After hard work you have got nice data...
You just want to click on button...

Run the fit
In order to get a nice fit!
However there is a high probability that you get into trouble...

Hopefully this lecture will help you understand the fit process and avoid traps.
The normal probability law

*In many detectors (notably gaz detectors) counting follows a Poisson distribution law. As soon as the counting rate is high the « normal law » becomes a good approximation. It is much easier to handle.*

« bell » curve

**Normal** distribution

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

- **Expected value**: \( \mu \)
- **Mean square deviation**: \( \sigma \)
- **Variance**: \( \sigma^2 \)

**Poisson** distribution

\[ P(x; \mu) = \frac{(e^{-\mu}) (\mu^x)}{x!} \]

- **Expected value**: \( \mu \)
- **Mean square deviation**: \( \mu \)
- **Variance**: \( \mu \)
In this lecture all examples will be Small Angle Scattering experiments. Therefore we remind below how such experiments are performed: a (quasi) monochromatic beam crosses a sample (possibly anisotropic); the outcoming neutrons scattered at a small angle of the main beam build an image in a 2 dimensions detector.

However most of the fit principles apply to any type of experiment and modelling.
Data treatment

G. Pépy
Wigner Research Center for Physics, P.O. Box 49, H-1525 Budapest, Hungary
gpepy@laposte.net

Summary:

1 – Generalities about the fit process

2 – least squares, steepest descent...

3 – SANS

4 – error bars

5 – exotic techniques

6 – Examples from the PXY and HWXY 2D softwares

7 – Developments

8 – Thanks
1 – Generalities about a fit process

**Least square fit**

Define a distance  $\chi^2$

$$\chi^2 = \frac{1}{N - p} \sum_{i=1}^{i=N} \left( \frac{I_i - Y_i\{P\}}{\Delta I_i} \right)^2$$

- $I_i$: intensity in pixel $i$
- $N$: number of data points
- $P$: number of free parameters
- $\{P\}$: set of parameters
- $Y_i$: calculated intensity in pixel $i$
- $\Delta I_i$: uncertainty of $I_i$

If the random variables $I_i$ are independent and follow a normal probability law $\chi^2 \to 1$

which is a good test for the fit
Least square fit

In the vicinity of the $\chi^2$ minimum
Develop the $\chi^2$ up to 2nd order

Calculate the 1st order derivatives versus the parameters

Hypothesis: close to the $\chi^2$ minimum all derivatives are 0

This hypothesis provides a linear equation system

$[\alpha]$ is the curvature matrix

Whose solution (matrix inversion) provides a vector of parameters increments

$[C]$ is the covariance matrix

If you are lucky this vector leads to a next point in the parameter hyperspace where $\chi^2$ is smaller. It may not be the case as all this is highly non linear; one may try to shorten the vector keeping the same direction...

Then one repeats (iterates) the process.
If the $\chi^2$ is not a quadratic form this process is not efficient (the hypothesis and calculation on derivatives are bad)

One prefers the **steepest descent method**, taking a step $C$ along the 1st derivative vector. $C$ is now simply a scalar.

**Levenberg-Marquardt** have proposed a clever method to pass from “steepest descent” to “least square”:

Replace $\alpha$ by $[\alpha]' = [\alpha] + \lambda [I]$ (i.e. multiply the diagonal elements of the curvature matrix by $1 + \lambda$)

1) start with a modest $\lambda \sim 1$,
2) compute $[\alpha], \beta$ (and save it) and $\chi^2$ and save them
3) calculate the parameter increments with $[\alpha]'$ diagonal $\chi^2$ elements
4) compute new parameters and the corresponding $\chi^2$
5) if the fit has converged, or too many iterations, stop!
6) if the fit improves, keep new parameters, divide $\lambda$ by 10 and return to 2)
7) if the fit worsens, multiply $\lambda$ by 10, return to 3)

( no new computation of $[\alpha]$ needed, so is efficient)

**NOTE** - to obtain the proper error estimates on parameters set $\lambda = 0$ for a final calculation.
Least squares follows A (well behaved, else might work using partial shifts) or B (blows up), steepest descent follows C or D (local minimum), Marquardt steers between B & D or A& C but might fall into the local minimum.

\[ \lambda \] is small for least squares or \[ \lambda \] is large for steepest descent

This slide and the previous one were « inspired » from a talk at FLNP, Dubna, by Dr. Richard Heenan, ISIS Facility,
Some problems

