Exercise

The result of the algorithm for smoothing data introduced above depends on the width σ chosen for the kernel. Calculate the smoothed data for a wide range of σ values. If the width of the kernel is less than the distance between the data points, one obtains essentially the original data. If, on the other hand, the width is larger than the entire x-interval, the smoothed data are nearly constant. Determine a value for σ that appears appropriate to you.

Literature

Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P. (1992) Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, Cambridge, New York

Wolfram S. (1996) The Mathematica Book, 3rd ed. Wolfram Media, Champaign, IL, and Cambridge University Press, Cambridge

1.5 Nonlinear Fit

One goal of science is to find regularities in measured data and to express them mathematically. Therefore, model functions with free parameters are frequently fitted to the data. The models are – hopefully – the results of theories; the computer is needed to find the "best" parameter values and provide a measure of the quality of the fit. To this end, comprehensive tools from the mathematical field of statistics, as well as equally comprehensive program packages, are available to the researcher.

If the models are linear in the parameters, the mathematical theory for this is particularly well developed. Frequently, however, the physicists' models are more complex, requiring nonlinear fits. Here, we want to use a simple example to demonstrate the principle of nonlinear parameter search. In doing so, we will estimate the quality of the fit by a χ^2 test, in which we have to set a measure of our confidence ourselves. We will never be able to prove that our model represents reality; the test will, however, allow us to exclude false models with a high degree of probability. For an accepted model, we will also be able to specify a measure for the precision of the parameters.

Using the example of a damped oscillation that is measured at just eleven points in time, yielding very noisy data, we fit the data by a model with four parameters and determine their errors.

Theory

Assume N data pairs $\{Y_i, t_i\}$, to which a model function $g(\boldsymbol{a}, t_i)$ with an unknown M-component parameter vector $\boldsymbol{a}$ is to be fitted. Assume further

that it is known that for each t_i the data Y_i are measured with errors ε_i that vanish when averaged over many experiments and have the variances σ_i^2, i.e., $\langle \varepsilon_i \rangle = 0$ and $\langle \varepsilon_i^2 \rangle = \sigma_i^2$. The "best" set of parameters $\boldsymbol{a}$ is defined to be the vector $\boldsymbol{a}_0$ whose components minimize the quadratic deviation χ^2, where χ^2 is defined as follows:

$$\chi^2(\boldsymbol{a}) = \sum_{i=1}^{N} \left[\frac{Y_i - g(\boldsymbol{a}, t_i)}{\sigma_i} \right]^2 . \tag{1.28}$$

The distribution of $\chi^2(\boldsymbol{a}_0)$ is known if $g(\boldsymbol{a}_0, t_i)$ is a suitable model and the errors $\varepsilon_i = Y_i - g(\boldsymbol{a}_0, t_i)$ are uncorrelated and follow a Gaussian distribution. One can verify by repeating the experiment many times that the probability for χ_0^2 is given by

$$P_{N-M}\left(\chi_0^2\right) = \frac{1}{\Gamma\left(\frac{N-M}{2}\right)} \int_0^{\chi_0^2/2} \mathrm{e}^{-t} t^{\frac{N-M}{2}-1} \mathrm{d}t . \tag{1.29}$$

The integral is known in mathematical literature as an incomplete gamma function $\gamma((N-M)/2, \chi_0^2/2)$. $\chi^2(\boldsymbol{a}_0)$ only has $N-M$ independent variables, since M degrees of freedom are fixed by the minimum condition. For large values of $N-M$, the central limit theorem applies: χ^2 is distributed according to a Gaussian with mean $\langle \chi^2 \rangle = N - M$ and variance $2(N-M)$.

