmpaket wird als C++ Portierung (MINUIT2) in ROOT verwendet.

Während des Minimierens wird $F(X, P)$ für verschiedene X ausgewertet, wobei F nicht notwendigerweise analytisch ist. Die Auswahl von P geschieht dabei mit 3 Methoden:

- SEEK
  Suche des Minimums mit Monte Carlo Methoden, es wird zu Beginn verwendet, wenn keine sinnvollen Startwerte bekannt sind
- SIMPLX
  Verwendet das Simplex-Verfahren von Nelder und Mead, dabei werden Vergleiche von Funktionswerten im Parameterraum vorgenommen. Eine Schrittweitensteuerung führt zur Annäherung an das Minimum.
- MIGRAD
  Verwendet einen Algorithmus von R. Fletcher, der die Funktion und den Gradienten in der variable metric method zum Finden des Minimums verwendet.
- HESSE
  Berechnet die Hesse matrix der zweiten Ableitung und daraus die Kovarianz Matrix
- MINOS
  Berechnet Fehler Intervalle mit Profile Likelihood Verhältnissen

MINUIT – Programmpaket zur Datenanpassung

- Die einzelnen Schritte können während des Minimierungsprozesses nacheinander auch mehrfach in anderer Reihenfolge aufgerufen werden
- Parameter p_i aus $P = (p_1, ...p_n)$ lassen sich in den einzelnen Schritten konstant setzen und wieder freigeben.
- Generell sind Probleme bei der Konvergenz für Modelle mit sehr stark korrelierten Parametern zu erwarten.
- Problematisch sind lokale Minima, Kanten/Sprünge oder undefinierte Bereiche der Likelihood Funktion

→ Benötigt häufig eine Vereinfachung der Modelle und Reduktion des Parameterraumes

MINUIT long writeup: https://root.cern.ch/download/minuit.pdf

Minimization Methoden in ROOT

Mit Hilfe einer common interface Klasse (ROOT::Math:Minimizer) erhält man Zugang zu verschiedenen über plug-ins implementierte FIT-Algorithmen.

- MINUIT → benutzt den in den 1970iger Jahren entwickelten FORTRAN code als Klasse `TMinuit` mit den Algorithmen Migrad,Simplex und Minimize
- Minuit2 → neue Implemetierung von Minuit in C++ mit den Algorithmen Migrad,Simplex, Minimize und Fumili2
- Fumili → nur für least-square und log-likelihood Minimierung
- GSLMultiMin → conjugate gradient minimization algorithm der GNU Scientific Library (GSL) Fletcher-Reeves, BFGS https://www.gnu.org/software/gsl/doc/html/multimin.html
- GSLMultiFit → Levenberg-Marquardt Algorithmus aus der GSL nur für least-square func. https://www.gnu.org/software/gsl/doc/html/lls.html
- Linear → direkte Lösungen für least-square Funktionen
- GSLSimAn → Simulated Annealing Algorithmus der GSL https://www.gnu.org/software/gsl/doc/html/siman.html
- Genetic → basiert auf dem genetic Algorithmus der in TMVA verwendet wird

• Die FIT Algorithmen stehen sowohl für fits in ROOT als auch in rooFIT/rooStats zur Verfügung