- non linearity
- the minima depths differentiate best if the pixels uncertainties are small
- systematic errors in the data or the model \( \rightarrow \chi^2 \neq 1 \)
- nature of the probability law: Poisson or... worse
- bad evaluation of the pixel uncertainties
Before trying to model SANS results, one has generally to make basic data corrections: the most important is calibration (thanks to a flat scatterer like $H_2O$...). Others consist in sample background or calibration background subtractions. Smoothing data may help avoid Poisson law. Pseudo-calibration decreases the effect of calibration file uncertainties. **High quality experiments exhibit small uncertainties.**
Data error bars (uncertainties)

\[
\left( \frac{d\Sigma}{d\Omega} \right) = F_w \frac{I_s/C_s - I_r/C_r}{I_w/C_w - I_c/C_c}
\]

\[
C_s = \frac{L_{SDs}^2}{A_s d_s M_s T_s}
\]

Let the index hold for:
- \(s\) sample
- \(r\) reference
- \(w\) calibration (water)
- \(c\) background (sample container)

\[
\Delta\left( \frac{d\Sigma}{d\Omega} \right) = F_w \left\{ \frac{\Delta I_s/C_s}{I_w/C_w - I_c/C_c} + \frac{\Delta I_r/C_r}{I_w/C_w - I_c/C_c} + \frac{\Delta I_w}{C_w} \frac{I_s/C_s - I_r/C_r}{(I_w/C_w - I_c/C_c)^2} + \frac{\Delta I_c}{C_c} \frac{I_s/C_s - I_r/C_r}{(I_w/C_w - I_c/C_c)^2} \right\}
\]

Everything involved in the correction process will increase the final uncertainties.

**Beware of poor quality background data!**
Parameters error bars

The parameter error bars are best obtained after a least square fitting converges

Let us consider the $\chi^2$ expression:

$$\chi^2 = \chi^2_{\text{min}} + \delta \vec{p}[\alpha] \delta \vec{p}$$

and the $\chi^2$ change when a single parameter is changed

$$\Delta \chi^2_1 = \chi^2_{M-1} - \chi^2_{\text{min}}$$

It is possible to show that $\Delta \chi^2_1$ has a probability distribution as a normal law

Therefore one may write

$$\Delta \chi^2_1 = \left(\delta p_1\right)^2 / C_{11}$$

with $C_{11}$ the diagonal element of the covariance matrix for parameter 1

And as for $\Delta \chi^2_1 < 1$ \( \delta p_1 < 1\sigma \) (68% of cases, 1st standard deviation)

$$\delta p_i = \sqrt{C_{ii}}$$
Robust fitting

Limitations of Least-Squares:
- data span many orders of magnitude
- significant outliers are present

Robust fitting proposes to modify the minimisation function by $w_i$ weights.

\[
\chi^2 = \frac{1}{N - P} \sum_{i=1}^{N} (w_i e_i)^2 \quad e_i = \frac{I_i - Y_i \{P\}}{\Delta I_i}
\]

<table>
<thead>
<tr>
<th>model</th>
<th>parameter k</th>
<th>$e_i &lt; k$</th>
<th>$e_i &gt; k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Huber</td>
<td>1.345</td>
<td>$w_i = 1$</td>
<td>$w_i = k/e_i$</td>
</tr>
<tr>
<td>bisquare</td>
<td>4.685</td>
<td>$w_i = (1-(e_i/k)^2)^2$</td>
<td>$w_i = 0$</td>
</tr>
</tbody>
</table>

$\{I_i\}$ intensity in pixel $i$
$N$ number of data points
$P$ number of free parameters
$\{P\}$ set of parameters
$Y_i$ calculated intensity in pixel $i$
$\Delta I_i$ uncertainty of $I_i$
$w_i$ weight
$e_i$ estimator / residual

Huber P 1981 *Robust statistics* (New York: Wiley)
Fox J 2002 *An R and S-PLUS Appendix Companion to Applied Regression*
« Parabolic » smoothing

Each pixel intensity is replaced by a weighted average over the neighbouring pixels

This modifies the uncertainty of pixel $i$

Which becomes:

$$\Delta I_i = \sqrt{\sum_j (w_j \Delta_j)^2}$$

<table>
<thead>
<tr>
<th></th>
<th>1/16</th>
<th>1/8</th>
<th>1/16</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/8</td>
<td>1/4</td>
<td>1/8</td>
<td></td>
</tr>
<tr>
<td>1/16</td>
<td>1/8</td>
<td>1/16</td>
<td></td>
</tr>
</tbody>
</table>

Table 1

Pseudo-calibration file

The underlying hypothesis is that the detector cells efficiency is mostly due to the pre-amplifiers efficiency. For most gaz detector there is one preamplifier for each line and column. Thence the trick:

Calculate 2 vectors as the average of each line, $\vec{A}_i$ and each column, $\vec{B}_j$

and build a new calibration file as a normalised product of these 2 vectors

$$[C] = \frac{1}{\langle C \rangle} t\vec{A}_i \otimes \vec{B}_j$$

$\langle C \rangle$ is the average of the calibration file

$t$ means

The uncertainty is reduced by a factor equal to the number of cells in a row.