Now what is the meaning of the χ^2 distribution (1.29) for the precision of our fit? Let us assume that our experiment yields the value $\chi_0^2 = \chi^2(\boldsymbol{a}_0)$ for the minimum of (1.28), with a value $P(\chi_0^2) = 0.99$. This means that if the model is correct the value of χ^2 would be less than χ_0^2 in 99% of all experiments. We could still accept that as a valid fit, since we cannot exclude the possibility that our experiment belongs to those 1% of all cases for which χ^2 is greater than or equal to χ_0^2. If, however, we had $P = 0.9999$, our fit would only be correct if it belonged to the 0.01% of the experiments for which $\chi^2 \geq \chi_0^2$. In this case we would certainly conclude that our assumptions, and our model in particular, are not correct. On the other hand, the value of χ_0^2 must not be too small, since our data points do have statistical errors. The probability for an experiment to yield a smaller value than our χ_0^2 is again given by the distribution (1.29). The interval for the value of $P(\chi_0^2)$ that we want to accept is called the *confidence interval.* Where we set the limit, at 1% or at 0.01%, depends on ourselves and on our experience with similar problems.

Let us assume now that we are satisfied with the quality of our fit, i.e., $\chi^2(\boldsymbol{a})$ has a minimum for some parameters $\boldsymbol{a}_0$ and the value of $\chi^2(\boldsymbol{a}_0)$ falls within our confidence interval. If we could repeat the experiment several times, we would of course obtain different errors ε_i and consequently a different parameter vector $\boldsymbol{a}_0$. We perform just one experiment, however, but want to use the data to estimate the error of $\boldsymbol{a}_0$ anyway.

Here, the χ^2 distribution (1.29) helps us again, as contours in M-dimensional $\boldsymbol{a}$-space with constant values of $\chi^2(\boldsymbol{a})$ are a measure of the error of $\boldsymbol{a}_0$. For small deviations $|\boldsymbol{a} - \boldsymbol{a}_0|$, these contours are ellipsoids with M principal axes. Their lengths are a measure of the change one can allow for $\boldsymbol{a}_0$ along these axes before the fit is no longer acceptable.

One can even quantify this statement. To this end, we generate artificial data Y_i by adding errors ε_i that are distributed according to a Gaussian with $\langle \varepsilon_i^2 \rangle = \sigma_i^2$ to the model $g(\boldsymbol{a}_0, t_i)$: $Y_i = g(\boldsymbol{a}_0, t_i) + \varepsilon_i$. With these data, we run the fit procedure again, i.e., we search for a new minimum $\boldsymbol{a}_1$, using the simulated data. If repeated several times, this yields the parameter vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots$. From the width of the distribution of each component of the $\boldsymbol{a}_k$ we obtain the error bars for the fit parameters $\boldsymbol{a}_0$.

If the deviation $|\boldsymbol{a} - \boldsymbol{a}_0|$ is small enough to permit truncating the expansion of $\chi^2(\boldsymbol{a})$ (from the experimental data) about $\boldsymbol{a}_0$ after the quadratic term, it can be shown that the quantity $\Delta = \chi^2(\boldsymbol{a}) - \chi^2(\boldsymbol{a}_0)$ is again distributed according to the distribution function P from (1.29), this time with M rather than $N - M$ degrees of freedom. Therefore, if we require as before that our fit belong to those 99% of all possible fits closest to the correct model, then the inequality

$$P_M \left[\chi^2(\boldsymbol{a}) - \chi^2(\boldsymbol{a}_0)\right] \leq 0.99 \tag{1.30}$$

determines the region of allowed values of $\boldsymbol{a}$. In parameter space, the regions of constant Δ are ellipsoids. The projection of this $(M-1)$-dimensional surface onto the axis i then yields the error interval for the parameter a_i.

Algorithm

For the nonlinear fit, too, it is easiest to use the functions available in *Mathematica*. To find the minimum of $\chi^2(\boldsymbol{a})$, we use the function `NonlinearRegress` from the package `Statistics`NonlinearFit``. It offers various possibilities for entering data and initial conditions; in addition, one can change the method of the minimum search and have the program display intermediate results. Of course, one can also provide one's own definition of $\chi^2(\boldsymbol{a})$ and use `FindMinimum` to find the value $\boldsymbol{a}_0$.

