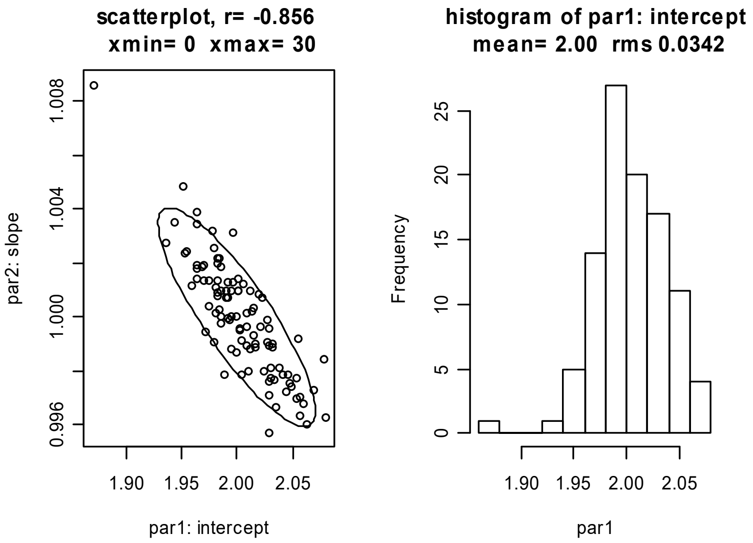
Parameter Estimation Applied to Medical and Biological Sciences

February - March 2022

Dr. Ivo H.M. van Stokkum

Faculty of Science

vrije Universiteit *amsterdam*



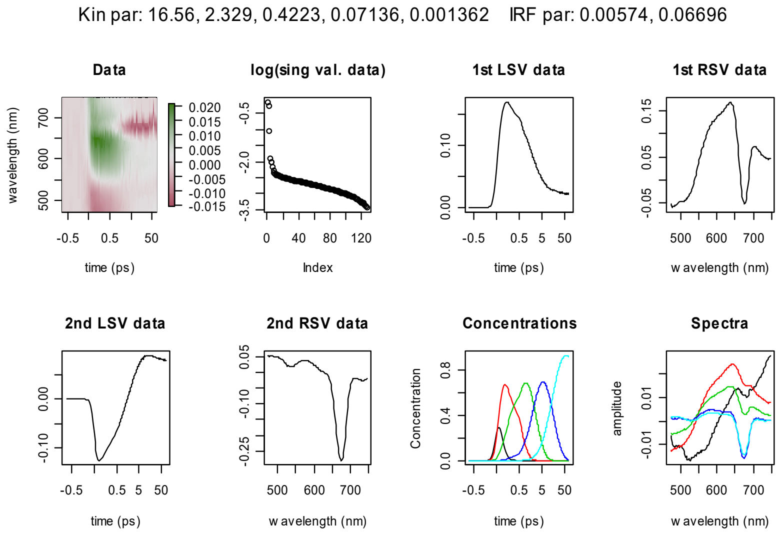


Table of contents

2022

2022

Occam's razor

Occam's razor (also written as Ockham's razor from William of Ockham, and in Latin *lex parsimoniae*) is a principle of parsimony, economy, or succinctness used in logic and problem-solving. It states that among competing hypotheses, the hypothesis with the fewest assumptions should be selected (from wikipedia). When modelling unknown data the correct model is often unknown, but a suitable class of models is available. When two models describe the data equally well, the principle of parsimony guides us to adopt the most simple model, which is in general the model with the least number of parameters.

Interpretation of a figure

“A picture is worth a thousand words” (“Een plaatje zegt soms meer dan duizend woorden”) is a well known proverb. In scientific publications pictures, graphs, plots or figures convey part of the message. When interpreting a figure, answer some of the following questions:
What quantities vary along the axes, and what are their units?
Are the axes linear, logarithmic (NB remember that ${}^{10}\log(2)\cong 0.3$ and ${}^{10}\log(3)\cong 0.5$ etc.) or …?
Are these raw data, model computations, mathematical functions, or a combination thereof?
How well does a fit approximate the raw data?
What can be said about residuals or measurement errors?
When parameters are estimated, can you discern them in the picture (slope, intercept, decay rate or lifetime, amplitude, location and widths of bands in spectra, ...)
What relations are present (linear, exponential, logarithmic, power, square root, ...)?
Are special points present (minima, maxima, cusp, hole, ...) ?
What is the message of the figure?
After answering these questions:
Write a caption, and give a title.

Writing a caption is an important academic skill, which will be judged in your report or practical exam.

Functional relations

Interpretation of an x-y plot requires judging which functional relations are present (linear, exponential, logarithmic, power, square root, ...)?

Connected to this is the type of axis (linear, logarithmic). The table below summarizes and indicates when a logarithmic axis simplifies the interpretation of an x-y plot.

NB a requirement for a logarithmic axis is positivity of all arguments.

Overview of common functional relations

relation	formula	natural logarithm ($\log \equiv \ln$)	axes	slope	intercept
proportional	y = a x		linear-linear	a	0
	y = a x	log(y)=log(a)+log(x)	log-log	1	log(a)
reciprocal	$y = \frac{a}{x}$	log(y)=log(a)-log(x)	log-log	-1	log(a)
linear	y = a x+b		linear-linear	a	b
exponential	y = b exp(ax)	log(y) = ax+log(b)	linear-log	a	log(b)
	$y = 2^x$	$\log(y) = x \cdot \log(2)$	linear-log	log(2)	0
	$y = 10^x$	$\log(y) = x \cdot \log(10)$	linear-log	log(10)	0
logarithmic	$y = a \cdot \log(x)$		log-linear	a	0
power	$y = b \cdot x^a$	$\log(y) = a \cdot \log(x) + \log(b)$	log-log	a	log(b)
	$y = x^a$	$\log(y) = a \cdot \log(x)$	log-log	a	0
square root	$y = \sqrt[a]{x}$	$\log(y) = (\log(x))/a$	log-log	1/a	0
quadratic polynomial	$y = ax^2 + bx + c$		linear-linear		

Note that a linear relation in a log-log x-y plot requires careful inspection of the slope to judge the relation between x and y.

Many programs (Excel, Matlab, Origin, ...) use logarithm base 10 (*log10)* instead of the natural logarithm *ln*. Then the slope of the linear-log plot (where a trend line has been fitted) must be divided by $\log(10)$.

Parameter estimation crash course

1.1. Introduction

Parameter estimation plays an important role in all kinds of sciences, in particular when data need to be described with the help of a model. This introduction deals with how to estimate the parameters of a mathematical model that describes measurements. In particular we will introduce least squares estimation techniques. After this introduction you should be able to explain the principles of parameter estimation, write down a simple mathematical model for observations with additive noise, pinpoint the parameters to be estimated, distinguish between linear and nonlinear parameters, and judge the model usefulness from the residual structure.

1.2. A simple example: estimation of the mean

The principles of parameter estimation will be illustrated by describing a simple problem: estimation of the mean of a population from a number of observations (samples) from that population. At first glance the method seems obvious: averaging. However, there are some assumptions concerning the nature of the observations which need to be fulfilled, otherwise averaging can be a bad method. This brings us to the need for a model based description of the observations.

Let y_i be the *i*-th observation, and call the (population) mean of the observations μ, which is an unknown parameter. Suppose the observations contain additive noise, and suppose that all observations are statistically independent and that all observations possess the same variance σ^2, where σ is the standard deviation of the noise. The mathematical model now reads:

$$\underline{y}_i = \mu + \underline{\nu}_i \qquad \text{Eq.1.1}$$

where $\underline{\nu}_i$ is the noise added to the *i*-th observation. The underlining indicates that ν is a stochastic variable. Because ν is stochastic the *i*-th observation $\underline{y}_i$ is also stochastic. A number of *n* repetitions of the experiment will result in an ensemble of observations $\underline{y}_1, \underline{y}_2, \ldots, \underline{y}_i, \ldots, \underline{y}_{n-1}, \underline{y}_n$. We assume next that the noise ν is drawn from a normal distribution written as $\nu \sim N(0, \sigma^2)$ (which means ν is normally distributed with zero mean ($E[\underline{\nu}_i] = 0$) and variance $\mathrm{var}(\underline{\nu}_i) = E[(\underline{\nu}_i - E[\underline{\nu}_i])^2] = \sigma^2$). Now we can write down the probability density function (which is explained in Fig.1.) for the *i*-th observation $\underline{y}_i$:

$$f(\underline{\nu}_i) = \frac{e^{-\underline{\nu}_i^2/(2\sigma^2)}}{\sqrt{2\pi\sigma^2}} \qquad f(\underline{y}_i|\mu) = \frac{e^{-(\underline{y}_i - \mu)^2/(2\sigma^2)}}{\sqrt{2\pi\sigma^2}} \qquad \text{Eq.1.2}$$

Because all observations are independent (and identically distributed) we get for the joint

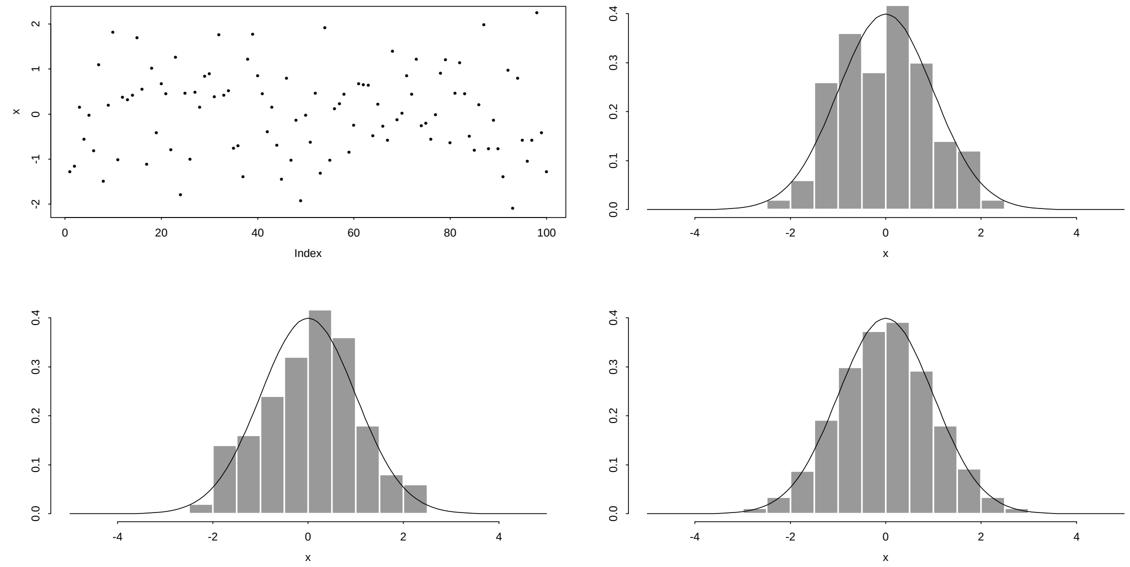


Fig.1. Upper left: hundred samples drawn from a standard normal distribution $N(0, 1)$. Upper right: accompanying histogram, compared to normal probability density function (solid line). Lower left: Histogram from next hundred samples, note that it differs from the previous. Lower right: Histogram from ten thousand samples, note that it approaches the normal probability density function more closely.

probability density function the product of all *n* terms from Eq.1.2, which is called the **likelihood function**. The natural logarithm of this product contains the sum of the arguments of the exponential terms, and is called the log likelihood function:

$$f(\underline{y}_1, \underline{y}_2, \ldots, \underline{y}_i, \ldots, \underline{y}_{n-1}, \underline{y}_n | \mu) = \prod_{i=1}^{n} (2\pi\sigma^2)^{-1/2} \exp\left(-\frac{(\underline{y}_i - \mu)^2}{2\sigma^2}\right)$$

$$\log(f(\underline{y}_1, \underline{y}_2, \ldots, \underline{y}_i, \ldots, \underline{y}_{n-1}, \underline{y}_n | \mu)) = -\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n} \frac{(\underline{y}_i - \mu)^2}{2\sigma^2} \qquad \text{Eq.1.3}$$

The second term on the right hand side of Eq.1.3,

$$\sum_{i=1}^{n} \frac{(\underline{y}_i - \mu)^2}{2\sigma^2} \qquad \text{Eq.1.4}$$

contains the sum of the squares of the deviations of the individual observations from the mean μ. We now estimate μ by maximizing the (logarithm of) the likelihood function from Eq.1.3 with respect to the unknown parameter μ, which is called maximum likelihood estimation. The $\underline{\hat{\mu}}$ thus estimated (estimation is indicated by a circumflex) is the parameter which makes the observations most likely. Maximizing Eq.1.3 is equivalent to minimizing Eq.1.4, which is called least squares estimation. The two estimation methods are equivalent when all observations are independent and (identically) normally distributed.

To minimize Eq.1.4 we note that it contains only positive contributions, thus it is sufficient that the derivative with respect to the parameter μ is zero:

$$\frac{\partial}{\partial\mu}\left(\sum_{i=1}^{n}\frac{(\underline{y}_i-\mu)^2}{2\sigma^2}\right) = -\sum_{i=1}^{n}\frac{(\underline{y}_i-\mu)}{\sigma^2} = 0 \qquad \text{Eq.1.5}$$

from which we derive the least squares estimator (where the circumflex ^ indicates *estimator*)

$$\hat{\underline{\mu}} = \frac{1}{n}\sum_{i=1}^{n}\underline{y}_i \qquad \text{Eq.1.6}$$

which is exactly the *average* of all observations ! Because the observations are stochastic, the estimator $\hat{\underline{\mu}}$ is also stochastic. Using the property that variances of independent variables are additive ($\mathrm{var}(\sum_i(\underline{y}_i)) = \sum_i \mathrm{var}(\underline{y}_i)$) we can calculate the variance of the estimator $\hat{\underline{\mu}}$:

$$\mathrm{var}(\hat{\underline{\mu}}) = \mathrm{var}\left(\frac{1}{n}\sum_{i=1}^{n}(\underline{y}_i)\right) = \left(\frac{1}{n}\right)^2\sum_{i=1}^{n}\mathrm{var}(\underline{y}_i) = \left(\frac{1}{n}\right)^2(n\sigma^2) = \frac{\sigma^2}{n} \qquad \text{Eq.1.7}$$

where we have also used the variance property $\mathrm{var}(a\underline{y}_i) = a^2\mathrm{var}(\underline{y}_i)$. Thus what we have achieved by the parameter estimation is a *variance reduction*: we started with *n* observations, which were assumed to be normally distributed with mean μ and variance σ^2. The least squares estimate summarizes these *n* observations by *one* estimate $\hat{\underline{\mu}}$ for the parameter μ which possesses an *n* times smaller variance.