This works very well for calibrators as vanadium or Plexiglas, not so well for H$_2$O as it exhibits a « cuvette » effect at larger $Q$. 
Simulated annealing

Fighting local minima:
Accept worse solutions to allow for a more extensive search for the optimal solution. The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to improve its properties.

Initialization of the method:
- determine a parameter neighbourhood where the parameters will be picked randomly
- chose a start « temperature » $T$
- the system energy $E$ will be the function to minimize: $\chi^2$
- define an acceptance function $P$

To avoid becoming stuck at a bad local minimum $P$ must be positive even when $\Delta E > 0$ ($\Delta E = E_{\text{new}} - E_{\text{old}}$). When $T$ tends to zero, $P$ must tend to zero if $\Delta E > 0$ and to a positive value otherwise.

- for a given $T$ define a maximum number of trials, $n_{\text{Trials}}$ (it may be one)
- define how to decrease $T$ and a maximum number of steps

$$P = \exp\left(-\frac{\Delta E}{T}\right)$$
Simulated annealing

Iteration
- calculate $E$ for a new set of parameters in the neighbourhood
- if $\Delta E < 0$ accept the new set
- if $\Delta E > 0$ accept the new set with a probability $P$ (Metropolis rule)
- if $n < n_{\text{Trial}}$ try a new set, otherwise decrease $T$
- iterate

Small uphill moves are more likely than large ones.
The system will increasingly favor moves that go "downhill". With $T=0$ the procedure reduces to the greedy algorithm, which makes only the downhill transitions.

Discussion:
This method is very much time consuming.
Not clear that it is useful if
- the parameter starting set is carefully chosen
- the experiment quality is good: big difference between local minima and optimal minimum
General problems of a 2D fitting software

Main features of a high level data treatment:

- The measurements quality
- The data and results display
- The reproducibility
3 – Examples

a) Do converging beams create anisotropy?
b) A problem of systematic error and statistics
c) Sheared liquid crystal polymer. Subtracted representations
   Liquid crystal polymer + contamination a smectic Bragg peak
a) Fit a Lorentzian anisotropy versus an anisotropic Porod scattering
b) 2 data files fit with a single model
c) Représentation and model function in polar coordinates
d) « Embedded » data files, together fitted
e) SAXS by nanochannels (track etched polymer membranes)
f) SANS by nanochannels (mica), 2 data files together
Do 2 beams converging at the detector generate anisotropy?
A problem of systematic error and of statistics.
Sheared side chain liquid crystal polymer [1]
Weighted difference: measure - model
SANS by a liquid crystal polymer, PAOCH$_3$ [2]

While slower than the classic steepest descent process, the various MINUIT process are always more efficient, especially when the start parameters are far off.

<table>
<thead>
<tr>
<th></th>
<th>Khi2</th>
<th>Khi2 calls</th>
<th>Time s</th>
</tr>
</thead>
<tbody>
<tr>
<td>RKGP</td>
<td>0.918</td>
<td>180</td>
<td>0.68</td>
</tr>
<tr>
<td>MINUIT</td>
<td>0.8362</td>
<td>763</td>
<td>1</td>
</tr>
<tr>
<td>MINUIT strategy 2</td>
<td>0.8367</td>
<td>1509</td>
<td>1.99</td>
</tr>
<tr>
<td>MINUIT Huber</td>
<td>0.8538</td>
<td>1096</td>
<td>1.52</td>
</tr>
</tbody>
</table>

far start parameters

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RKGP</td>
<td>5.8813</td>
<td></td>
<td>1.88</td>
</tr>
<tr>
<td>MINUIT</td>
<td>0.84</td>
<td>1345</td>
<td></td>
</tr>
<tr>
<td>MINUIT strategy 2</td>
<td>0.8362</td>
<td>1521</td>
<td>2.08</td>
</tr>
<tr>
<td>MINUIT Huber</td>
<td>0.8535</td>
<td>907</td>
<td>1.28</td>
</tr>
</tbody>
</table>

Data obtained on spectrometer PAXY, LLB, Saclay $\lambda=0.4$nm $D=2$m

The sample is a side chain liquid crystal polymer in the smectic A phase, hence the Bragg peak on the left of the picture. Half the chains are fully deuterated; the contrast with the protonated ones produces the central scattering on the right, which shows that the chains are restricted between the smectic layers.