To generate the data, and for our model, we choose a damped sinusoidal oscillation

```
f[t_] := a Sin[om t + phi] Exp[-b t]
```

with four parameters $\boldsymbol{a}$ = `{a, om, phi, b}`. This oscillation is measured at 11 points in time t_i for the parameter set $\boldsymbol{a}$ = `{1,1,0,0.1}`, and noise in the form of uniformly distributed random numbers is added to the data:

```
data =
Table[{t, Sin[t]Exp[-t/10.] + 0.4*Random[] - 0.2}//N,
                                    {t, 0, 3Pi, 0.3Pi}]
```

For the errors ε_i (which are not distributed according to a Gaussian) we have

$$\langle \varepsilon_i \rangle = 0, \quad \sigma_i^2 = \langle \varepsilon_i^2 \rangle = \frac{1}{0.4} \int_{-0.2}^{0.2} x^2 \, \mathrm{d}x = \frac{2}{150} .$$

The search for a minimum is facilitated if we can provide an approximate set of values for $\boldsymbol{a}_0$. After loading the statistics packages needed via

```
Needs["Statistics`Master`"]
```

we call the minimum search routine:

```
NonlinearRegress[data,f[t],t,
                 {{a,1.1},{om,1.1},{phi,.1},{b,.2}},
                 ShowProgress->True]
```

The χ^2 distribution is available in *Mathematica* as well, in the package `Statistics`ContinuousDistributions`. It has the self-explanatory name `ChiSquareDistribution[...]`. For the argument, the number of degrees of freedom has to be provided, which in our case, with $N = 11$ data points and $M = 4$ parameters is given by $N - M = 7$. By using `PDF` one obtains the density distribution, i.e., the integrand of $P_7(\chi_0^2)$, (1.29), whereas `CDF` yields the integral, i.e., $P_7(\chi_0^2)$ itself. With `Quantile`, the function $P_7(\chi_0^2)$ is inverted. Thus,

```
Quantile[ChiSquareDistribution[7], 0.95]
```

yields the value of χ_0^2 that is defined by the condition that 95% of all experiments will result in a χ^2 value smaller than χ_0^2. If two parameters are kept fixed, one can use `ContourPlot` to display, for the remaining two parameters, the area in parameter space for which $\chi^2(\boldsymbol{a}) = \chi^2(\boldsymbol{a}_0) + \Delta$.

Results

Figure 1.8 shows the function `f[t]` for `a=1`, `om=1`, `phi=0`, and `b=0.1`, as well as the 11 data points that were obtained from `f[t]` with the random errors ε_i. Starting from `a=1.1`, `om=1.1`, `phi=0.1`, and `b=0.2`, `NonlinearRegress` finds the minimum $\boldsymbol{a}_0$ of $\chi^2(\boldsymbol{a})$ in about eight steps. In order to extract the best fit parameters from the result, we have to enter the command `BestFitParameters /. NonlinearRegress[...]`, yielding

```
{a -> 0.824, om -> 0.976, phi -> 0.024, b -> 0.068}        (1.31)
```

The χ^2 distribution is shown in Fig. 1.9. Strictly speaking, the distribution (1.29) is valid only for Gaussian-distributed errors ε_i, but we do not expect a large difference for our uniformly distributed errors ε_i. For the correct set of parameters, this density function yields the distribution of χ_0^2 for different experiments, i.e., for different realizations of the ε_i.