1.3. Weighting the observations

When the variance of the observations is no longer constant, but each observation possesses a standard error σ_i, then we arrive at the weighted sum of the squares of the residuals

$$\sum_{i=1}^{n}\frac{(\underline{y}_i-\mu)^2}{2\sigma_i^2} \qquad \text{Eq.1.8}$$

To minimize Eq.1.8 it is again sufficient that the derivative with respect to the parameter μ is zero:

$$\frac{\partial}{\partial\mu}\left(\sum_{i=1}^{n}\frac{(\underline{y}_i-\mu)^2}{2\sigma_i^2}\right) = -\sum_{i=1}^{n}\frac{(\underline{y}_i-\mu)}{\sigma_i^2} = -\sum_{i=1}^{n}\frac{\underline{y}_i}{\sigma_i^2}+\mu\sum_{i=1}^{n}\frac{1}{\sigma_i^2} = 0 \qquad \text{Eq.1.9}$$

from which we derive the *weighted* least squares estimator

$$\hat{\underline{\mu}} = \sum_{i=1}^{n}\frac{\underline{y}_i}{\sigma_i^2}\Big/\sum_{i=1}^{n}\frac{1}{\sigma_i^2} \qquad \text{Eq.1.10}$$

which is the *weighted average* of all observations.

1.4. Linear and nonlinear parameters

We now extend our mathematical model for the observations by replacing in Eq.1.1 the mean μ by a model function $g(x_i|\theta)$ of known independent variables x_i and vector of unknown parameter(s) θ. Commonly the Greek letter θ (theta) is used for the unknown parameters of a nonlinear model. Elements of the vector θ are numbered θ_j where j can start from 0 or 1. With linear models θ_0 represents a constant. The mathematical model for observation $\underline{y}_i$ now reads:

$$\underline{y}_i = g(x_i|\theta)+\underline{\nu}_i \qquad \text{Eq.1.11}$$

Without repeating the whole derivation from the previous section we give here the least squares criterion (replacing μ in Eq.1.4 by $g(x_i|\theta)$):

$$\sum_{i=1}^{n}\frac{(\underline{y}_i-g(x_i|\theta))^2}{2\sigma^2} \qquad \text{Eq.1.12}$$

We now have to minimize Eq.1.12 with respect to the parameter(s) θ.

A model is linear when the derivatives of the model function $g(x_i|\theta)$ with respect to the parameters θ do not depend on any parameter, otherwise the model is nonlinear. A few examples may illustrate this. Find out whether a model is linear or nonlinear by computing all the partial derivatives $\frac{\partial}{\partial\theta_j}g(x_i|\theta)$ for all parameters θ_j:

$$g(x_i|\theta) = \theta_0+\theta_1 x_i \qquad \text{Eq.1.13}$$

$$g(x_i|\theta) = \theta_0+\theta_1 x_i+\theta_2 x_i^2 \qquad \text{Eq.1.14}$$

$$g(x_i|\theta) = \theta_0+\theta_1 x_i+\sin(\theta_2 x_i) \qquad \text{Eq.1.15}$$

$$g(x_i|\theta) = \theta_0+\theta_1 x_i+\theta_2\sin(x_i) \qquad \text{Eq.1.16}$$

$$g(x_i|\theta) = \theta_2 \exp(-\theta_1 x_i) \qquad \text{Eq.1.17}$$

$$g(x_i|\theta) = \theta_3 \exp(-\theta_1 x_i) + \theta_4 \exp(-\theta_2 x_i) \qquad \text{Eq.1.18}$$

With linear models a closed form solution like Eq.1.6 exists, otherwise minimization techniques are necessary. The model function of Eq.1.17 contains an exponential decay, and when the independent variable x_i represents time, it is appropriate to describe e.g. radioactive decay or first order reaction kinetics. In that case the parameter in the exponent (θ_1 in Eq.1.17) is called decay rate constant, and its reciprocal $1/\theta_1$ is called lifetime. The amplitude parameter θ_2 in Eq.1.17 is called a *conditionally linear* parameter (under the condition that θ_1 is known).

1.5. Case study

We present here an example from biophysics. A measured decay trace was fitted with Eq.1.17 (Fig.2., top) and with Eq.1.18 (Fig.2.,bottom). To judge the quality of the fit we must first investigate the residuals. In particular we look for trends as a function of the independent time variable x_i. We note that the residuals (inset Fig.2.a) from a fit of the decay trace with a single exponential decay are unsatisfactory because some structure is present in the beginning. When fitting with a sum of two exponential decays (Eq.1.18) the residuals look much more random, compare the inset of Fig.2.b with the upper left of Fig.1. Thus we accept this fit, and now we investigate the estimated parameters and their standard errors. First, the root mean square error *rmse* (which is estimated from the residuals) is reported ($\hat{\sigma} = (539 \pm 16) \times 10^{-3}$ in Fig.2. Actually, this is less than 1% of the maximum of the trace.).

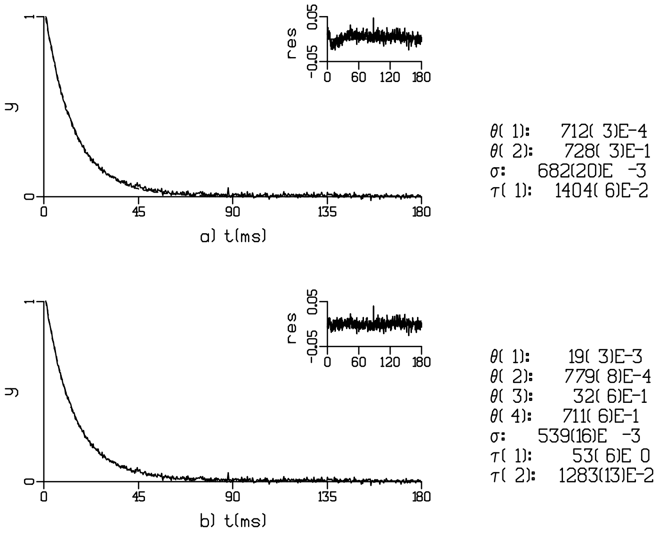


Fig.2. Measured decay trace (solid line) fitted with one (a) or two (b) exponential decays (. The insets show the residuals, the difference between data and model fit which according to the model should be normally distributed (compare with Fig.1. upper left).
Note that the plots are normalized, the maximum of the trace is 74.

In general, the rms value of *n* observations of a quantity ε is defined as

$$\varepsilon_{\mathrm{rms}} = \sqrt{\left(\sum_{i=1}^{n} \varepsilon_i^2\right)/n} \qquad \text{Eq.1.19}$$

To judge whether a parameter is significantly different from zero we compute its *t*-value:

$$t = \theta_i/\sigma_{\theta_i} \qquad \text{Eq.1.20}$$

As a rule of thumb, *t*-values below 2 are considered not significantly different from zero. Here all *t*-values are above 5. Finally, we note that the short lifetime (almost 13 ms, $\tau_2 = 1/\theta_2$) has a huge amplitude (θ_4), much larger than the small amplitude (θ_3) of the long lifetime(53 ms, $\tau_1 = 1/\theta_1$). This rationalizes the rather small *t*-values of the latter component.

1.6. Conclusion

Crucial to all parameter estimation is the model of the observations (Eq.1.1, Eq.1.11). When additive noise is assumed, independent and identically normally distributed, the least squares estimator is the maximum likelihood estimator and thus the best estimator. From the residuals of a fit we can conclude on the usefulness of the model to describe our observations. Only when we consider the residuals of the fit satisfactory, we can investigate whether the parameters are sensible. Parameters with t-values larger than 2 (as a rule of thumb) are significantly different from zero. In some cases we also investigate whether the estimated parameters can be interpreted scientifically (e.g. a negative decay rate usually does not make sense). After all these checks we can accept the model.

1.7. Transformation of the observations

Sometimes the model with additive noise (of equal variance) is not appropriate. Then a transformation of the observations can help to obtain a new model where the noise is more close to the normal distribution. The most used transformation is the logarithm:

$$\underline{z}_i = \log(\underline{y}_i) = \log(g(x_i|\theta)) + \underline{\nu}_i \qquad \text{Eq.1.21}$$

This means that the errors in the original, untransformed, model are proportional to the observations (sometimes also termed multiplicative noise):

$$\underline{y}_i = \exp(\log(g(x_i|\theta)) + \underline{\nu}_i) = \exp(\underline{\nu}_i)(g(x_i|\theta)) \qquad \text{Eq.1.22}$$

An advantage of the logarithmic transformation is that models with a single exponential ($g = \theta_2 \exp(-\theta_1 x_i)$) or power dependence ($h = \theta_2 x_i^{\theta_1}$) become linear ($\log(g) = \log(\theta_2) - \theta_1 x_i$ and $\log(h) = \log(\theta_2) + \theta_1 \log(x_i)$), and thus suitable for parameter estimation by means of linear regression (this is used with Add Trendline in Excel).

1.8. Transformation of the parameters

With nonlinear regression transformation of the parameters can be used to impose constraints on parameters. The most used transformation is again the logarithm. Suppose that it is known that the parameter θ_1 in $g = \theta_2 \exp(-\theta_1 x_i)$ must be non-negative. Then we can estimate the transformed parameter $\phi_1 = \log(\theta_1)$, which is then unconstrained. The new model reads $g = \theta_2 \exp(-\exp(\phi_1) x_i)$. After the estimation of ϕ_1 the desired parameter is computed as $\hat{\theta}_1 = \exp(\hat{\phi}_1)$. Other transformations are the hyperbolic tangent, with which two bounds can be applied. Suppose $a < \theta_2 < b$ then $\left(\frac{b+a}{2}\right) + \left(\frac{b-a}{2}\right)\tanh(\phi_2)$ always satisfies the bounds, with $-\infty < \phi_2 < \infty$.

1.9. Least absolute values and Robustness

The least absolute values (LAV) criterion (replacing the squares in Eq.1.12 by absolute values) is defined as:

$$\sum_{i=1}^{n} \frac{|y_i - g(x_i|\theta)|}{\sigma_i} \qquad \text{Eq.1.23}$$

Theoretically, the LAV estimator is the maximum likelihood estimator for the double exponential (Laplace) distribution

$$\frac{1}{4\sigma} \exp\left(-\frac{|\nu|}{2\sigma}\right) \qquad \text{Eq.1.24}$$

which has heavier tails than the normal distribution. There are several advantages for estimating parameters using LAV: it is less vulnerable to outliers, and less susceptible to deviations from the usually assumed normal distribution of the additive errors. Thus the LAV estimator is more robust against deviations from the model assumptions. However, there are two huge disadvantages: first, the LAV estimator is much more difficult to compute. There is no longer an analytical solution for models with only linear parameters (in contrast to the linear least squares estimator), and thus optimization must always be done numerically. Furthermore, there are no simple algorithms available (in contrast to the nonlinear least squares estimator). Second, in contrast to the least squares estimator, there is no easy way to estimate the standard errors of the parameters. For these practical reasons LAV estimators are rarely used. Only in simple cases, like with the Solver exercises in Excel.

Chapter 2 Parameter estimation

Introduction

The aim of this chapter is to teach students in theory and in practice how to estimate the parameters of a mathematical model that describes physical measurements. Parameter estimation is based upon stochastics, statistics, linear algebra and numerical techniques. The computer plays a central role in the application of mathematical theories. First the data are collected by, or stored in, a computer. Second, through graphical display and simple analysis the data can be visualized. Third, by means of analysis programs, which make use of libraries containing mathematical and statistical routines, the data can be used to estimate the parameters of a mathematical model that is suggested to describe these data. Furthermore the computer can be used to simulate such a model, and these simulated data can be compared with real data. A modern problem solving environment (PSE) called R will be used.
In the appendices the notation convention as well as some definitions and basic results from linear algebra and probability theory can be found.