• Die Algorithmen können in Kombintion hintereinander ausgeführt werden

• Über den Minimizer können weitere Algorithmen implementiert werden

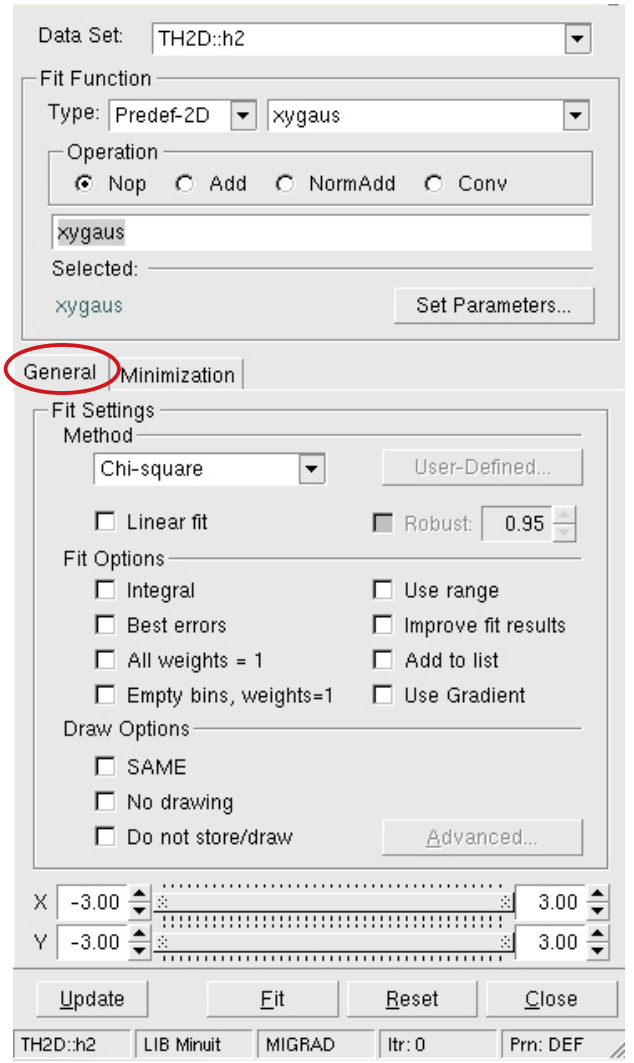
Fit Panel in ROOT

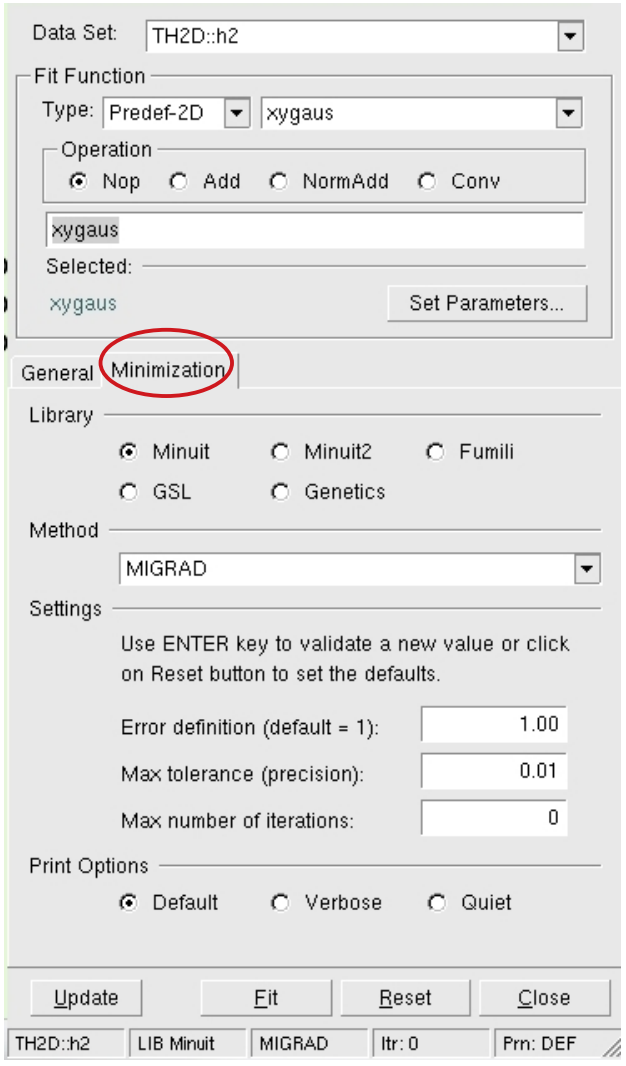
In ROOT steht ein GUI zur Datenanpassung zur Verfügung, das ROOT Objekte wie histogram, graph und trees bedient. Der Aufruf erfolgt über

```
Tbrowser/Canvas/Plot
  → Tools
  → FitPanel
```

- Auswahl des Datenobjekts
- Fit Modell
- Startwerte setzen
- Auswahl der Fit Methode
- Setzen der Fit Optionen
- Auswahl des Fit Algorithmus
- Einschränken des Fit Bereiches
- Advanded Drawing Tools
  Contours und Scan Minimum

- Anwendung:
  Erste Fit Resultate und
  Erstellen eines Prototypen





Anpassung an eine 2D korrelierte Gauss Verteilung mit dem Fit Panel

Arbeitsvorschlag:

- Schreiben Sie ein ROOT Macro, das die ROOT Funktion bigaus verwendet, um ein 2D Histogramm mit einem x/y Mean und Sigma von 1 zu füllen. Der Korrelationsfaktor soll 0.3 betragen. Zum Zeichnen verwenden Sie “colz” und “lego”.