We obtain the confidence interval for the value of χ_0^2 from

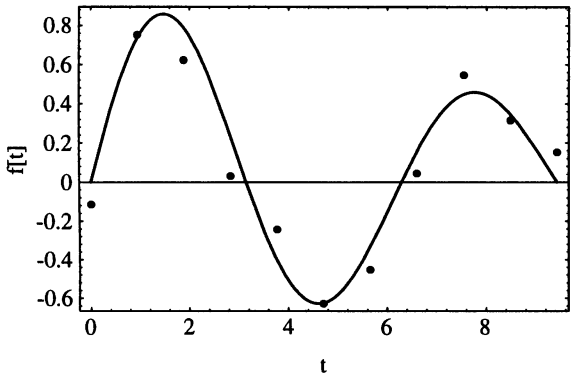

Fig. 1.8. Damped oscillation and noisy data

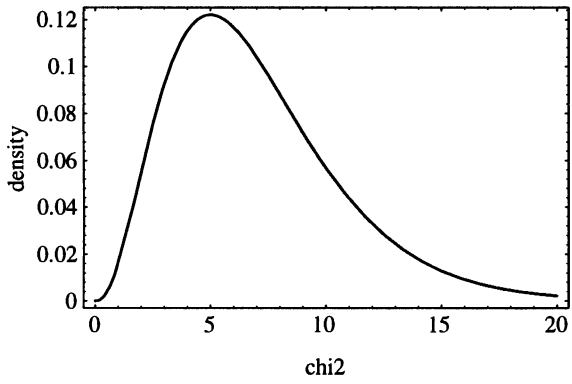

Fig. 1.9. χ^2 density distribution with seven degrees of freedom

```
limit[x_] = Quantile[ChiSquareDistribution[7],x]
```

and the result of `{limit[0.05], limit[0.95]}` is $\{2.2, 14.1\}$. This means that, given a large number of experiments (with the correct parameters), 5% of them should yield a value $\chi_0^2 \geq 14.1$ and 5% a value $\chi_0^2 \leq 2.2$. In our case, we obtain $\chi_0^2 = 9.4$, well inside the confidence interval; therefore, we have no reason to doubt the result of our fit.

Contours for $P_4(\chi^2(\boldsymbol{a}) - \chi_0^2) = 0.68$ and $P_4(\chi^2(\boldsymbol{a}) - \chi_0^2) = 0.90$ can be seen in Fig. 1.10. These are three-dimensional surfaces in four-dimensional parameter space, which is why we can only plot slices. Of course, the optimal parameter set is situated exactly in the center of all these contour surfaces. The `(a,b)` slice shows that variations of the amplitude `a` can be compensated for by a change in $\mathtt{b}^{-1}$, the time constant of the damping, without reducing the quality of the fit. Therefore, one cannot simply specify the cut along the optimum value of `b` for the error in `a`, but has to use the projection onto the `a`-axis instead. In particular, the left-hand panel of Fig. 1.10 shows that it is not possible to determine an upper limit for the time constant $\mathtt{b}^{-1}$ since the contour of the outer confidence region passes through `b=0` (i.e., $\mathtt{b}^{-1} = \infty$).

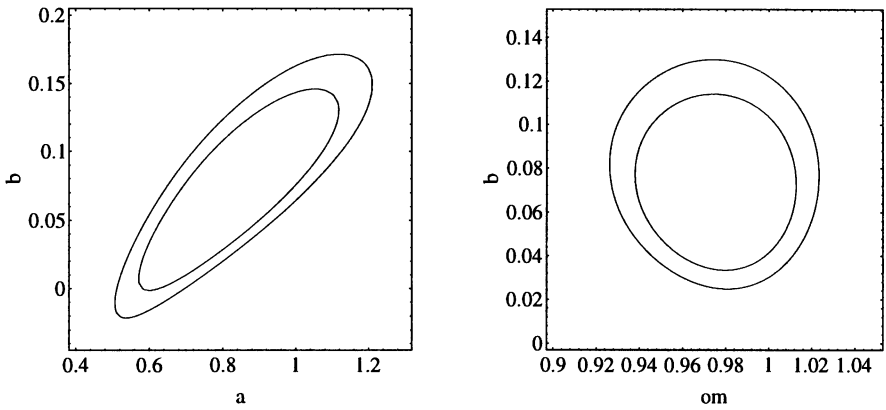

Fig. 1.10. Contours of constant $\chi^2(\boldsymbol{a})$. Intersection with the `a-b` plane in parameter space (*left*) and with the `om-b` plane (*right*)