2.1 Criteria for parameter estimation

This section is devoted to statistical theories of parameter estimation.
Given a mathematical model:

$$\underline{y}_i = g(x_i|\theta) + \underline{\nu}_i \qquad \text{Eq. 2.1}$$

where y and ν are column vectors of length n, representing the stochastic observations and additive noise. θ is a column vector of length k, containing the unknown parameters. Each observation $\underline{y}_i$ depends upon l independent variables, contained in the vector x_i. g is a scalar function of x_i and θ. Given this parameter estimation problem, how can one best estimate θ ? To answer this question one needs to know the probability density function (PDF) of the observations, given the unknown parameters, $f(\underline{y}|\theta)$. Based upon this PDF one can define estimators for θ . Natural demands for an estimator $\hat{\underline{\theta}}$ are unbiasedness ($E[\hat{\underline{\theta}}] = \theta$) and minimal (co)variance. The variance of parameter estimate $\hat{\underline{\theta}}_j$ is defined as: $\mathrm{var}(\hat{\underline{\theta}}_j) = E[(\hat{\underline{\theta}}_j - E[\hat{\underline{\theta}}_j])^2] = E[\hat{\underline{\theta}}_j^2] - (E[\hat{\underline{\theta}}_j])^2$. The covariance of parameter estimates $\hat{\underline{\theta}}_j$ and $\hat{\underline{\theta}}_m$ is defined as:

$$\mathrm{cov}(\hat{\underline{\theta}}_j, \hat{\underline{\theta}}_m) = E[(\hat{\underline{\theta}}_j - E[\hat{\underline{\theta}}_j])(\hat{\underline{\theta}}_m - E[\hat{\underline{\theta}}_m])] \qquad \text{Eq. 2.2}$$

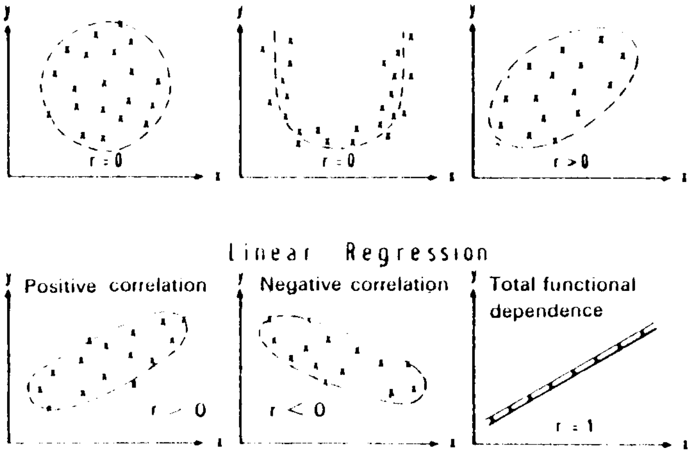
Covariance and correlation coefficient

Correlation is scaled covariance. We estimate the correlation coefficient *r* of two random variables *x* and *y* with

$$\hat{\underline{r}}_{x,y} = \frac{\frac{1}{n}\sum_{i=1}^{n}(\underline{x}_i-\bar{\underline{x}})(\underline{y}_i-\bar{\underline{y}})}{\underline{s}_x\underline{s}_y} \qquad \text{Eq. 2.3}$$

where the mean and the variance are estimated as usual: $\bar{\underline{x}} = \frac{1}{n}\sum_{i=1}^{n}\underline{x}_i$ and $\underline{s}_x^2 = \frac{1}{n}\sum_{i=1}^{n}(\underline{x}_i-\bar{\underline{x}})^2$. Illustrations of correlations are shown in Fig.2.1. The plot in the middle of the top row of Fig.2.1. implies a U-shaped relation. Suppose we have linear model

Fig.2.1.



$$\underline{y}_i = \theta_0 + x_i\theta_1 + \underline{\nu}_i \qquad \text{Eq. 2.4}$$

where *x*,*y* and ν are column vectors of length *n.* We need to estimate two unknown parameters $\theta = (\theta_0, \theta_1)$. With a particular realization of the noise we will arrive at two estimates $\hat{\theta} = (\hat{\theta}_0, \hat{\theta}_1)_1$. Fig.2.2. shows an example.

However, with a different realization of the noise we will arrive at two different estimates $\hat{\theta} = (\hat{\theta}_0, \hat{\theta}_1)_2$, and so on. Now we can produce a scatterplot of all estimates $(\hat{\theta}_0, \hat{\theta}_1)_j$ for the intercept and the slope of the straight line and investigate the properties of these estimates, for an example with 100 estimates see Fig.2.3.(left). The correlation coefficient *r* of the two estimated parameters is computed to be -0.856. This means that a positive deviation from the

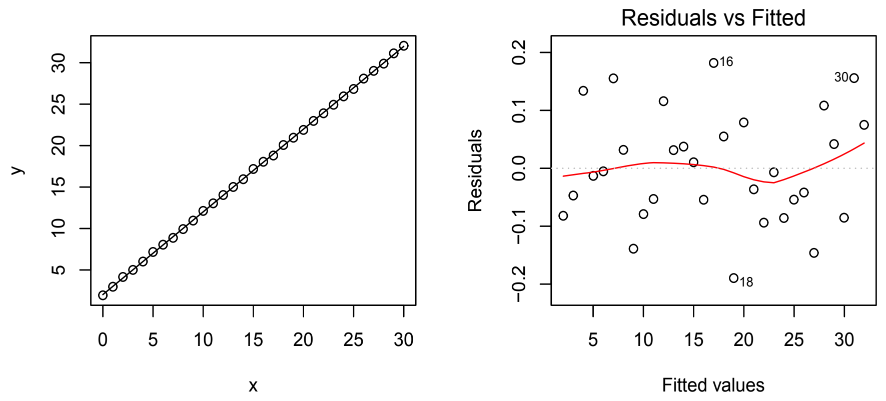


Fig.2.2. Simulation of straight line fit (left) and residuals against Fitted values $\hat{\underline{y}} = \hat{\underline{\theta}}_0 + \hat{\underline{\theta}}_1 x$ (right)

true value of the slope parameter $\hat{\theta}_0 - \theta_{0,\text{ true}}$ will be accompanied by a negative deviation from the true value of the intercept parameter $\hat{\theta}_1 - \theta_{1,\text{ true}}$. The estimates are said to be negatively correlated. Note that on average the estimators are unbiased ($E[\hat{\underline{\theta}}] = \theta$), the mean of the intercept and the slope parameter are equal to the true values of 2 and 1 respectively. The root mean square (rms) deviation from the true value is 0.0342 and 0.00213 respectively.

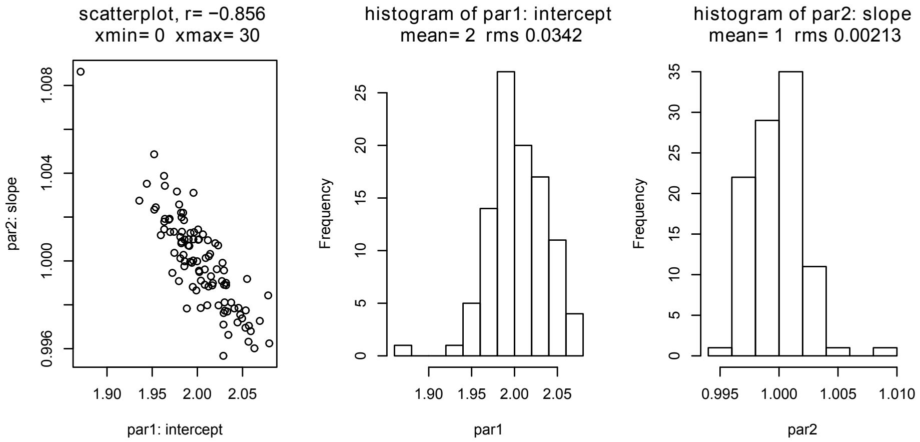


Fig.2.3. Scatterplot of all hundred estimates $(\hat{\theta}_0, \hat{\theta}_1)_j$ for the intercept and the slope of the straight line (left), histogram of intercept estimates (middle) and slope estimates (right).

Fig.2.3. demonstrates that parameter estimates are stochastic, and that the most important properties are the precision (minimal standard error, width of the histogram) of a parameter and the correlation between estimates of two different parameters. For a general model with a vector θ of unknown parameters these are summarized by the autocovariance matrix $D(\theta)$ (see below).

The covariance matrix

The covariance matrix (denoted by D) of two random vectors $\underline{y}$ and $\underline{z}$ is defined by:

$$D(\underline{y}, \underline{z}) = E[(\underline{y} - E[\underline{y}])(\underline{z} - E[\underline{z}])^T] = E[\underline{y}\underline{z}^T] - E[\underline{y}]E[\underline{z}^T] \qquad \text{Eq. 2.5}$$

As a special case, the autocovariance matrix of the parameter vector θ is given by

$$D(\underline{\theta}) \equiv D(\underline{\theta}, \underline{\theta}) = E[(\underline{\theta} - E[\underline{\theta}])(\underline{\theta} - E[\underline{\theta}])^T] = E[\underline{\theta}\underline{\theta}^T] - E[\underline{\theta}]E[\underline{\theta}^T] \qquad \text{Eq. 2.6}$$

The diagonal elements of the autocovariance matrix are the variances $D(\theta)_{ii} = \mathrm{var}(\theta_i)$, whereas the off-diagonal elements are the covariances $D(\theta)_{ij} = \mathrm{cov}(\theta_i, \theta_j)$ (see also Appendix 2.5). Naturally, variances are always non-negative. This can be generalized to the parameter vector by the following theorem.

Theorem 2.1 The autocovariance matrix is positive semidefinite:

$$D(\underline{y}) \equiv D(\underline{y}, \underline{y}) \geq 0 \qquad \text{Eq. 2.7}$$

Proof:

$$\begin{aligned} D(A\underline{y} + b) &= E[(A\underline{y} + b - E[A\underline{y} + b])(A\underline{y} + b - E[A\underline{y} + b])^T] = \\ &E[A(\underline{y} - E[\underline{y}])(\underline{y} - E[\underline{y}])^T A^T] = AD(\underline{y})A^T \end{aligned} \qquad \text{Eq. 2.8}$$

Now if we substitute $A = a^T$ and $b = 0$ we get $D(a^T\underline{y}) = a^T D(\underline{y})a$. Since $a^T\underline{y}$ is a scalar its variance must be non-negative, thus $a^T D(\underline{y})a \geq 0$ which proves Eq. 2.7.□

Eq. 2.8 is also very helpful to calculate the (co)variance of linear combinations of parameters.

Example 2.1. From the fit of simulated data according to Eq. 2.4 an estimate of the autocovariance matrix can be computed:

$$D(\theta) = \begin{bmatrix} \mathrm{var}(\theta_0) & \mathrm{cov}(\theta_0, \theta_1) \\ \mathrm{cov}(\theta_0, \theta_1) & \mathrm{var}(\theta_1) \end{bmatrix} = \begin{bmatrix} (\sigma(\theta_0))^2 & \rho\sigma(\theta_0)\sigma(\theta_1) \\ \rho\sigma(\theta_0)\sigma(\theta_1) & (\sigma(\theta_1))^2 \end{bmatrix} \qquad \text{Eq. 2.9}$$

From the single realization depicted in Fig.2.2. the estimates are collated in Table 2.1. Note

Table 2.1.Estimates from the straight line estimate of Fig.2.2.

Parameter	Estimate $\hat{\theta}_i$	Std. Error $\hat{\sigma}(\hat{\theta}_i)$	t value $\hat{t}_i = \hat{\theta}_i/(\hat{\sigma}(\hat{\theta}_i))$	Pr(>|t|)	$\hat{\rho}(\theta_0, \theta_1)$
Intercept θ_0	2.026	0.034	59.66	<2e-16	-0.86
Slope θ_1	0.998	0.002	513.3	<2e-16	

that the standard error estimates $\hat{\sigma}(\hat{\theta}_i)$ from this single realization are virtually identical to the rms deviations from the true values 0.0342 and 0.00213 of 100 simulations in Fig.2.3. Likewise, the estimated correlation coefficient $\hat{\rho}(\theta_0, \theta_1)$ is virtually identical to the *r* of -0.856 in Fig.2.3. Below we will derive how we can estimate the autocovariance matrix from a single realization of the straight line simulation. □

Example 2.2. Suppose we have estimated the parameter vector $\theta = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix}^T$ with autocovariance matrix $D(\theta) = \begin{bmatrix} \mathrm{var}(\theta_1) & \mathrm{cov}(\theta_1, \theta_2) \\ \mathrm{cov}(\theta_1, \theta_2) & \mathrm{var}(\theta_2) \end{bmatrix}$. When we are interested in the difference $\delta = \theta_1 - \theta_2 = A\theta$ (with $A = \begin{bmatrix} 1 & -1 \end{bmatrix}$) we find using Eq. 2.8:

$$\mathrm{var}(\delta) \equiv D(\delta) = \begin{bmatrix} 1 & -1 \end{bmatrix} D(\theta) \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \mathrm{var}(\theta_1) + \mathrm{var}(\theta_2) - 2\mathrm{cov}(\theta_1, \theta_2) \qquad \text{Eq. 2.10}$$

Note that when θ_1 and θ_2 are strongly correlated ($\mathrm{cov}(\theta_1, \theta_2) \equiv \rho\sigma(\theta_1)\sigma(\theta_2)$ with correlation coefficient ρ nearly 1) then $\sigma(\delta) \approx |\sigma(\theta_1) - \sigma(\theta_2)|$. Thus the precision (remember that a small σ corresponds to a large precision) of the difference of two parameters can be larger than the precision of the individual parameters when their correlation coefficient is nearly 1. □

Now we can use Theorem 2.1 to find a lower bound for $D(\underline{\hat{\theta}})$.

> Theorem 2.2 Let $\underline{\hat{\theta}}$ be any absolutely unbiased estimator of θ based upon a statistic of the observations $\underline{y}$, then the covariance of the error in the estimator is bounded below by the inverse of the Fisher information matrix *M*:

$$D(\underline{\hat{\theta}}) = D(\theta - \underline{\hat{\theta}}) = E_y[(\theta - \underline{\hat{\theta}})(\theta - \underline{\hat{\theta}})^T | \theta] \ge M^{-1} \qquad \text{Eq. 2.11}$$

where

$$M \equiv E_y\left\{\left[\frac{\partial}{\partial\theta}\log f(\underline{y}|\theta)\right]^T\left[\frac{\partial}{\partial\theta}\log f(\underline{y}|\theta)\right]\middle|\theta\right\} = -E_y\left[\frac{\partial^2}{\partial\theta^2}\log f(\underline{y}|\theta)\middle|\theta\right] \qquad \text{Eq. 2.12}$$

Eq. 2.11 is called the Cramér-Rao inequality, and M^{-1} is called the Minimum Variance Bound or the Cramér-Rao lower bound. It quantifies how the stochastic properties of the observations, $f(\underline{y}|\theta)$, determine the stochastic properties of the estimator $\underline{\hat{\theta}}$ computed from those observations. We will not prove Theorem 2.2 here, instead we illustrate the meaning of it with an example.