- Führen Sie einen Fit mit bigaus im Fit Panel aus und probieren Sie verschiedene Fit Parameter und Methoden.

Gauss2D.C

Fitting in ROOT

Informationen zur Datenanpassung in ROOT

- https://root.cern/doc/master/group__tutorial__fit.html
- Beispiel Programme sind auch in $ROOTSYS/tutorial/fit zu finden

- Die Datenanpassung kann sowohl mit in ROOT definierten Funktionen als auch mit selbst definierten Funktionen an Daten in Histogramm, TGraph Objekten und ungebinnten Daten vorgenommen werden.

- Die Definition von Funktionen und Verwendung in ROOT wurde schon diskutiert

- Methoden, die den Fit beeinflussende Optionen konfigurieren, werden im Programm gerufen.

```
ROOT::Math::MinimizerOptions::SetDefaultMinimizer("Minuit2");
ROOT::Math::MinimizerOptions::SetDefaultTolerance(1.E-6);
gStyle→SetOptFit(1111);
```

`default` settings liefern oft sinnvolle Resultate, aber eine genaue Inspektion der Meldungen ist zwingend erforderlich (Einschalten von debug Optionen).

- Auf die Fitergebnisse und Fitfehler kann mit der Klasse `TFitResult` und `TFitResultPtr` zugegriffen werden.

ROOT – Fitting mit Minuit

Was passiert im Hintergrund?

- Erzeugung der Least Square oder Likelihood Funktion $F(X)$ (Minuit intern FCN)
- Minimierung bezüglich aller Parameter mit unterschiedlichen Algorithmen durchführen
- Fehler und Korrelationen der Parameter bestimmen

Parameter können während der Minimierung verschiedene Zustände annehmen:
- frei variierbar
- variierbar innerhalb von Grenzen
- fixed

Optionen steuern das Verhalten

ROOT – Fitting mit Minuit

- Fit Optionen

```
TFitResultPtr Fit(TF1 *func,Option_t *opt,Option_t *gopt,Axis_t xmin,Axis_t xmax)
```

The list of fit options is given in parameter option.

option	description
"W"	Set all weights to 1; ignore error bars
"U"	Use a User specified fitting algorithm (via SetFCN)
"Q"	Quiet mode (minimum printing)
"V"	Verbose mode (default is between Q and V)
"E"	Perform better Errors estimation using Minos technique
"B"	User defined parameter settings are used for predefined functions like "gaus", "expo", "poln", "landau". Use this option when you want to fix one or more parameters for these functions.
"M"	More. Improve fit results. It uses the IMPROVE command of TMinuit (see TMinuit::mnimpr). This algorithm attempts to improve the found local minimum by searching for a better one.
"R"	Use the Range specified in the function range
"N"	Do not store the graphics function, do not draw
"0"	Do not plot the result of the fit. By default the fitted function is drawn unless the option "N" above is specified.
"+"	Add this new fitted function to the list of fitted functions (by default, any previous function is deleted)
"C"	In case of linear fitting, do not calculate the chisquare (saves time)
"F"	If fitting a polN, use the minuit fitter
"EX0"	When fitting a TGraphErrors or TGraphAsymErrors do not consider errors in the coordinate
"ROB"	In case of linear fitting, compute the LTS regression coefficients (robust (resistant) regression), using the default fraction of good points "ROB=0.x" - compute the LTS regression coefficients, using 0.x as a fraction of good points
"S"	The result of the fit is returned in the TFitResultPtr (see below Access to the Fit Result)

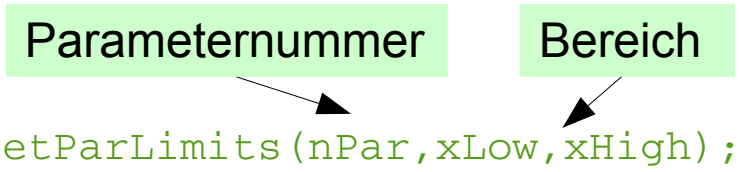
ROOT – Fitting mit Minuit

- Kontrolle über die Fit Parameter

Die Startwerte des Fits werden über das Setzen von Funktionsparametern erzeugt.

```
f→SetParameter(nPar,ParameterVal);
```

Der numerische Bereich der Fit Parameter läßt sich während des Fittens limitieren, in dem Bereichsgrenzen für die Parameter der Fit Funktion gesetzt werden.