If one varies the frequency `om` rather than the amplitude `a` the compensation discussed above does not happen, as shown in the right-hand panel of Fig. 1.10. The true values, `om=1` and `b=0.1`, lie close to the edge of the inner confidence region; therefore, the confidence interval should not be too narrow. If one plots `f[t]` for two extreme values from Fig. 1.10, namely `a=1.3`, `b=0.15` and `a=0.5`, `b=0`, one can see in Fig. 1.11 that both curves still represent the data relatively well. The small number of data ($N = 11$) and the large error do not permit a better fit.

As discussed in the theory section, there is an alternative method for estimating the errors of the parameters. One takes the optimum parameter set $\boldsymbol{a}_0$ for one experiment and uses it to generate new, artificial data that are fitted in turn. Repeating this for the same $\boldsymbol{a}_0$ yields a set of parameter vectors $\boldsymbol{a}_k$. Figure 1.12 shows the results of 100 iterations, using the $\boldsymbol{a}_0$ from (1.31), together with the contours from Fig. 1.10. It can be seen that in the (`a`,`b`) projection even more than 90% of the data fall within the outer contour

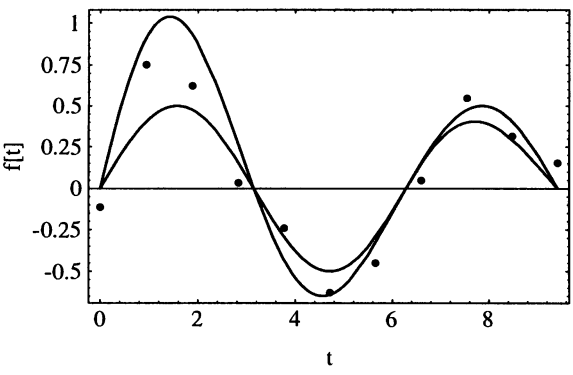

Fig. 1.11. Two fit curves generated from parameters from the outer contour of the `a-b` plot shown in Fig. 1.10

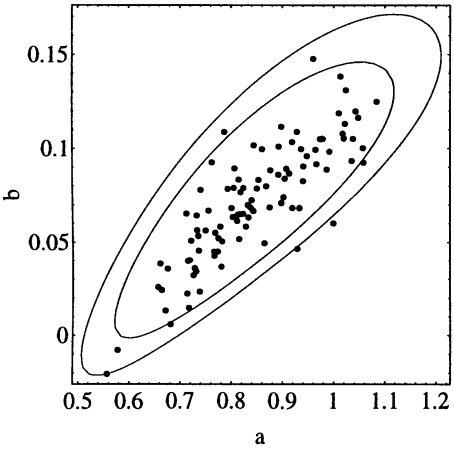

Fig. 1.12. Parameter values a and b from the fits of the artificial data are shown together with the contours from the left-hand plot of Fig. 1.10

of the slice. In order to specify error limits for $\boldsymbol{a}_0$, we calculate $\chi^2(\boldsymbol{a}_k)$ for each of the 100 parameter vectors $\boldsymbol{a}_k$ and the original data set $\{Y_i\}$ and sort the $\boldsymbol{a}_k$ by increasing χ^2. For a confidence level of 68%, for example, we then take into account only the first 68 of these vectors, the last of which sets a χ^2 limit $\chi^2_{68} = \chi^2(\boldsymbol{a}_{68})$. All these remaining $\boldsymbol{a}_k$ then lie inside the quasi-ellipsoid $\chi^2(\boldsymbol{a}) \leq \chi^2_{68}$. We project the points $\boldsymbol{a}_k$ onto the axes of parameter space to obtain the extreme dimensions of the ellipsoid. In this way, we can eventually make the statement that with 68% certainty the true vector $\boldsymbol{a}_{\text{true}}$ lies in a quasi-ellipsoid that is contained in the rectangular parallelepiped

$$\texttt{a} = 0.82 \pm 0.26\,, \quad \texttt{b} = 0.07 \pm 0.06\,, \quad \texttt{om} = 0.98 \pm 0.05\,, \quad \texttt{phi} = 0 \pm 0.2\,.$$