Example 2.3. Assume the following mathematical model (for a straight line):

$$\underline{y}_i = \theta_0 + x_i\theta_1 + \underline{\nu}_i \qquad \text{Eq. 2.13}$$

The noise samples $\underline{\nu}_i$ are independent and identically distributed (iid). The distribution of the noise is known: the mean, $E[\underline{\nu}_i]$, is zero, and the variance, $E[(\underline{\nu}_i - E[\underline{\nu}_i])^2]$, is equal to σ^2. This distribution, well known as Gaussian white noise, is abbreviated with $N(0, \sigma^2)$. Because the noise samples are iid, the observations are also independent with PDF:

$$f(\underline{y}_i|\theta) = f(\theta_0 + x_i\theta_1 + \underline{\nu}_i|\theta) = (2\pi\sigma^2)^{-1/2}e^{-(\underline{y}_i - \theta_0 - x_i\theta_1)^2/(2\sigma^2)} \qquad \text{Eq. 2.14}$$

For the logarithm of the PDF of $\underline{y}$ we get:

$$\log f(\underline{y}|\theta) = \sum_{i=1}^{n} \log f(\underline{y}_i|\theta) = -\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^{n} \frac{(\underline{y}_i - \theta_0 - x_i\theta_1)^2}{2\sigma^2} \qquad \text{Eq. 2.15}$$

Differentiating with respect to θ gives us:

$$\left(\frac{\partial}{\partial\theta}\log f(\underline{y}|\theta)\right)^T = \sigma^{-2}\sum_{i=1}^{n}\begin{bmatrix} \underline{y}_i - \theta_0 - x_i\theta_1 \\ x_i(\underline{y}_i - \theta_0 - x_i\theta_1) \end{bmatrix} \qquad \text{Eq. 2.16}$$

and

$$\frac{\partial^2}{\partial\theta^2}\log f(\underline{y}|\theta) = -\sigma^{-2}\sum_{i=1}^{n}\begin{bmatrix} 1 & x_i \\ x_i & x_i^2 \end{bmatrix} \qquad \text{Eq. 2.17}$$

Now the expectation in Eq. 2.12 denotes an expectation with respect to y, and since Eq. 2.17 is independent of y, M is a constant matrix. To find the Minimum Variance Bound we only need to invert M:

$$D(\underline{\hat{\theta}}) \geq M^{-1} = \frac{\sigma^2}{n\sum x_i^2 - \left(\sum x_i\right)^2}\sum_{i=1}^{n}\begin{bmatrix} x_i^2 & -x_i \\ -x_i & 1 \end{bmatrix} \qquad \text{Eq. 2.18}$$

Note that this MVB can be calculated from the independent variables x alone. In case a certain precision is desired an optimal x can be chosen. In the literature a particular choice of independent variables is referred to as *experimental design*.

Note further that the MVB changes when we take a model with only θ_0 (then $D(\underline{\hat{\theta}}_0) \geq \sigma^2/n$) or with only θ_1 (then $D(\underline{\hat{\theta}}_1) \geq \sigma^2/(\sum x_i^2)$). These two MVBs are smaller than or equal to the diagonal elements in Eq. 2.18.□

Estimators whose covariance reaches the MVB are called efficient.

Maximum Likelihood Estimators

The estimator that maximizes the PDF, or equivalently the log-likelihood function $\log f(\underline{y}|\theta)$ is called the Maximum Likelihood (ML) estimator. The ML estimator can be interpreted as providing the value of θ that makes the measurement most likely. It leads to the likelihood equation:

$$\frac{\partial}{\partial\theta}\log f(\underline{y}|\underline{\hat{\theta}}_{ML}) = 0 \qquad \text{Eq. 2.19}$$

Without proof we give here

> Theorem 2.3. If there exists an unbiased estimator having the MVB as covariance, it can be determined as a solution of the likelihood equation.

When the MVB cannot be reached, then, asymptotically, the error covariance matrix can be approximated by the inverse of the Fisher information matrix: M^{-1}.

Now suppose we have the model $\underline{y}_i = g(x_i|\theta) + \underline{\nu}_i$ (Eq. 2.1) where the noise is assumed to be Gaussian with zero mean and covariance matrix $D(\underline{\nu}) = V$. Then the likelihood function is given by (see Transformation of random variables on page 48):

$$f(\underline{y}|\theta) = ((2\pi)^n detV)^{-1/2} e^{-\frac{1}{2}(\underline{y}-g)^T V^{-1}(\underline{y}-g)} \qquad \text{Eq. 2.20}$$

and the likelihood equation is:

$$\frac{\partial}{\partial\theta}\log f(\underline{y}|\underline{\hat{\theta}}_{ML}) = -\frac{1}{2}\frac{\partial}{\partial\theta}\{(\underline{y}-g)^T V^{-1}(\underline{y}-g)\} = 0 \qquad \text{Eq. 2.21}$$

Now the term between the brackets is the sum of the squares of the residuals $\underline{y}_i - g(x_i|\theta)$ weighted with V^{-1}, which is the Least Squares performance index. Maximizing the log likelihood function is equivalent to minimizing the LS performance index. Thus under the assumptions that:

- the noise $\underline{\nu}$ is additive and Gaussian,
- the parameters θ are unknown, and
- the noise $\underline{\nu}$ and the parameters θ are independent,

the Maximum Likelihood and Least Squares estimators are identical. In practice Least Squares estimation techniques are also widely applied when the assumptions are not fulfilled, see section 2.2 and section 2.3. Then there is no guarantee that the MVB will be reached.

2.2 Linear Least Squares

In this section we will investigate the use of least squares techniques to estimate the unknown parameters in linear models. When the function $g(\xi_i|\theta)$ is linear in the parameters θ_j we get

$$\underline{y}_i = \sum_{j=0}^{k} g_j(\xi_i)\theta_j + \underline{\nu}_i \qquad \text{Eq. 2.22}$$

which represents the general form of a linear model. Note that, in general, the observations are nonlinear in the independent variables.

For example, with a polynomial fit up to order *k* we can write

$$\underline{y}_i = \sum_{j=0}^{k} \xi_i^j\theta_j + \underline{\nu}_i = 1\theta_0 + \xi_i\theta_1 + \xi_i^2\theta_2 + \ldots + \xi_i^k\theta_k + \underline{\nu}_i \qquad \text{Eq. 2.23}$$

We can collect the $g_j(\xi_i)$ into a matrix *X* with elements $x_{ij} = g_j(\xi_i)$ and thus we arrive at:

$$\underline{y_i} = 1\theta_0 + x_{i1}\theta_1 + x_{i2}\theta_2 + \ldots + x_{ik}\theta_k + \underline{\nu}_i = x_{i\cdot}\ \theta + \underline{\nu}_i \text{ and} \qquad \text{Eq. 2.24}$$

$$\underline{y} = X\theta + \underline{\nu} \qquad \text{Eq. 2.25}$$

where *y* and ν are column vectors of length *n*, representing the stochastic observations and additive noise, both Gaussian, independent and identically distributed (iid) with unknown variance σ^2: $N_y(X\theta, \sigma^2 I)$ $\quad N_\nu(0, \sigma^2 I)$. θ is a column vector $(\theta_0\theta_1\ldots\theta_k)^T$ of length $k+1$, containing the unknown parameters, which we assume to be unconstrained. The independent variables are contained in the design matrix *X,* which contains *n* rows (one row for each observation) $x_{i\cdot} = (x_{i0}x_{i1}\ldots x_{ik})$ $\quad x_{i0} \equiv 1$ and $k+1$ columns (one column for each parameter). In the special case that the observations are also linear in the independent variables (straight line), we have $x_{i1} = g_1(\xi_i) = \xi_i$. The linear relation Eq. 2.24 is also called a regression, and the estimation of θ a regression analysis. Eq. 2.24 represents a Gauss-

Markoff model, which, depending upon the column rank of X is of full rank or not of full rank.

Intermezzo 2.1 Generalized Least Squares

The generalized linear model posesses a more general covariance matrix of the observations. It reads $N_y(X\theta, \sigma^2 V)$ $\quad N_\nu(0, \sigma^2 V)$. Assume that V is known and positive definite, so that V^{-1} exists. First we will show how the model with the general covariance matrix V can be transformed into a model with $D(\underline{\nu}') = D(\underline{y}') = \sigma^2 I$. Let the Cholesky decomposition (Eq. 2.105) of the symmetric, positive definite matrix V^{-1} be

$$V^{-1} = GG^T \text{ and let } y' = G^T y \qquad X' = G^T X \qquad \nu' = G^T \nu \ . \qquad \text{Eq. 2.26}$$

Then premultiplication of Eq. 2.25 with G^T gives the desired model $\underline{y}' = X'\theta + \underline{\nu}'$ with

$$D(\underline{\nu}') = D(G^T\underline{\nu}) = G^T D(\underline{\nu}) G = G^T \sigma^2 (GG^T)^{-1} G = \sigma^2 I \qquad \text{Eq. 2.27}$$

where we have used Eq. 2.8. This simpler model, where the observations are uncorrelated and have equal variances $E[\underline{\nu}'_i \underline{\nu}'_j] = \sigma^2 \delta_{ij}$, is called homoscedastic. Models with observations possessing different variances are called heteroscedastic. The heteroscedastic model can easily be transformed to the homoscedastic model of Eq. 2.25 using Eq. 2.26. Thus in the following we will only use the homoscedastic model. □

In the method of least squares the sum of the squares of the residuals $\underline{S}(\theta)$ is minimized with respect to θ

$$\frac{\partial}{\partial\theta}\underline{S}(\theta) = \frac{\partial}{\partial\theta}\{(\underline{y} - X\theta)^T(\underline{y} - X\theta)\} = 0 \qquad \text{Eq. 2.28}$$

which gives (using Eq. 2.93)

$$-2X^T(\underline{y} - X\hat{\underline{\theta}}) = 0 \qquad \text{Eq. 2.29}$$

which is equivalent to

$$X^T X \hat{\underline{\theta}} = X^T \underline{y} \qquad \text{Eq. 2.30}$$

Eq. 2.30 are called the normal equations since the residuals $\hat{\underline{e}} = \underline{y} - X\hat{\underline{\theta}}$ are orthogonal to the least squares estimate $\hat{\underline{y}} = X\hat{\underline{\theta}} : \hat{\underline{y}}^T \hat{\underline{e}} = 0$ cf. Eq. 2.29 and Fig.2.4.

When X is of full rank the solution of Eq. 2.30 is given by

$$\hat{\underline{\theta}} = (X^T X)^{-1} X^T \underline{y} \equiv X^\dagger y \qquad \text{Eq. 2.31}$$

where $X^\dagger$ denotes the Moore-Penrose generalized inverse of X. We recall that with additive normally distributed noise this least squares estimator is the maximum likelihood estimator and its variance

$$D(\underline{\hat{\theta}}) = D(X^{\dagger}\underline{y}) = X^{\dagger}D(\underline{y})X^{\dagger^T} = (X^TX)^{-1}X^T\sigma^2IX(X^TX)^{-1} = \sigma^2(X^TX)^{-1} \qquad \text{Eq. 2.32}$$

achieves the MVB. Check this with the help of Eq. 2.12, Eq. 2.20 and Eq. 2.93. When X is not of full rank a generalized inverse has to be taken, a case which will not be treated here.

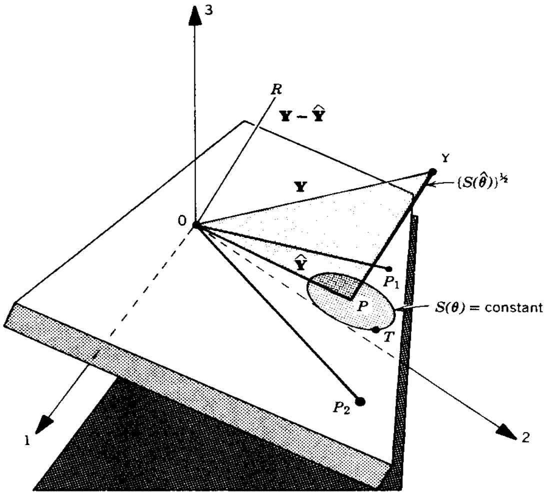


Fig.2.4. Example of geometry of linear least squares in sample space, for explanation see the text with Fig.2.11.

Example 2.4. Returning to the straight line model of Example 2.3

$$\underline{y}_i = \theta_0 + x_i\theta_1 + \underline{\nu}_i \qquad \text{Eq. 2.33}$$

the design matrix X reads

$$X = \begin{bmatrix} 1 & x_1 \\ \dots & \dots \\ 1 & x_n \end{bmatrix} \qquad \text{Eq. 2.34}$$

and the unweighted least squares estimator is given by (use Eq. 2.8)

$$\underline{\hat{\theta}} = (X^TX)^{-1}X^T\underline{y} = \frac{1}{n\sum x_i^2 - (\sum x_i)^2}\sum_{i=1}^{n}\begin{bmatrix} x_i^2 & -x_i \\ -x_i & 1 \end{bmatrix}\begin{bmatrix} \sum y_i \\ \sum x_iy_i \end{bmatrix} =$$

$$= \frac{1}{n\sum x_i^2 - (\sum x_i)^2}\begin{bmatrix} \sum x_i^2 \sum y_i - \sum x_i \sum x_iy_i \\ n\sum x_iy_i - \sum y_i \sum x_i \end{bmatrix} = \frac{1}{\overline{x^2} - \bar{x}^2}\begin{bmatrix} \overline{x^2}\cdot\bar{y} - \overline{xy}\cdot\bar{x} \\ \overline{xy} - \bar{y}\cdot\bar{x} \end{bmatrix} \qquad \text{Eq. 2.35}$$

where $\bar{x} = (\sum x_i)/n$,$\bar{y} = (\sum y_i)/n$, $\overline{xy} = (\sum x_iy_i)/n$, and $\overline{x^2} = (\sum x_i^2)/n$.

The autocovariance matrix is given by

$$D(\underline{\hat{\theta}}) = \begin{bmatrix} \mathrm{var}(\theta_0) & \mathrm{cov}(\theta_0, \theta_1) \\ \mathrm{cov}(\theta_0, \theta_1) & \mathrm{var}(\theta_1) \end{bmatrix} = \frac{\sigma^2}{n\sum x_i^2 - (\sum x_i)^2} \sum_{i=1}^{n} \begin{bmatrix} x_i^2 & -x_i \\ -x_i & 1 \end{bmatrix} \qquad \text{Eq. 2.36}$$

which is identical to the MVB derived in Eq. 2.18.

Note that even with this very simple linear model the analytical solution is quite complicated. Therefore in practice numerical methods are used to calculate least squares estimates and their autocovariance matrix. The most popular method, which avoids the calculation of the $(X^T X)^{-1}$ that would lead to loss of numerical accuracy, is based on the QR decomposition.

Stages of a linear regression analysis

The stages of a linear regression analysis consist of the following:

(a) Use the data y and the design matrix X to estimate the parameters $\hat{\theta} = X^{\dagger} y$, and compute the residuals and summary statistics (t-values and correlation coefficients).