Um Parameter konstant zu setzen, werden die obere und untere Bereichsgrenze identisch gewählt.

```
f→SetParLimits(nPar,xFixVal,xFixVal);
```

Eine weitere Möglichkeit Parameter für den Fit auf einen festen definierten Wert zu setzen ist die Methode

```
f→FixParameter(nPar,ParameterVal);
```

Der gesamte Fit Bereich einer Funktion kann über den Definitionsbereich der Fit Funktion eingeschränkt werden.

```
f→Fit("func","","",Axis_t xmin,Axis_t xmax);
```

ROOT – Fitting mit Minuit

- Abschnittweise Datenanpassung mit mehreren Funktionen

Der Fit Bereich einer Funktion kann über den Definitionsbereich der Fit Funktion eingeschränkt werden. Das erlaubt abschnittsweise verschiedene Funktionen zu definieren und zu fitten.

```
// Function:  exp + Gauss
double funcExpGauss(double* bi, double* par){
   double x = bi[0] ;
   return (par[0]*exp(-0.5*pow(((x-par[1])/par[2]),2))+par[3]*exp(par[4]*x));
}
// Function:  Gauss
double funcGauss(double* bi, double* par){
   double x = bi[0] ;
   return (par[0]*exp(-0.5*pow(((x-par[1])/par[2]),2));
}
// Function:  exp
double funcExp(double* bi, double* par){
   double x = bi[0] ;
   return (par[0]*exp(par[1]*x));
}
…..
TF1 *fitexpgaus = new TF1("fitexpgaus",funcExpGauss,0.,8.,5);
TF1 *fitexp = new TF1("fitexp",funcExp,0.,2.,2);
TF1 *fitgaus = new TF1("fitgaus",funcGauss,2.2,4.2,3);
…...
hist→Fit(fitexpgaus,"R");      Fit mit Fkt fitexpgaus an Daten in hist  im definierten Bereich
hist→Fit(fitexp,"R+");         Hinzufügen des Fit mit Fkt fitexp im definierten Bereich
hist→Fit(fitgaus,"R+");        Hinzufügen des Fit mit Fkt fitgaus im definierten Bereich
```

ROOT – Fitting mit Minuit

- Integral mit Fehlerfortpflanzung

```
TVirtualFitter * fitter = TvirtualFitter::GetFitter();
…..
hist→Fit("fitgaus","0");
…..
double integral = fitgaus→Integral(RangeLow,RangeUp);
double * covMatrix = fitter→GetCovarianceMatrix();
double sigma_integral = fitgaus->IntegralError(RangeLow,RangeUp);
```

- Konfidenzintervall des Fitresultats für Histogramme

```
hist→Fit("fitgaus","0");
TVirtualFitter * fitter = TvirtualFitter::GetFitter();
…..
TH1D *confi = new TH1D("confi","Fitted function with 95 conf.band",100,0.,5);
fitter→GetConfidenceIntervals(confi);

// Draw confidence level
confi->SetStats(kFALSE);
confi->SetFillColor(7);
confi→Draw("E3");

// Draw data
hist->SetMarkerColor(4);
hist->SetMarkerStyle(20);
hist->Draw("E SAME");
```

default 0.95, oder anderen Wert setzen

Reihenfolge beim Zeichnen: Daten zuletzt

ROOT – Fitting mit Minuit

- Konfidenzintervall des Fitresultats für Graphen

```
TFitResultPtr fp = Graph->Fit(fitFunc,"MES");

…...
TGraphErrors *GraphConf = new TgraphErrors(nPoints);
GraphConf->SetTitle("Fit Polynomial with .95 confidence;x;f(x)");
for (int i=0; i<nPoints; i++)
      GraphConf->SetPoint(i, Graph->GetX()[i], 0);

(TVirtualFitter::GetFitter())->GetConfidenceIntervals(GraphConf);
…...
// Draw confidence level
GraphConf→SetLineColor(kBlue);
GraphConf→Draw("AP");

// Draw data
Graph->SetMarkerColor(kRed);
Graph->SetMarkerStyle(21);
Graph->Draw("P SAME");
```