While slower than the classic steepest descent process, the various MINUIT process are always more efficient, especially when the start parameters are far off.
SANS by a liquid crystal polymer, anisotropy? [3]

The lorentz/lorentz scattering by the main chain of this liquid crystal polymer is polluted by a strong anisotropic power law, due to the scattering of anisotropic catalyst particles.

All process are equivalent.
Fit of 2 data files with the same model function
A few parameters (widths) differ, the others are kept the same [4]

A nematic liquid crystal polymer (D11 spectrometer, ILL, Grenoble) at 2 different temperatures. The model is common to both files. Only the widths are independent. The prefactor is the same.
Fully oriented biaxial micels (PAXY spectrometer, LLB, Saclay. The model is the product of a Gaussian versus the radius and 2 symmetric Lorentzian versus the angle. This is a rare case when a polar representation is best. Pixels in the filter are shown from different points of view: radius and angle.
Empty bubbles in tungsten wires (D11 spectrometer, ILL, Grenoble). The model is a set of ellipsoids with a Schultz-Zimm size distribution + a centered power law. The measurement took place at 2 different distances, 6 and 28m in order to increase the Q range.
Model representation

Porod and Porod + sphere
Porod and Porod + ellipsoid
Porod and sphere + ellipsoid

W-un, W-970, W-1420, W-2470
Nanochannels in polycarbonate foils (ID01 beam line, ESRF, Grenoble) [9]. A rough sample surface produces Porod scattering singled out in the red filter. The nanochannel model is a cylinder form factor, product of a Bessel function (green oscillating radial part) and a fast decreasing sine (blue longitudinal part).
SANS by rhombohedral nanochannels in mica

The data files were obtained at the LLB, Saclay, for two orientations of the channels at 90°.

Both files were fitted together, with the same function, exchanging $R_a$, $R_b$

$\lambda = 1.4\text{nm}$  $D = 14\text{m}$  $R_a = 45\text{nm}$  $R_b = 25\text{nm}$

Courtesy of C. Trautmann, GSI, Germany
Polymer conformation in nano & mesoporous polymer nanotubes [8]

Nanochannel simplified model + isotropic lorentzian chain form factor model.
The dotted line is without the chain scattering.
USANS in Julich

Graph

Date Map

Filename: juel00001.dat
Wavelength: 12.7
Distance: 9300
Sample trans: 1.000
Refer trans: 1.000
Calib trans: 1.000
Calib average: 1.000

Sample centre: 84.8
Beam size: 127.38
Cell size: 0.40
Average: 73.92
Minimum: 0

Filter: 3 rectangles
80 90 80 80
140 150 100 180
40 120 138 144
Total: 491462.3
cell by cell calibration
smoothing < 100

2_dim_XY

Model description:
cell unit: 2_dim_XY
khs2: 0.970
R%: 0.000
Intensity: 0.981
iso_power: 0.981
anisotropy: 2.153
exponent: 2.153
Schim_dist: 0.180

Ellipsoid:
11.200E3 1.1200E3
Intensity: 4.194E3 4.194E3
width: 9.632 9.632

Graph

Date Map

 Functions: Schim_dist 32

Maximum:
7098.84
x=80, y=125

Legend:
- Clear
- Quit
- Store group
- Store file
- Fit
- Hole
- Previous
- Model
- Image
- Mask/filter
- Brightness
Developements ...

- users models
- MINUIT in HWXY
- parallel calculation
Thanks to contributors

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noirez L</td>
<td>LLB, Saclay</td>
</tr>
<tr>
<td>Martin N</td>
<td>LLB, Saclay</td>
</tr>
<tr>
<td>Len A.</td>
<td>Wigner, Budapest</td>
</tr>
<tr>
<td>Baroni P.</td>
<td>LLB, Saclay</td>
</tr>
<tr>
<td>Papoular R.</td>
<td>LLB, Saclay</td>
</tr>
<tr>
<td>Islamov A</td>
<td>FLNP, JINR, Dubna</td>
</tr>
<tr>
<td>Kiselev M</td>
<td>FLNP, JINR, Dubna</td>
</tr>
<tr>
<td>Kuklin A</td>
<td>FLNP, JINR, Dubna</td>
</tr>
<tr>
<td>Apel P</td>
<td>FLNR, JINR, Dubna</td>
</tr>
<tr>
<td>Trautmann C</td>
<td>GSI Darmstadt</td>
</tr>
</tbody>
</table>
Références

Data treatment

“Robust” Fit
Huber P 1981 *Robust statistics* (New York: Wiley)
Fox J 2002 *An R and S-PLUS Appendix Companion to Applied Regression*

HWXY software description