(b) Judge whether the fit is adequate and the residuals are acceptable. Some systematic error may be acceptable. If the fit is not satisfactory, extend the model and continue with step (a).

(c) Investigate sensibleness of the parameter estimates. If a parameter is not significantly different from zero, omit its contribution from the model, and continue with step (a) using this reduced model.

Example 2.5. In spectroscopy the properties of a mixture of components are a superposition of the spectroscopic properties of the components weighted by their concentration. With absorption this is known as the Beer-Lambert law. Thus the observed spectrum $\psi(\lambda)$ is a superposition of the contributions of the n_{comp} different components:

$$\underline{\psi}(\lambda) = \sum_{l=1}^{n_{\mathrm{comp}}} c_l \varepsilon_l(\lambda) + \underline{\nu}(\lambda) \qquad \text{Eq. 2.37}$$

where c_l and $\varepsilon_l(\lambda)$ denote, respectively, the concentration and the spectrum of component l. In vector and matrix form this equation reads:

$$\underline{\psi}_{\lambda_j} = \sum_{l=1}^{n_{\mathrm{comp}}} \varepsilon_{\lambda_j l} c_l + \underline{\nu}_{\lambda_j} \qquad \underline{\psi} = Ec + \underline{\nu} \qquad \text{Eq. 2.38}$$

$\varepsilon_{\lambda_j l}$ denotes the contribution of component l at wavelength λ_j, and is the jl-th element of the spectral matrix E. Matrix E has n_{λ} rows and n_{comp} columns. Both column vectors $\underline{\psi}$ and $\underline{\nu}$ are of length n_{λ}, whereas the column vector c is of length n_{comp}. Note that when the spectra of the components are known this represents a linear regression. The parameters to be

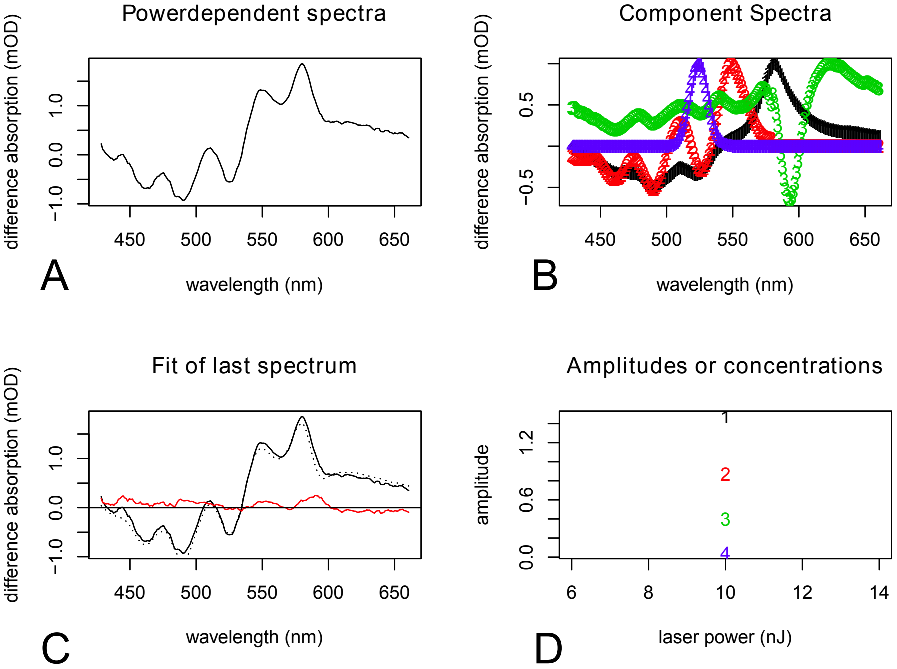


Fig.2.5. (A) measured spectrum at 10nJ (B) spectra of the four different components (C) fit (dashed) of data (solid) and residuals (solid line straddling the x-axis) (D) four amplitude estimates.

estimated are the concentrations of the components. Formally, the estimates are

$$\hat{\underline{c}} = (E^T E)^{-1} E^T \underline{\psi} \qquad \text{Eq. 2.39}$$

Data described by the model from Eq. 2.38 can be measured as a function of a second variable, namely the laser power. Since all measurements at different laser power (indicated by index p) are statistically independent, and share the same matrix of component spectra E, we can use a single matrix equation (omitting the additive noise for clarity)

$$\underline{\Psi}_{\lambda_j p} = \sum_{l=1}^{n_{\mathrm{comp}}} \varepsilon_{\lambda_j l} c_{pl} \qquad \underline{\Psi} = EC^T \qquad \text{Eq. 2.40}$$

where matrix $\underline{\Psi}$ has n_{λ} rows and n_{power} columns. Matrix C contains the amplitude parameters of all components at the different laser powers, it has n_{power} rows and n_{comp} columns. The least squares estimator of (the transpose of) this matrix C is given by:

$$\hat{\underline{C}}^T = (E^T E)^{-1} E^T \underline{\Psi} = E^{\dagger} \underline{\Psi} \qquad \text{Eq. 2.41}$$

Determining the linear least squares estimates using the QR decomposition

In the linear least squares problem $\underline{y} = X\theta + \underline{\nu}$, $D(\underline{\nu}) = \sigma^2 I$ of full rank we have

$$\underline{\hat{\theta}} = (X^T X)^{-1} X^T \underline{y} = X^{\dagger} \underline{y} \qquad \text{Eq. 2.42}$$

where $X^{\dagger}$ denotes the Moore-Penrose generalized inverse of *X*. We will discuss and illustrate the use of the QR decomposition of the design matrix *X* to calculate $\hat{\theta}$. The advantage of using the QR decomposition (or of using the SVD) is numerical stability. Let the QR decomposition of the $n \times q$ matrix *X* be

$$X = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} \qquad \text{Eq. 2.43}$$

R is a $q \times q$ upper triangular matrix. Q_1 and Q_2 are, respectively, $n \times q$ and $n \times (n-q)$ matrices such that $\begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ is an $n \times n$ orthogonal matrix. From Eq. 2.43 we have $X = Q_1 R$ which is termed the skinny QR decomposition. Using $Q_1^T X = R$ we have $Q_1^T \underline{y} = R\underline{\hat{\theta}}$, a triangular system which can easily be solved for $\underline{\hat{\theta}}$. The Moore-Penrose generalized inverse is given by

$$X^{\dagger} = R^{-1} Q_1^T \qquad \text{Eq. 2.44}$$

Thus $X^{\dagger} X = I_q$ and $XX^{\dagger} = Q_1 Q_1^T$ is the orthogonal projection operator which projects the observation vector $\underline{y}$ onto the column space of Q_1 (and thus of *X*). We will formalise this into

> Theorem 2.4. The orthogonal projection matrix *P* that projects an arbitrary vector on the column space of *X* is given by

$$P = XX^{\dagger} = Q_1 Q_1^T \qquad \text{Eq. 2.45}$$

Proof: An orthogonal projection matrix is defined by $P^2 = P$ and $P^T = P$. Check: the projection *Py* is orthogonal to $y - Py$: $(Py)^T(y - Py) = y^T(P^T - P^T P)y = 0$. Now $P^2 = Q_1 Q_1^T Q_1 Q_1^T = Q_1 Q_1^T = P$ because of the orthogonality of the columns of the *Q* matrix. The symmetry of *P* is straightforward.□

All possible expected response vectors $X\theta$ form the so-called expectation plane. The residual vector is orthogonal to this plane and is given by

$$\underline{\hat{e}} = (I - XX^{\dagger})\underline{y} = (I - P)\underline{y} = Q_2 Q_2^T \underline{y} \qquad \text{Eq. 2.46}$$

From Eq. 2.8, Eq. 2.42 and Eq. 2.44 we can easily find the covariance matrix of the estimator:

$$D(\underline{\hat{\theta}}) = D(X^{\dagger}\underline{y}) = X^{\dagger}D(\underline{y})X^{\dagger T} = \sigma^2 R^{-1}R^{-T} \qquad \text{Eq. 2.47}$$

where we have used $D(\underline{y}) = \sigma^2 I$.

The covariance matrix of the residuals follows directly from Eq. 2.46

$$D(\underline{\hat{e}}) = D((I-P)\underline{y}) = (I-P)D(\underline{y})(I-P) = \sigma^2(I-P) \qquad \text{Eq. 2.48}$$

Since $\mathrm{trace}(AB) = \mathrm{trace}(BA)$ (Eq. 2.100) we find $\mathrm{trace}(Q_2Q_2^T) = \mathrm{trace}(Q_2^TQ_2) = \mathrm{trace}(I_{n-q}) = n-q = \mathrm{rank}(Q_2Q_2^T)$, where $q \equiv \mathrm{rank}X = k+1$. Thus we have

$$D(\underline{\hat{e}}) = \sigma^2(I-P) \qquad \mathrm{rank}D(\underline{\hat{e}}) = n-q \qquad \mathrm{trace}D(\underline{\hat{e}}) = \sigma^2(n-q) \qquad \text{Eq. 2.49}$$

We can use Eq. 2.49 to estimate an unknown σ^2:

$$\underline{\hat{\sigma}}^2 = \frac{\underline{\hat{e}}^T\underline{\hat{e}}}{n-q} \qquad \text{Eq. 2.50}$$

In words: the estimator of the variance is equal to the residual sum of squares divided by the number of degrees of freedom for this sum, $n-q$. This estimator is unbiased:

$$E[\underline{\hat{\sigma}}^2] = \frac{E[\underline{\hat{e}}^T\underline{\hat{e}}]}{n-q} = \frac{E[\mathrm{trace}D(\underline{\hat{e}})]}{n-q} = \sigma^2 \qquad \text{Eq. 2.51}$$

where we have used Eq. 2.49. To find the least squares estimator of $D(\underline{\hat{\theta}})$ when σ^2 is unknown, we change D into $\underline{\hat{D}}$ and σ^2 into $\underline{\hat{\sigma}}^2$ in Eq. 2.47 to find

$$\underline{\hat{D}}(\underline{\hat{\theta}}) = \underline{\hat{\sigma}}^2 R^{-1}R^{-T} \qquad \text{Eq. 2.52}$$

Intermezzo 2.2 Chi-squared, F and t distributions

When $\underline{x}$ is $N_x(0, I)$ then $\underline{x}^T\underline{x} = \underline{u}$ has the central χ^2 distribution with n degrees of freedom. The random variable $\underline{u}$ possesses expectation and variance:

$$\underline{u} = \underline{x}^T\underline{x} \sim \chi_n^2 \qquad E[\underline{u}] = n \qquad D(\underline{u}) = 2n \qquad \text{Eq. 2.53}$$

Two independent variables each having central χ^2 distributions form the basis of the central F-distribution. If $\underline{u}_1 \sim \chi^2_{n_1}$ and $\underline{u}_2 \sim \chi^2_{n_2}$ then $\underline{v} = \dfrac{\underline{u}_1/n_1}{\underline{u}_2/n_2} \sim F_{n_1,n_2}$, the central F-distribution with n_1 and n_2 degrees of freedom. The random variable $\underline{v}$ has expectation and variance:

$$\underline{v} = \frac{\underline{u}_1/n_1}{\underline{u}_2/n_2} \sim F_{n_1, n_2} \qquad E[\underline{v}] = \frac{n_2}{n_2-2} \quad (n_2 > 2) \qquad D(\underline{v}) = \frac{2n_2^2(1+(n_2-2)/n_1)}{(n_2-2)^2(n_2-4)} \quad (n_2 > 4) \qquad \text{Eq. 2.54}$$

Finally, the ratio of a normally distributed variable to one that has a χ^2 distribution is the basis of Student's *t*-distribution. Thus $z = \frac{x}{\sqrt{u/n}} \sim t_n$. Its mean and variance are given by

$$z = \frac{\underline{x}}{\sqrt{\underline{u}/n}} \sim t_n \qquad E[\underline{z}] = 0 \qquad D(\underline{z}) = \frac{n}{n-2} \qquad \text{Eq. 2.55}$$

Note that $z^2 = \frac{\underline{x}^2}{\underline{u}/n} \sim F_{1,n}$. Some examples of these three distributions are shown in Fig.2.6., Fig.2.7., and Fig.2.8.

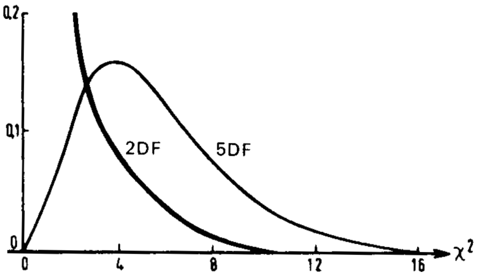


Fig.2.6. Probability density of the central χ_n^2 distribution for $n = 2$ and $n = 5$.

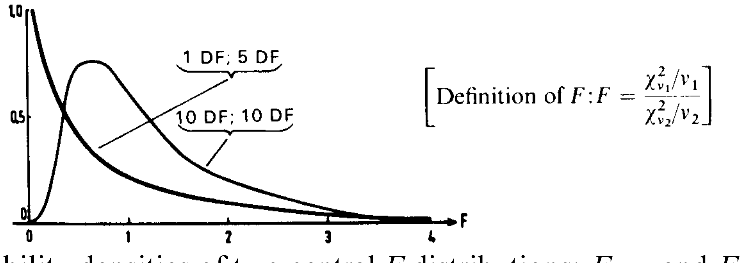


Fig.2.7. Probability densities of two central *F* distributions: $F_{1,5}$ and $F_{10,10}$.

