

There are many programs available for analysis and presentation of data. I recommend using Genplot, which can be downloaded for free from www.genplot.com. It can be used for fitting and other analysis tasks and for making presentation-quality plots. It generally works faster than Mathematica and other general-purpose programs. Online documentation for Genplot is available at <http://kea.princeton.edu/Help/genplot.html>

In this lab you will learn how to use it and explore some data analysis techniques.

Create fake data set:

```
create y=gnoise(x) -from 0 -to 100 -points 1000
```

gnoise is a function that returns a random number with a Gaussian distribution and standard deviation equal to 1.

```
plot
eval @ave(y)
eval @std(y)
eval @std(y)/sqrt(npt)    - this gives the standard deviation of the average
trans hist 0.2            - transforms data into a histogram that gives the
                           number of points in bins with a width of 0.2
plot -hist                - makes a plot of the histogram. See what happens
                           if you first generate a dataset with more points.
```

We can fit the distribution to a Gaussian. Here are the steps needed to fit data:

Go to non-linear fitting mode:

```
nlsfit
```

Define the fitting function with parameters

```
define f(x)=a*exp(-(x-x0)^2/(2*s))
fun f
```

Define initial values for the variables

```
setvar a 100
setvar x0 0
setvar s 1
```

Specify which parameters you want to vary during the fit. You can also specify the partial derivatives of the fitting function with respect to those parameters or let the program calculate them numerically with / command.

```
vary a /
vary x0 /
vary s /
```

If you want to remove one of the parameters from the list being varied use remove command

```
fit                - this executes the fit command
ov -fit -lt 1     -this will overlay the resulting fit over the data using a solid line
```

As you can see the distribution is overall well described by a Gaussian, but there are deviations in each bin from the fit. What do you think is the distribution of the number of counts in each bin of the histogram?

In the fit so far we assumed that the uncertainty is the same for each bin and did not apply any relative weighting in the sum to calculate χ^2 . A more correct procedure is to use weights corresponding to the uncertainty in each data point.

```
weight 1/y      - this sets the weight for each point, so that  $\chi^2 = \sum w(y - f(x))^2$ 
                 $w = 1/\sigma^2 = 1/(\sqrt{y})^2 = 1/y$ 
fit             Now the fit is done using weighting.
```

What is the approximation being made here?

Look at the quantity called reduced χ^2 , which is equal to χ^2/N and should be close to 1. The fit also gives estimates for the uncertainty in each parameter. You can check that this uncertainty is reasonable by generating several random datasets, converting them to histograms and doing the fit.

Data saving and retrieval:

```
write test.dat  -this writes data in a two-column ASCII format
read test.dat   -if you have more columns you can use
read -col 2 3   to read 2 and 3 column as x and y
```

Data averaging and filtering

```
create y=gnoise(x) -from 0 -to 100 -points 1000
arch noise         - saves data in a curve called noise for later use
plot
trans average 10   - averages 10 points into 1
ov
retr noise
trans fft_filter 1/(1+exp(3*(f$-1))) - this applies an low-pass
                                     filter using Fast Fourier Transform.
ov                 - data have less noise
```

The filtering function in frequency space is motivated by the Fermi-Dirac distribution. It gives flat response for frequencies below 1 Hz and exponential attenuation for frequencies above 1 Hz. The width of the transition region is 1/3 Hz.

Fourier Transforms

```
create y=0.5*sin(2*Pi*x)+gnoise(x) -from 0 -to 100 -points 1000
plot                                 - the sine wave is barely visible in the noise
trans fft -magn                      -this gives the fft magnitude transform (sum in quadrature
                                     of real and imaginary components)
plot                                  - now you can see the peak in frequency space.
```

Note that the Fourier transform extends to 5 Hz. This is the result of the Nyquist theorem, for data with a sampling rate of 10 Hz (1000 points over an interval of 100 sec), the frequency range is equal to 10 Hz/2.

Digital lock-in amplifier

```
create y=0.1*sin(2*Pi*x)+gnoise(x) -from 0 -to 100 -points 1000
```

Now the signal is even smaller

```
let y 2*y*sin(2*Pi*x)
```

```
trans fft_filter 1/(1+exp(10*(f$-0.5)))
```

```
eval @ave(y)
```

This number should give the amplitude of the sine wave.

What do you think is the uncertainty in the estimate of this amplitude?

Making nice plots.

You can use menus on top of genplot window to make labels for plots and change their appearance. Plots can be saved to windows metafiles or EPS files.

Putting it to good use: A data analysis challenge will be posted on the class website. You will need to complete it for your report on this lab.