default 0.95, oder anderen Wert setzen

ROOT – Fitting mit Minuit

- Zugriff auf Fit Parameter und Resultate

```
#include <TMinuit.h>
…..
//set default to Minuit since we will use gMinuit
TVirtualFitter::SetDefaultFitter("Minuit");
…..
// Create fit Function
TF1* f = new TF1("f",myFunc,xRmin,xRmax,nDim);
…..
// Perform the fitting
TFitResultPtr fp = Graph->Fit(f, "S");
…..
// Get fit results
double ParVal = f→GetParameter(0);
double SigmaVal = f→GetParError(0);
double chi2 = f→GetChisquare();
int ndof = f→GetNDF();
// Get covariance matrix
TMatrixDSym V = fp→GetCovarianceMatrix();
// Get inverse covariance matrix, initialize Vinv with V
TMatrixDSym Vinv(V);
Vinv.Invert();
// print fit results
fp->Print();
```

Auf die Fit Resultate kann über die Funktion zugegriffen werden

Zugriff über die Klasse TFitResult

ROOT – Fitting mit Minuit

Resultat eines Polynom 3ter Ordnung Fits an 10 Datenpunkte:

Algorithmus

Ergebnis

Fit result correct

```
Processing myFit.cc...
 FCN=3.66649 FROM MIGRAD    STATUS=CONVERGED      161 CALLS         162 TOTAL
                     EDM=8.64467e-08    STRATEGY= 1      ERROR MATRIX ACCURATE
  EXT PARAMETER                                          STEP         FIRST
  NO.   NAME      VALUE            ERROR                 SIZE      DERIVATIVE
   1  p0           -1.09959e+00   6.36479e-01   6.41805e-05  -1.42436e-03
   2  p1            2.59017e+00   4.34582e-01   8.53173e-06  -8.59404e-03
   3  p2           -4.95702e-01   8.03164e-02   9.89440e-07  -2.56661e-02
   4  p3            2.64949e-02   4.37554e-03   1.08931e-07   1.22836e-01

Chi2 = 3.66649
NDf  = 6
```

(Anzahl der Datenpunkte – Parameter)

[Error(p0)] ^{2}

Kovarianzmatrix:

```
          0             1             2             3
0     0.4051050    -0.2647390    0.04657980   -0.00242942
1    -0.2647391     0.1888611   -0.03451851    0.00184014
2     0.04657982   -0.03451852   0.006450732  -0.000349284
3    -0.002429423   0.001840143  -0.000349284   1.91454e-05
```

Datenanpassung an ein Histogramm

Arbeitsvorschlag:

- Schreiben Sie ein ROOT Macro, das den Tree “FitTest” aus dem ROOT File FitTestTTree.root liest und ein Modell anpasst.

- Welche Modell lässt sich verwenden? Geben Sie den Fitparametern Namen.

- Geben Sie möglichst viel Kontrollinformation während der Fit Prozedur aus.

- Schalten Sie Optionen ein, die Fit und Fitfehler verbessern.

- Greifen Sie auf die Fit Informationen im Programm zu.

- Zeichnen Sie die Daten, den Fit und ein 95% Konfidenzinterval. Kontrollieren Sie die Parameterausgabe im Plot.

- Trennen Sie den Fit in Signalbereich und Untergrundbereich. fitTestAnalyse.C

- Wie können wir die Anzahl der Ereignisse einschliesslich Fehler des Signalanteils bestimmen?

ROOT – Fitting mit Minuit

Die Fehler der Fit Parameter werden während des Fit Prozesses immer mitgeführt und können jederzeit dargestellt werden.

- Bedeutung der Fehler:

  CURRENT GUESS ERROR → not to be believed
  APPROXIMATE ERROR → die Fehler wurden berechnet aber sind nicht genau
  ERROR MATRIX ACCURATE → Minuit glaubt, das die Fehler richtig sind

- Hinweise auf problematische Fits:

  Warnungsmeldungen beim Minimierungsprozess
  Kein Minimum gefunden
  Grosser EDM Wert
  Korrelationskoeffizienten sind genau 0 oder sehr nahe 1
  Covariance matrix ist nicht positiv-definite
  Parameter sind am Rand der gesetzten Limits
  Numerische Probleme
  Unphysikalische lokale Minima