Confidence intervals and confidence regions

Armed with these three distributions we can now derive inference regions for θ. When the model is correct and of full rank then the estimated parameter vector is normally distributed:

$$\underline{\hat{\theta}} \sim N(\theta, D(\theta)) \qquad \text{Eq. 2.56}$$

where $D(\theta)$ is estimated from Eq. 2.52. Thus for a single parameter $\underline{\hat{\theta}}_j$ we find

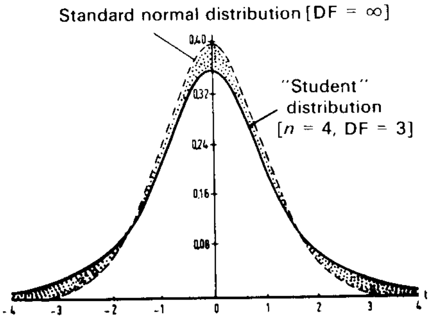


Fig.2.8. Probability density of the $N(0, 1)$ distribution (dashed) and the Student t-distribution (solid) with 3 degrees of freedom ($n = 4$ observations). With a decreasing number of degrees of freedom the maximum of the Student t-distribution drops and the shaded area grows. In comparison with the $N(0, 1)$ distribution more probability is concentrated in the tails and less in the central part.

$$\frac{\underline{\hat{\theta}}_j - \theta_j}{\sigma_{\underline{\theta}_j}} \sim N(0, 1) \qquad \text{Eq. 2.57}$$

When we possess only an estimate of σ^2, based upon $n - q$ degrees of freedom, we get by definition a t-distribution

$$\frac{\underline{\hat{\theta}}_j - \theta_j}{\hat{\sigma}_{\underline{\hat{\theta}}_i}} \sim t_{n-q} \qquad \text{Eq. 2.58}$$

Since the t-distribution is symmetric we have $P[t \le -t_{n-q, \alpha/2}] = P[t \ge t_{n-q, \alpha/2}] = \frac{\alpha}{2}$ and $P[-t_{n-q, \alpha/2} \le t \le t_{n-q, \alpha/2}] = 1 - \alpha$. Thus we have the following $100(1-\alpha)\%$ confidence limits for θ_j:

$$\underline{\hat{\theta}}_j \pm t_{n-q, \alpha/2}\hat{\sigma}_{\underline{\hat{\theta}}_j} . \qquad \text{Eq. 2.59}$$

The remainder of this section can best be understood in combination with the next section on the geometry of least squares. To obtain a joint $100(1-\alpha)\%$ confidence region for all the parameters θ we must find the distribution of $(\theta - \underline{\hat{\theta}})^T X^T X(\theta - \underline{\hat{\theta}})$ (see Eq. 2.125). By definition the sum of squares is given by

$$\underline{S}(\theta) = (\underline{y} - X\theta)^T(\underline{y} - X\theta) = \underline{\nu}^T\underline{\nu} \sim \sigma^2\chi^2_n \qquad \text{Eq. 2.60}$$

which, using the normal equations, can be written as

$$\underline{S}(\theta) = \underline{y}^T\underline{y} + \theta^T X^T X\theta - 2\theta^T X^T\underline{y} = \underline{y}^T\underline{y} + \theta^T X^T X\theta - 2\theta^T X^T X\underline{\hat{\theta}} \qquad \text{Eq. 2.61}$$

Note that $\underline{S}(\theta)$ possesses a component in X-space. The residual sum of squares is given by

$$\underline{S}(\underline{\hat{\theta}}) = (\underline{y} - X\underline{\hat{\theta}})^T(\underline{y} - X\underline{\hat{\theta}}) = SSE \sim \sigma^2\chi^2_{n-q} \qquad \text{Eq. 2.62}$$

which, using the normal equations, can also be written as (applying Pythagoras)

$$\underline{S}(\hat{\underline{\theta}}) = \underline{y}^T\underline{y} + \hat{\underline{\theta}}^T X^T X\hat{\underline{\theta}} - 2\hat{\underline{\theta}}^T X^T\underline{y} = \underline{y}^T\underline{y} - \hat{\underline{\theta}}^T X^T\underline{y} = \underline{y}^T\underline{y} - \hat{\underline{\theta}}^T X^T X\hat{\underline{\theta}} \qquad \text{Eq. 2.63}$$

Note that $\underline{S}(\hat{\underline{\theta}})$ possesses no component in *X*-space. Thus the difference of these two sums of squares is equal to (applying Pythagoras)

$$\underline{S}(\theta) - \underline{S}(\hat{\underline{\theta}}) = (\theta - \hat{\underline{\theta}})^T X^T X(\theta - \hat{\underline{\theta}}) \sim \sigma^2\chi_q^2 \qquad \text{Eq. 2.64}$$

Since $\underline{S}(\hat{\underline{\theta}})$ and $\underline{S}(\theta) - \underline{S}(\hat{\underline{\theta}})$ are distributed independently, their ratio corresponds to an *F*-distribution:

$$\frac{\underline{S}(\theta) - \underline{S}(\hat{\underline{\theta}})}{q} \Big/ \frac{\underline{S}(\hat{\underline{\theta}})}{n-q} \sim F_{q,\,n-q} \qquad \text{Eq. 2.65}$$

Combining these last two equations a joint $100(1-\alpha)\%$ confidence region for all the parameters θ is defined by

$$(\theta - \hat{\underline{\theta}})^T X^T X(\theta - \hat{\underline{\theta}}) \le q\hat{\underline{\sigma}}^2 F_{q,\,n-q,\,\alpha} \qquad \text{Eq. 2.66}$$

where $P[F \ge F_{q,\,n-q,\,\alpha}] \equiv \alpha$. From Eq. 2.65 we find

$$\underline{S}(\theta) = \underline{S}(\hat{\underline{\theta}})\left(1 + \frac{q}{n-q}F_{q,\,n-q,\,\alpha}\right) \qquad \text{Eq. 2.67}$$

which defines a contour with $\underline{S}(\theta) = \text{constant}$. We will examine such contours in the sample space (in which the mechanism of linear least squares can be best understood) and in the parameter space.

Example 2.6. For the linear model of Eq. 2.4 we again simulated 100 estimates of the parameters $(\hat{\theta}_0, \hat{\theta}_1)_j$ for the intercept and the slope of the straight line. The number of degrees of freedom (*df*) of each realization was only five (seven datapoints minus two parameters), which is different from Fig.2.3. where it was 29. We again investigate the properties of these 100 estimates, see Fig.2.9. The scatterplot in Fig.2.9.A shows that approximately 90% of the estimates are within the 90% confidence region around the true values. Conversely, in 90% of the realizations the true value falls within the 90% confidence region around that estimate. The histograms of the estimated parameters are well described by normal distributions according to Eq. 2.57, centered at the true values with standard deviation equal to the rms width of the histogram. The mean square error histogram (Fig.2.9.D) behaves according to $(\sigma^2\chi_{df}^2)/df$, Eq. 2.62, with $\sigma = 0.1$. Finally, histograms of the deviation from the true value divided by the estimated standard deviation, $(\hat{\theta} - \theta_{\text{true}})/\hat{\sigma}_{\hat{\theta}}$, of intercept estimates (Fig.2.9.E) and slope estimates (Fig.2.9.F) are well described by t_{df}-distributions according to Eq. 2.58 (solid lines). Note that with the small $df = 5$ these differ clearly from standard normal distributions (dotted). □

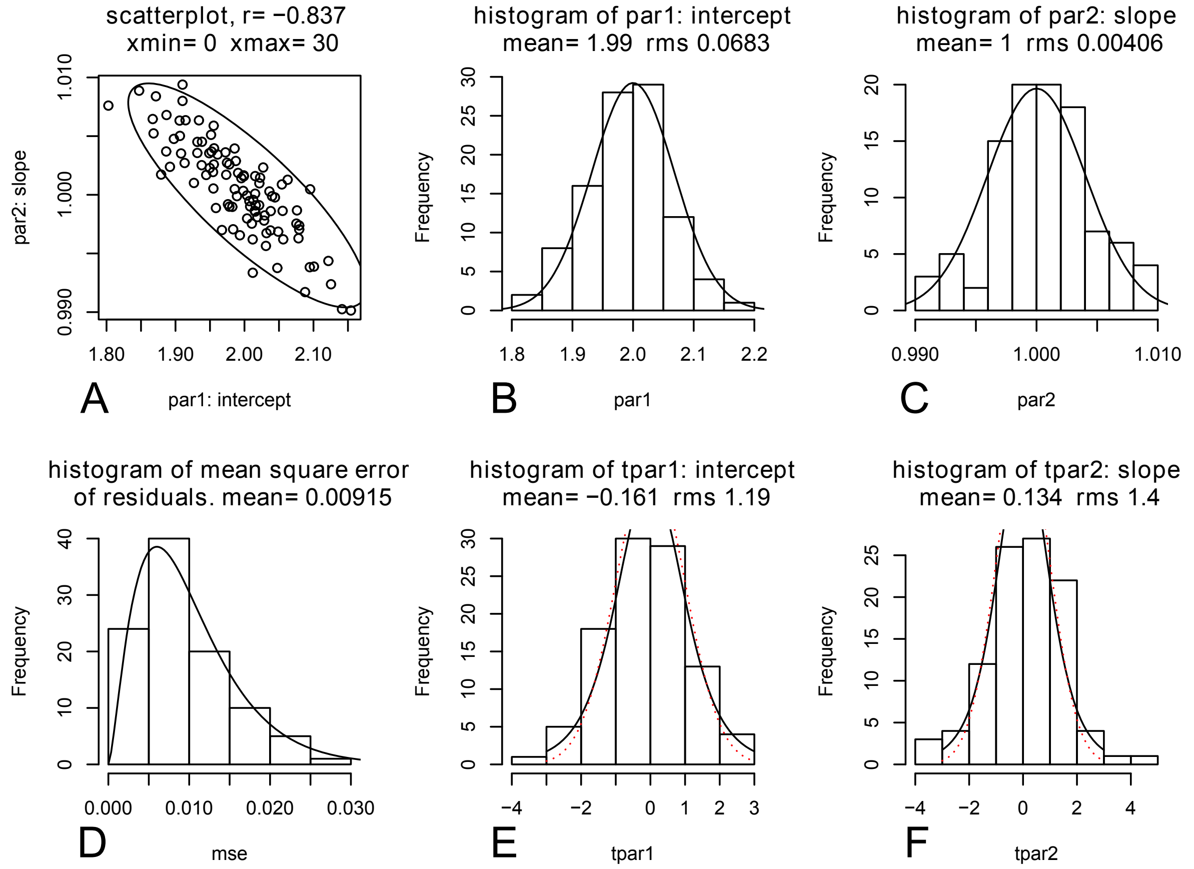


Fig.2.9. Scatterplot of all hundred estimates $(\hat{\theta}_0, \hat{\theta}_1)_j$ for the intercept and the slope of the straight line (A), histogram of intercept estimates (B) and slope estimates (C). Histograms of mean square error (D), and of the deviation from the true value divided by the estimated standard deviation, $(\hat{\theta} - \theta_{\text{true}})/\hat{\sigma}_{\hat{\theta}}$, of intercept estimates (E) and slope estimates (F). In addition theoretical results have been added: (A) the 90% confidence ellipse according to Eq. 2.67, centered around the estimate closest to the true values, (B) and (C) normal distributions according to Eq. 2.57, centered at the true values with standard deviation equal to the rms width of the histogram, (D) $(\sigma^2\chi^2_{df})/df$ according to Eq. 2.62, (E) and (F) t_{df}-distributions according to Eq. 2.58 (solid lines) and standard normal distributions (dotted). Note that here $df = 5$.

Testing for the significance of a parameter

Having derived the above distributions we can now test the null hypothesis $H_0 : \theta_j = A$. Most important is to test the null hypothesis $H_0 : \theta_j = 0$. The alternative hypothesis is then $H_1 : \theta_j \neq 0$. Define the *t*-statistic

$$\underline{t}_A = \frac{\underline{\hat{\theta}}_j - A}{\underline{\hat{\sigma}}_{\underline{\hat{\theta}}_i}} \qquad \text{Eq. 2.68}$$

which under the null hypothesis $H_0 : \theta_j = A$ is distributed as t_{df} (*df*=degrees of freedom of the fit, $df = n - q$) with mean *A*. In the following we take $A = 0$. If a confidence level $\alpha = 5\%$ is chosen, H_0 is rejected when $\underline{t}_0 = \underline{\hat{\theta}}_j / \underline{\hat{\sigma}}_{\underline{\hat{\theta}}_j}$ exceeds one of the critical values $\pm t_{df, \alpha/2}$. This is illustrated in Fig.2.10., where the filled areas indicate the regions where H_0 is rejected. As a rule of tumb $\underline{t}_0 = \underline{\hat{\theta}}_j / \underline{\hat{\sigma}}_{\underline{\hat{\theta}}_j}$ must exceed 2 for a parameter to be considered significantly different from zero, however with small $df = n - q$ it has to be larger! E.g. with $df = 12$, $t_{df, \alpha/2} = t_{12, 0.025} = 2.18$ and with $df = 3$, $t_{df, \alpha/2} = t_{3, 0.025} = 3.18$.

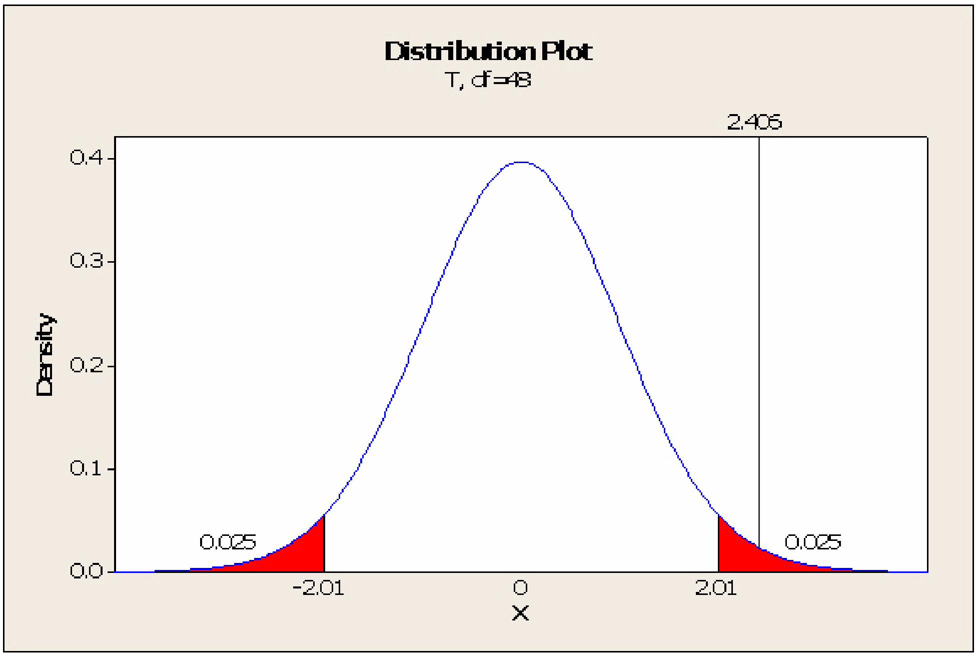


Fig.2.10. Testing the null hypothesis $H_0 : \theta_j = 0$. The observed $\underline{t}_0 = \underline{\hat{\theta}}_j / \underline{\hat{\sigma}}_{\underline{\hat{\theta}}_j} = 2.406$ exceeds $t_{df, \alpha/2} = t_{48, 0.025} = 2.01$, therefore H_0 is rejected and we conclude that the parameter is significantly different from zero.