Of course, not every point $\boldsymbol{a}$ of the rectangular parallelepiped belongs to the 68% region, owing to the correlations mentioned above. Whether or not it belongs to this region can be determined from its $\chi^2(\boldsymbol{a})$.

Exercise

The data $\{x_i, f(x_i)\}$ in the file `twinpeak.dat` are the result of the function

$$f(x_i) = \exp\left(-\frac{x_i^2}{2}\right) + a \exp\left[-\frac{(x_i - b)^2}{2\sigma^2}\right] + r_i\,,$$

where the r_i are normally distributed random numbers with mean 0 and width $\sigma_0 = 0.05$. You can load the data during a *Mathematica* session via

```
data = << twinpeak.dat
```

and display them with `ListPlot[data]`. Find the amplitude a, the position b, and the width σ of the additional peak and specify the errors of the three fit parameters.

Literature

Bevington P.R., Robinson D.K. (1992) Data Reduction and Error Analysis for the Physical Sciences. McGraw-Hill, New York

Honerkamp J. (1994) Stochastic Dynamical Systems: Concepts, Numerical Methods, Data Analysis. VCH, Weinheim, New York

Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P. (1992) Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, Cambridge, New York

Wolfram S. (1996) The Mathematica Book, 3rd ed. Wolfram Media, Champaign, IL, and Cambridge University Press, Cambridge

1.6 Multipole Expansion

The expansion of physics equations in terms of a small quantity is an important tool of the theoretical physicist. In this process, expressions may result that are complicated and nonintuitive. Frequently, one has no feel for the expansion's deviation from the true value.

The multipole expansion of an electrostatic potential, a favorite chapter from the course on electrodynamics, is a simple example of this. From far away, a charge distribution looks like a point charge. As the observer gets closer, the dipole moment will become noticeable. On even closer approach the quadrupole moment will be noticeable as well. This approximate description can be put into a compact mathematical form using a scalar, a vector, and a tensor. Now – with *Mathematica* – we can program this just as compactly, but also display it graphically and investigate the deviation from the exact potential.

Physics

The electrostatic potential $\Phi(\boldsymbol{r})$ of N pointlike charges e_i at the positions $\boldsymbol{r}^{(i)}$ is given by

$$\Phi(\boldsymbol{r}) = \sum_{i=1}^{N} \frac{e_i}{\left|\boldsymbol{r} - \boldsymbol{r}^{(i)}\right|} \,. \tag{1.32}$$

The electric field is determined by the gradient of Φ:

$$\boldsymbol{E} = -\boldsymbol{\nabla}\Phi(\boldsymbol{r}) \;, \boldsymbol{\nabla}\Phi = \left(\frac{\partial\Phi}{\partial x}, \frac{\partial\Phi}{\partial y}, \frac{\partial\Phi}{\partial z}\right) \,. \tag{1.33}$$

When observing this potential from a long distance, i.e., for $|\boldsymbol{r} - \boldsymbol{r}^{(i)}| \to \infty$, one can expand $\Phi(\boldsymbol{r})$:

$$\Phi(\boldsymbol{r}) = \frac{q}{r} + \frac{\boldsymbol{p}\cdot\boldsymbol{r}}{r^3} + \frac{1}{2r^5}\boldsymbol{r}\mathrm{Q}\boldsymbol{r} + \mathcal{O}\left(\frac{1}{r^4}\right) \,, \tag{1.34}$$