ROOT – Fit Fehler

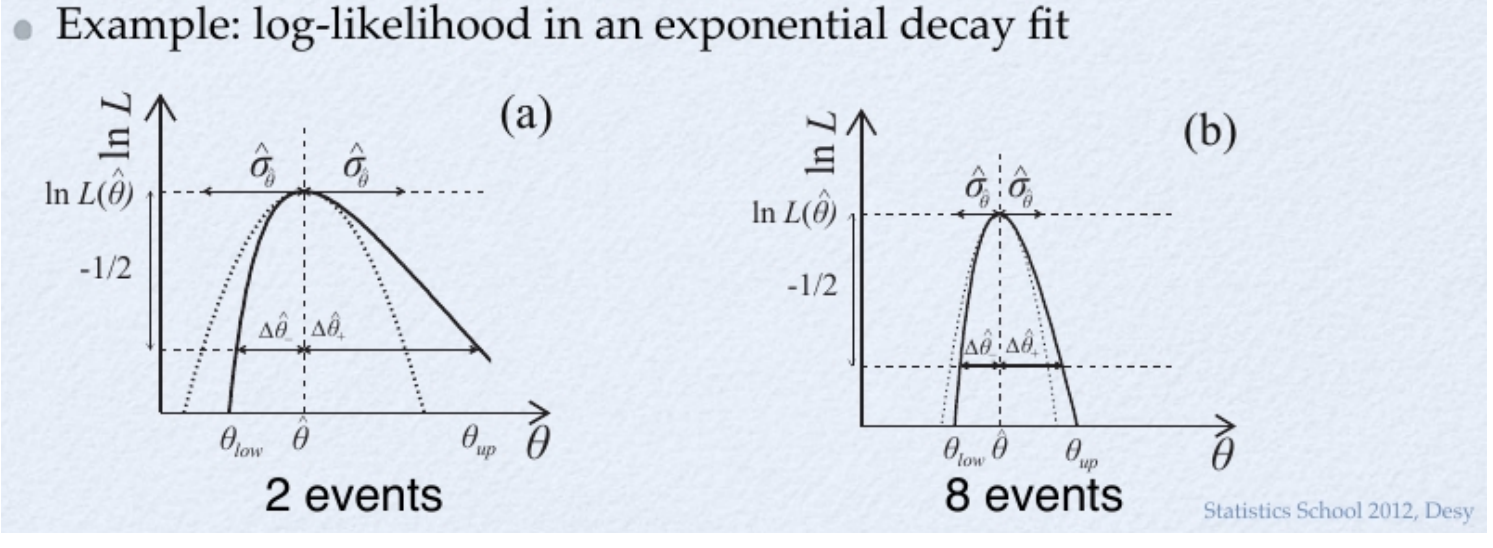
Innerhalb von MINUIT wird für einen Fit das Minimum im Parameter-Raum mit einer Funktion gesucht, die entweder einen Least Square (χ^2) Ansatz oder einen Likelihood Ansatz verwendet. Das Verhalten der Funktion am Minimum (2. Ableitung) definiert die Fehler der Parameter. In

$$\chi^2 = \sum_{i=1}^{n} \frac{(y_i - \eta_i)^2}{\sigma_i^2} = \sum_{i,j=1}^{n} (y_i - \eta_i) V_{ij} (y_j - \eta_j)$$

müssen die σ_i^2 bzw. V_{ij} Messfehler wiederspiegeln.

Im Fall der Likelihood Methode wird zur Berechnung der Fehler die 2. Ableitung der Likelihood Funktion verwendet. Aymptotisch sind die Schätzwerte der Parameter $\hat{\theta}$ gaussverteilt mit folgender Korrelationsmatrix

$$\hat{V}(\hat{\theta}) = \left[\left(-\frac{\partial^2 lnL(x;\theta)}{\partial^2 \theta} \right)_{\theta=\hat{\theta}} \right]^{-1}$$