Geometry of linear least squares

The sample space is an n-dimensional space which contains the observation vector y and the column vectors of X. An example with $n = 3$ and $q = 2$ is shown in Fig.2.11.

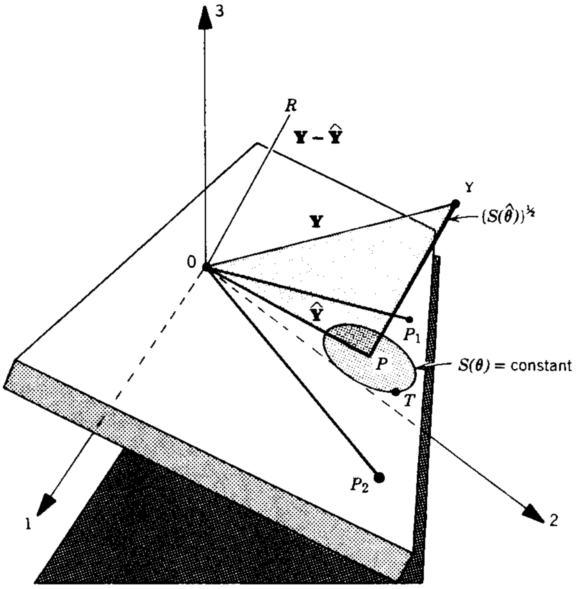


Fig.2.11. Example of geometry of linear least squares in sample space.

The two column vectors of X are designated $\overline{OP_1}$ and $\overline{OP_2}$. The projection of y onto the plane defined by the column vectors of X is equal to $\hat{y}$ ($\overline{OP}$) whereas the residual ($\overline{OR}$) $e = y - \hat{y}$ is orthogonal to $\hat{y}$. The circle around P through T represents a contour with $\underset{\sim}{S}(\theta) = \text{constant}$. The radius of the circle is given by $\sqrt{2S(\hat{\theta})F_{2,\,1,\,\alpha}}$.

The parameter space is a q-dimensional space in which a set of values $(\theta_0, \theta_1, \ldots, \theta_{q-1})$ of the parameters defines a point. The minimum value of $S(\theta)$ is attained at $\hat{\theta}$.

A contour $S(\theta) = K$ is defined by $(\theta - \hat{\theta})^T X^T X(\theta - \hat{\theta}) = K - S(\hat{\theta})$, which defines a closed ellipsoidal contour surrounding the point $\hat{\theta}$. When K is given by the right-hand side of Eq. 2.67 the contour encloses the $100(1-\alpha)\%$ confidence region. The contours in Fig.2.12. are concentric ellipses around $(\hat{\theta}_1, \hat{\theta}_2)$. Both the orientation and the shape of the ellipses are important. The principal axes of the ellipses can be found from the eigenvector/eigenvalue decomposition of $X^T X$. When the principal axes are parallel to the θ_1 and θ_2 axes, the estimates $\hat{\theta}_1$ and $\hat{\theta}_2$ are independent. Define $\tilde{\theta} = \theta - \hat{\theta}$ and let the eigenvector/eigenvalue decomposition of $X^T X$ be

$$X^T X = V \Lambda V^T \qquad \text{Eq. 2.69}$$

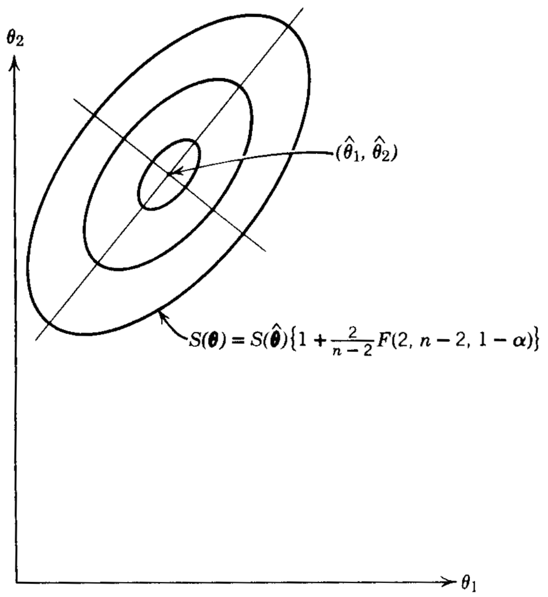

θ_2
$(\hat\theta_1, \hat\theta_2)$
$S(\boldsymbol{\theta}) = S(\hat{\boldsymbol{\theta}})\{1 + \frac{2}{n-2}F(2, n-2, 1-\alpha)\}$
θ_1

Fig.2.12. Example of geometry of linear least squares in parameter space.

Then the contours are defined by (see Eq. 2.66)

$$\tilde{\theta} V \Lambda V^T \tilde{\theta} = K - S(\hat{\theta}) \qquad \text{Eq. 2.70}$$

Now consider $\tilde{\theta} = c_i v_i$, where v_i is the eigenvector belonging to eigenvalue λ_i. Then from $c_1^2 \lambda_1 = c_2^2 \lambda_2$ we find that the lengths of the axes of the ellipse are proportional to $\lambda_i^{-1/2}$. Along the direction v_1, the short axis, θ is determined best, whereas along the direction v_2, the long axis, θ is determined worse.

2.3 Nonlinear Least Squares

In this section we will investigate the use of least squares techniques to estimate the unknown parameters in nonlinear models of the form:

$$\underline{y}_i = g(x_i | \theta) + \underline{\nu}_i \qquad \text{Eq. 2.71}$$

where y and ν are $n \times 1$ column vectors representing the stochastic observations and additive Gaussian noise. g is a scalar nonlinear function of the independent variables ($1 \times l$ row vector x_i) and the unknown parameters ($k \times 1$ column vector θ). When we abbreviate $g(x_i | \theta) \equiv g_i(\theta)$ and write $g = (g_1, \ldots, g_n)^T$ the model reads

$$\underline{y} = g(\theta) + \underline{\nu} \qquad \text{Eq. 2.72}$$

We assume that both y and ν are Gaussian $N_y(g(\theta), \sigma^2 I)$ $N_\nu(0, \sigma^2 I)$. Now the least squares estimator of θ is found by minimizing the weighted sum of squares of the residuals S

with respect to θ

$$\frac{\partial}{\partial\theta}S(\hat{\underline{\theta}}) = \frac{\partial}{\partial\theta}\{(\underline{y}-g(\hat{\underline{\theta}}))^T(\underline{y}-g(\hat{\underline{\theta}}))\} = 0 \tag{Eq. 2.73}$$

which gives

$$\left(\frac{\partial}{\partial\theta}g(\hat{\underline{\theta}})\right)^T(\underline{y}-g(\hat{\underline{\theta}})) = 0 \tag{Eq. 2.74}$$

Note that the Jacobian $\frac{\partial g}{\partial\theta}$ is an $n \times k$ matrix. We recall that under the appropriate conditions, given on page 20, the least squares estimator is equal to the maximum likelihood estimator. The estimator is generally consistent and converges in distribution to $N_{\hat{\underline{\theta}}}(\theta, M^{-1})$ where M^{-1} is the minimum variance bound and M the Fisher information matrix.

$$M = E\left[\left(\frac{\partial}{\partial\theta}\log f\right)^T\left(\frac{\partial}{\partial\theta}\log f\right)\right] = \sigma^{-2}\left(\frac{\partial g}{\partial\theta}\right)^T\frac{\partial g}{\partial\theta} \tag{Eq. 2.75}$$

where we started from the log-likelihood function

$$\log f = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\underline{y}-g)^T(\underline{y}-g) \tag{Eq. 2.76}$$

and have used $E[(\underline{y}-g)(\underline{y}-g)^T] = \sigma^2 I$. When the errors are not normally distributed, but satisfy $D(\underline{\nu}) = \sigma_\nu^2 I$, then under very general conditions the least squares estimator is consistent and asymptotically normally distributed with covariance $\sigma_\nu^2\left\{\left(\frac{\partial g}{\partial\theta}\right)^T\left(\frac{\partial g}{\partial\theta}\right)\right\}^{-1}$, which in general is not the MVB. It is emphasized that, in both cases, consistence and convergence in distribution are *asymptotic* properties.

In contrast to the linear least squares problem there is no closed form solution of Eq. 2.74, and we must resort to numerical minimization techniques.

Intermezzo 2.3 Numerical minimization techniques

We will first consider the steepest descend method. In this method the current value of the parameter vector t_c is changed in the direction opposite to the gradient of the sum of squares of the residuals $S(\theta)$ by an amount

$$\Delta t_{SD} = -\Delta\frac{\mathrm{grad}_\theta S(t_c)}{\|\mathrm{grad}_\theta S(t_c)\|} \tag{Eq. 2.77}$$

where Δ is the stepsize.

Theorem 2.5. Given a trial point t_i and a search direction p_i, the stepsize Δ_i that minimizes S in the search direction must produce orthogonality of the gradient in t_{i+1} and the search direction p_i:

$$\frac{\partial}{\partial\theta}S(t_{i+1})p_i = 0 \qquad \text{Eq. 2.78}$$

Proof: the next trial point is given by $t_{i+1} = t_i + \lambda p_i$. Now minimization of $S(t_{i+1})$ with respect to λ gives $\frac{\partial}{\partial\lambda}S(t_{i+1}) = \frac{\partial}{\partial\theta}S(t_{i+1})\frac{\partial t_{i+1}}{\partial\lambda} = \frac{\partial}{\partial\theta}S(t_{i+1})p_i = 0$.□

Thus an optimal steepest descent step Δt_{SD} starts perpendicular to a contour $S(t) = S(t_i)$ and ends parallel to a contour $S(t) = S(t_{i+1})$. This is illustrated in Fig.2.13. The gradient direction depends crucially upon the manner in which the variables are scaled. When the contours of S form a so called banana-shaped valley, then the steepest descend method results in an uneconomic zigzag path. An advantage of the method is that under very general conditions convergence to a minimum is guaranteed.

Example 2.7. Suppose that $S(\theta) = \frac{1}{2}\theta^T Q\theta$ where Q is symmetric and positive definite. Then $\text{grad}S(\theta) = Q\theta$ and

$$t_{i+1} = \left(I - \Delta\frac{Q}{\|Qt_i\|}\right)t_i \qquad \text{Eq. 2.79}$$

Now let the Cholesky decomposition of Q be $Q = GG^T$ and let $\theta' = G^T\theta$ then $S(\theta') = \theta'^T\theta'$ and $t'_{i+1} = (I - \Delta/\|t_i\|)t'_i$. Thus with the transformed θ the minimum $\theta = 0$ can be reached in one step, whereas in general Δ in Eq. 2.79 cannot be chosen so as to make $t_{i+1} = 0$. This example illustrates how the performance of the steepest descend method depends upon the scaling of the variables.□

Other minimization methods make use of more terms than only the gradient in a Taylor expansion of $S(t_c)$:

$$S(t_c + \tau) = S(t_c) + \frac{\partial}{\partial\theta}S(t_c)\tau + \frac{1}{2}\tau^T\frac{\partial^2}{\partial\theta^2}S(t_c)\tau \qquad \text{Eq. 2.80}$$

The Newton method minimizes in every iteration this quadratic polynomial. Minimization of the right hand side of Eq. 2.80 with respect to τ requires (using Eq. 2.93 for differentiation)

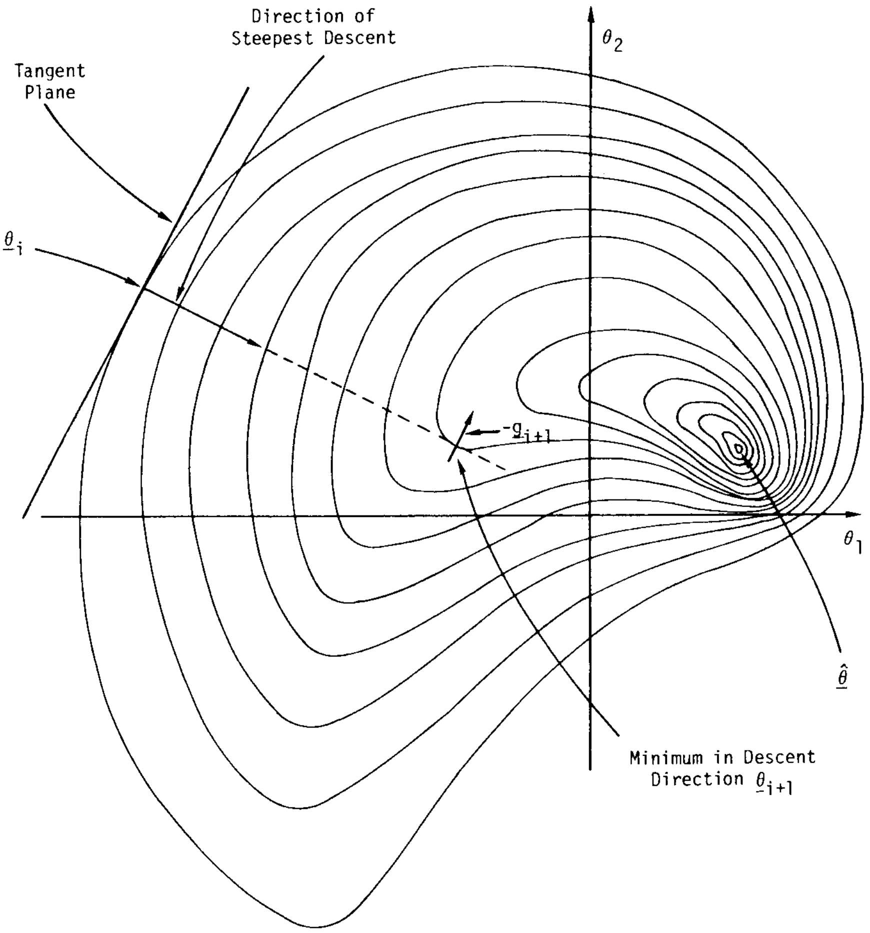

Direction of Steepest Descent
Tangent Plane
θ_2
$\underline{\theta}_i$
$-\underline{g}_{i+1}$
θ_1
$\hat{\underline{\theta}}$
Minimum in Descent Direction $\underline{\theta}_{i+1}$

Fig.2.13. Contours of $S(\theta)$ in parameter space.

$$\frac{\partial^2}{\partial\theta^2}S(t_c)\tau = -\left(\frac{\partial}{\partial\theta}S(t_c)\right)^T \qquad \text{Eq. 2.81}$$

Example 2.7, continued. Since $\frac{\partial^2}{\partial\theta^2}S(\theta) = Q$ Eq. 2.81 results in $Q\tau = -Qt_i$. Thus $\tau = -t_i$ and $t_{i+1} = 0$.□

When the cost function S is quadratic, then the Newton method guarantees that the absolute minimum is reached in one step.