ROOT – Fit Fehler

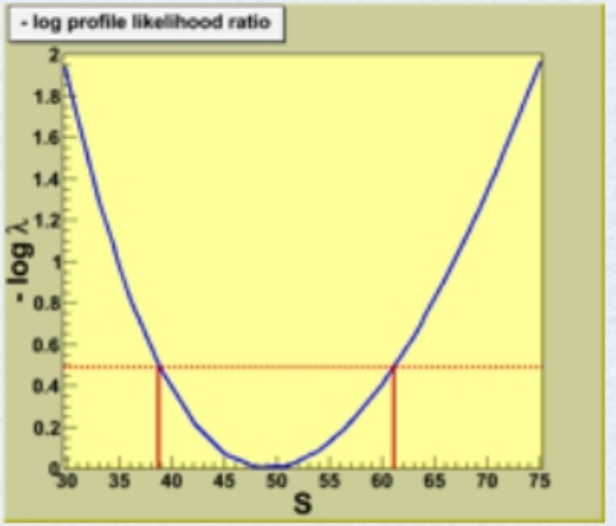
- A better approximation to estimate the confidence level in the parameter is to use directly the log-likelihood function and look at the difference from the minimum.

$$\lambda(\theta) = \frac{L(x|\theta)}{L(x|\hat{\theta})} \qquad -2\log\lambda(\theta) \approx (\theta-\hat{\theta})^T H(\theta-\hat{\theta})$$

$$-2\log\lambda(\theta) \sim \chi^2 \text{distribution}$$

$$-\log\lambda(\theta_{low} \equiv \hat{\theta} - \delta\hat{\theta}_-) = -\log\lambda(\theta_{up} \equiv \hat{\theta} + \delta\hat{\theta}_+) = \frac{1}{2}F^{-1}_{\chi^2}(0.68, 1) = 0.5$$

  - Method of Minuit/Minos (Fit option “E”)
    - obtain a confidence interval which is in general not symmetric around the best parameter estimate




```
TFitResultPtr r = h1->Fit(f1,"E S");
r->LowerError(par_number);
r->UpperError(par_number);
```

ROOT – Fit Fehler

Die Kovarianzmatrix der Fitparameter zeigt die Kopplung der einzelnen Parameter untereinander. Die Fitfehler der Parameter sind also keine quadratischen Boxen, sondern Ellipsen und können für jeweils 2 Parameter als Contour dargestellt werden.

- Beispiel

```
…….
// contours in 2 dimensions
TCanvas *c2 = new TCanvas("c2","contours",10,10,600,800);
c2->Divide(1,2);
c2->cd(1);
//Get first (1 sigma) contour for parameter 0 versus parameter 1
gMinuit->SetErrorDef(1); //setting for 2 sigma, note 4 and not 2!
TGraph *gr01 = (TGraph*)gMinuit->Contour(40,0,1);
gr01->Draw("alp");
```

ROOT – Fit Stabilität

Manchmal werden während des Fit Prozesses lokale Minima gefunden, dadurch ergeben sich falsche Fit Parameter.

- Ändern der start Parameter
- Wahl eines anderen Algorithmus → simulated annealing oder genetic

Häufig werden Fit Prozesse ohne Konvergenz beendet

- Fehlermatrix ist nicht positiv definit → Startparameter überprüfen / ändern
- Numerische Probleme der zu minimierenden Funktion → Fitmodel überprüfen
- Grosse Korrelationen in den Fit Parametern → Fitmodel überprüfen

Die Parameter von Polynomen sind “stark” korreliert

$$F(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3 + ...$$

- Verwendung von orthogonalen Polynomen, z.B. ChebyChev Polynome, im Fitmodell

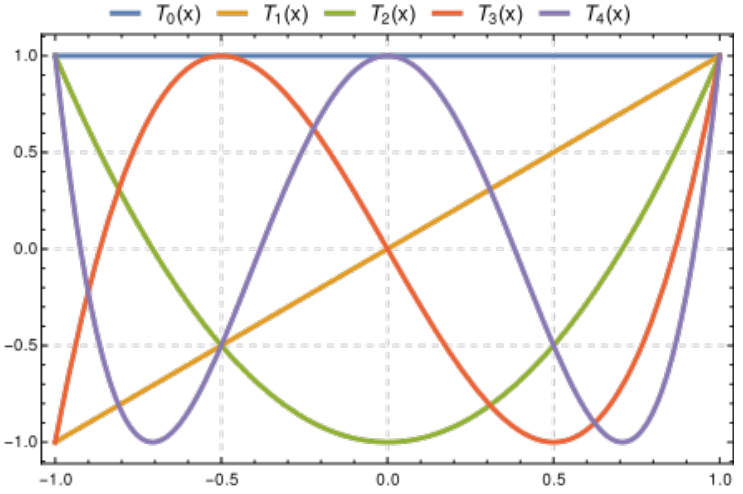
$$T_0(x) = 1$$

$$T_1(x) = x$$

$$T_2(x) = 2x^2 - 1$$

$$T_3(x) = 4x^3 - 3x$$

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$



ROOT – Graphische Darstellung

Drei Klassen (TGraph, TGraphErrors, TGraphAsymmErrors) können in ROOT zur graphischen Darstellung von Messungen verwendet werden. Grundsätzlich werden die Zahl der Messpunkte und Arrays mit den darzustellenden Werten gegeben.

```
const int nHisto = 6 ;
double Range[nHisto+1] ={0.,1.5,2.9,3.5,5.0,7.0,9.5};
// Die Funktionswerte stehen in Mean und MeanErr
double Mean[nHisto], MeanErr[nHisto];                    y-axis
double RangeBin[nHisto], RangeBinErr[nHisto];            x-axis
for(int j=0; j<nHisto;j++){
  RangeBin[j] = Range[j]+(Range[j+1]-Range[j])/2. ;
  RangeBinErr[j] = (Range[j+1]-Range[j])/2. ;
}
TGraphErrors *grRangeDep = new
       TgraphErrors(nHisto,RangeBin,Mean,RangeBinErr,MeanErr);
// Draw onto Canvas
grRangeDep→→SetTitle("Fit Polynomial;x;f(x)");
grRangeDep→SetMarkerColor(kRed);
grRangeDep→SetMarkerStyle(21);
grRangeDep→Draw("AP");
grRangeDep→GetXaxis()→ SetLimits(Xmin,Xmax);
grRangeDep→GetYaxis()→ SetLimits(Ymin,Ymax);
MyCanvas→Modified();
MyCanvas→Update();
```

Zeichnen des Graphen

Anpassen der Achsenskala

Update Canvas

Graphs' plotting options

Graphs can be drawn with the following options:

Option	Description
"A"	Axis are drawn around the graph
"I"	Combine with option 'A' it draws invisible axis
"L"	A simple polyline is drawn
"F"	A fill area is drawn ('CF' draw a smoothed fill area)
"C"	A smooth Curve is drawn
"*"	A Star is plotted at each point
"P"	The current marker is plotted at each point
"B"	A Bar chart is drawn
"1"	When a graph is drawn as a bar chart, this option makes the bars start from the bottom of the pad. By default they start at 0.
"X+"	The X-axis is drawn on the top side of the plot.
"Y+"	The Y-axis is drawn on the right side of the plot.
"PFC"	Palette Fill Color: graph's fill color is taken in the current palette.
"PLC"	Palette Line Color: graph's line color is taken in the current palette.
"PMC"	Palette Marker Color: graph's marker color is taken in the current palette.

Fit einer in TGraph dargestellten Messung

Datenpunkte:

```
// data to be fit by a polynomial
const int nP =  10 ; // Number of data points
double xData[nP]= {1.  ,2.  ,3.  ,4.  ,5.  ,6.  ,7.  , 8. ,9.  , 10. };
double yData[nP]= {1.1 ,2.3 ,2.7 ,3.2 ,3.1 ,2.4 ,1.7 ,1.5 ,1.5  ,1.7};
double dyData[nP]= {0.15,0.22,0.29,0.39,0.31,0.21,0.13,0.15,0.19,0.13 };
double dxData[nP] = {0.1,0.1,0.5,0.1,0.5,0.1,0.5,0.1,0.5,0.1};
```

Arbeitsvorschlag:

- Schreiben Sie ein ROOT Macro, das die gemessene Daten darstellt.
- Fitten Sie ein Model in Form eines Polynoms an die Daten. Wieviele Parameter verwenden Sie
- Zeichnen Sie das Fitresultat.
- Ignorieren Sie die Fitfehler der x-Koordinate.
- Geben Sie die Fitparameter, die Fehler, die Zahl der Freiheitsgrade und Kovarianzmatrix aus Und stellen Sie die Werte im Programm zur Verfügung.
- Zeichnen Sie das 0.95 Konfidenzintervall
- Zeichnen Sie die Fehlerkonturen von jeweils 2 Parametern.
- Reduzieren Sie die Anzahl der Parameter und vergleichen Sie die Fitresultate.

myFit.C