In the following we will assume that $V = I$ so that

$$\left(\frac{\partial}{\partial\theta}S(t_c)\right)^T = -2\left(\frac{\partial}{\partial\theta}g(t_c)\right)^T(\underline{y}-g(t_c)) = -2\sum_{i=1}^{n}\left(\frac{\partial}{\partial\theta}g_i(t_c)\right)^T(\underline{y}_i-g_i(t_c)) \qquad \text{Eq. 2.82}$$

$$\frac{1}{2}\frac{\partial^2}{\partial\theta^2}S(t_c) = \left(\frac{\partial}{\partial\theta}g(t_c)\right)^T\frac{\partial}{\partial\theta}g(t_c) - \sum_{i=1}^{n}\left(\frac{\partial^2}{\partial\theta^2}g_i(t_c)\right)(\underline{y}_i - g_i(t_c))$$

$$= J^TJ - \sum_{i=1}^{n} H_i(\underline{y}_i - g_i(t_c)) \qquad \text{Eq. 2.83}$$

where J and H_i denote, respectively, the Jacobian matrix of $g(t_c)$ and the Hessian matrix of $g_i(t_c)$. When the residuals are small then the second term $\sum_i H_i(\underline{y}_i - g_i(t_c))$ can be neglected which renders Eq. 2.81 into

$$J^TJ\tau = J^T(\underline{y} - g(t_c)) = -\frac{1}{2}\mathrm{grad}_\theta S(t_c) \qquad \text{Eq. 2.84}$$

This is called the Gauss-Newton method. Eq. 2.84 reminds us of ordinary least squares where we had $X^TX\underline{\hat{\theta}} = X^T\underline{y}$. When J^TJ is regular, the Gauss-Newton step is given by

$$\Delta t_{GN} = (J^TJ)^{-1}J^T(\underline{y} - g(t_c)) = -\frac{1}{2}(J^TJ)^{-1}\mathrm{grad}_\theta S(t_c) \qquad \text{Eq. 2.85}$$

Since the Gauss-Newton method is based upon the Taylor expansion Eq. 2.80 the method converges rapidly close to the minimum, whereas far from the minimum the method diverges as a rule. Far from the minimum the direction of the Gauss-Newton step is usually very close to the direction of the contour. When J^TJ is singular the Gauss-Newton step is no longer unique, a problem which is solved by the next method.

The steepest descend and the Gauss-Newton method are combined into the Marquardt method, where a search direction is determined from

$$(J^TJ + \lambda I)\tau = -\frac{1}{2}\mathrm{grad}_\theta S(t_c) \qquad \text{Eq. 2.86}$$

When $\lambda = 0$ we get Eq. 2.84, whereas with very large λ the second term on the left-hand side dominates, and τ is in the steepest descend direction. Note that the addition of λI also removes a possible rank deficiency of J^TJ. In the Marquardt iteration scheme λ is kept as small as possible to exploit the rapid convergence of the Gauss-Newton method. Furthermore, still a step size needs to be determined. The Marquardt method, which works very well in practice, is included in scientific program libraries like IMSL.

For illustrative purposes we return to the Gauss-Newton method which we now consider from the viewpoint of the nonlinear function. Thus now we make a Taylor expansion of $g(x_i|\theta)$:

$$g(x_i|t) \approx g(x_i|t_c) + \left(\frac{\partial}{\partial\theta}g(x_i|t_c)\right)(t-t_c) \qquad g(t) \approx g(t_c) + J(t-t_c) \qquad \text{Eq. 2.87}$$

Abbreviating $t - t_c = \tau$ we now have a linearized problem

$$\underline{y} - g(t_c) = J\tau + \underline{\nu} \qquad \text{Eq. 2.88}$$

where the Jacobian matrix *J* has *nobs* rows and *npar* columns. We thus have to minimize

$$S(t) = (\underline{y} - g(t_c) - J\tau)^T(\underline{y} - g(t_c) - J\tau) \qquad \text{Eq. 2.89}$$

which, when *J* is of full rank, possesses solution (see Eq. 2.42)

$$\hat{\underline{\tau}} = (J^TJ)^{-1}J^T(\underline{y} - g(t_c)) \qquad \text{Eq. 2.90}$$

which is equal to Eq. 2.85. The default algorithm used by the function *nls* in *R* is the Gauss-Newton method, where Eq. 2.90 is applied iteratively, possibly estimating a step size to scale and find the optimal $\hat{\underline{\tau}}$ in each step, until some convergence criterion is met.

Practical nonlinear regression

To check whether the minimum found by the chosen iterative minimization method is global, one can take a few different starting points. However, in general there is no guarantee that the outcome is a global minimum. The stages of a nonlinear regression analysis consist of the following:

(a) Use the data *y*, the control settings x_i, and the expectation function $g(x_i|\theta)$ to obtain starting estimates for the parameters θ

(b) Use the information from (a) in an iterative nonlinear estimation computer program to obtain the least squares estimates $\hat{\theta}$ and to produce linear approximation summary statistics.

(c) Investigate the fitted model for adequacy of fit and for sensibleness of the parameter estimates as in linear regression.

(d) Determine the adequacy of the approximation used for the summary statistics.

Summary statistics include the least squares parameter estimates $S(\hat{\theta})$, $\hat{\sigma}^2$ and its degrees of freedom $n - q$ ($\hat{\underline{\sigma}}^2 = \underline{S}(\hat{\underline{\theta}})/(n-q)$), and the approximate parameter estimators' covariance

(inverting Eq. 2.75)

$$\underline{\hat{D}}(\hat{\underline{\theta}}) = \hat{\underline{\sigma}}^2 \left\{ \left(\frac{\partial g}{\partial \theta}\right)^T \left(\frac{\partial g}{\partial \theta}\right) \right\}^{-1}_{\theta = \hat{\theta}} \qquad \text{Eq. 2.91}$$

Minimization routines often return the *R*-matrix of the QR decomposition of the Jacobian $\frac{\partial g}{\partial \theta}(\hat{\underline{\theta}})$. Then $\underline{\hat{D}}(\hat{\underline{\theta}})$ is found from Eq. 2.52.

The covariance $\underline{\hat{D}}(\hat{\underline{\theta}})$ can be used to determine linear approximation joint and marginal parameter inference regions using the methods of linear regression, see Confidence intervals and confidence regions on page 27. These confidence limits are based upon linearization of $g(\theta)$ around $\hat{\theta}$, and thus can only be approximate.

Example 2.8. A realization of a simulation of an exponential decay according to the model

$$\underline{y}_i = a\exp(-kx_i) + \underline{\nu}_i \qquad \text{Eq. 2.92}$$

with parameters for amplitude *a* and decay rate *k* is depicted in Fig.2.14.A.

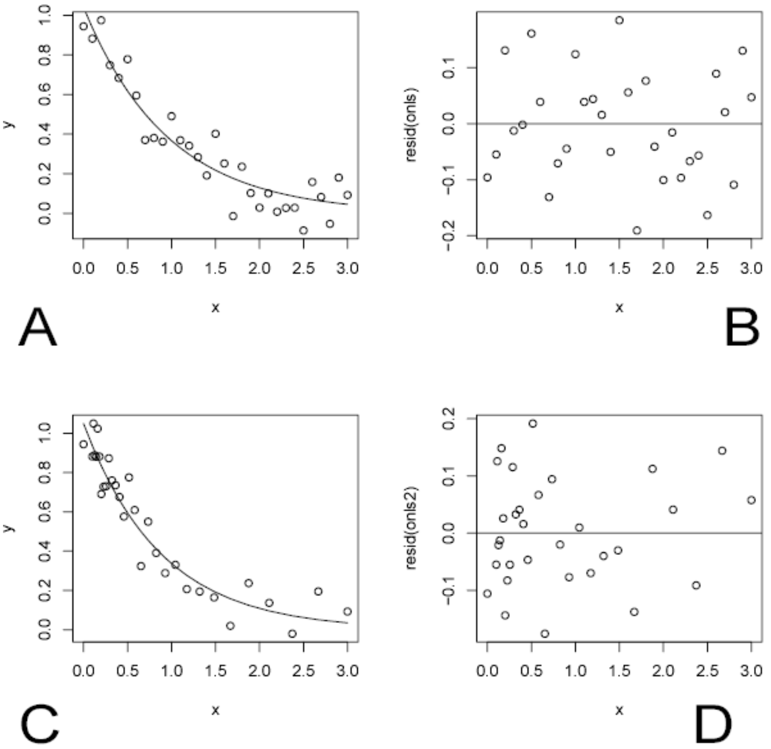


Fig.2.14. Simulation of exponential decay with a=1 and k=1. fit (A,C) and residuals (B,D). Two different experimental designs are compared: with equidistant time points (A), and with time points logarithmically equidistant between 0.1 and 3 (C).

Two different experimental designs are compared: with equidistant time points (A), and with logarithmically equidistant time points (C).

Analogous to the linear model, cf. Fig.2.9., we can now (for each design, and for each choice of the two parameters and noise level) repeat a simulation 100 times, investigate the statistical properties of these estimates, and compare them with the theory. A large asymmetry of *k*-histogram, Fig.2.15.B, is visible with a $k = 8$ simulation (and equidistant time points). The scatterplot, Fig.2.15.A, also shows these large *k*-estimates. The linear approximation confidence region is not very adequate in this case. This comes as no surprise, since very little information is present on such a fast decay with a time step of 0.1.□

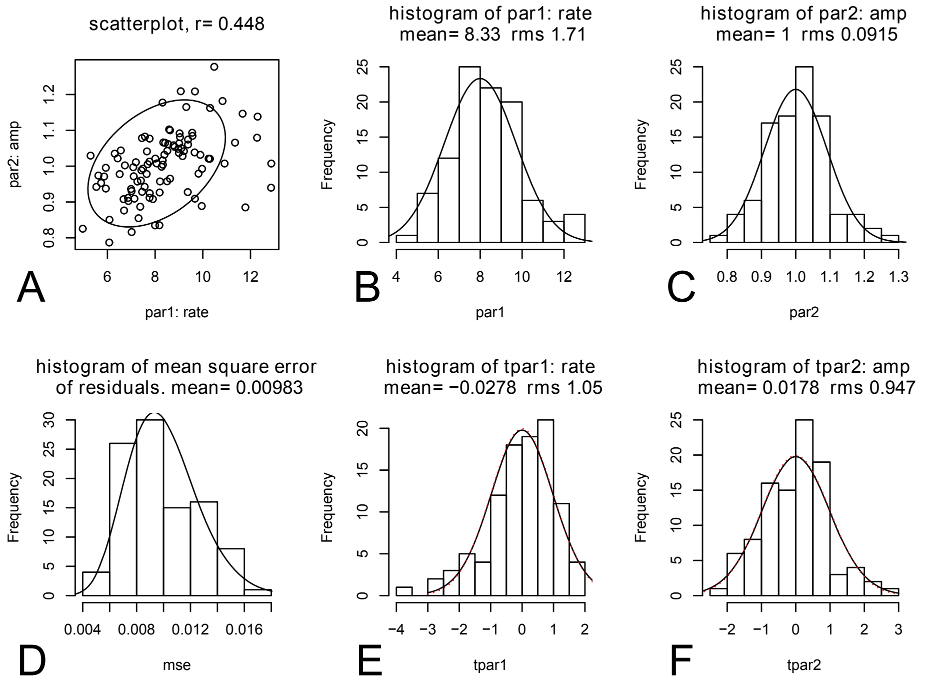


Fig.2.15. Scatterplot of all hundred estimates $(\hat{k}, \hat{a})_j$ for the decay rate and amplitude paramaters of the exponential decay (A), histogram of decay rate estimates (B) and amplitude estimates (C). Histograms of mean square error (D), and of the deviation from the true value divided by the estimated standard deviation, $(\hat{\theta} - \theta_{\text{true}})/\hat{\sigma}_{\hat{\theta}}$, of decay rate (E) and amplitude estimates (F). In addition, **linear approximation** theoretical results have been added: (A) the 90% confidence ellipse according to Eq. 2.67, centered around the estimate closest to the true values, (B) and (C) normal distributions according to Eq. 2.57, centered at the true values with standard deviation equal to the rms width of the histogram, (D) $(\sigma^2\chi^2_{df})/df$ according to Eq. 2.62, (E) and (F) t_{df}-distributions according to Eq. 2.58 (solid lines) and standard normal distributions (dotted). Note that here $df = 29$ and $k = 8$.

Appendix 2.1 Matrix fundamentals

Vectors and matrices are represented by, respectively, lower case and upper case characters, if possible italic. Underlining of characters denotes stochastic variables. A circumflex or hat ($\hat{a}$) denotes estimator. A^T is the transpose of A. $A^\dagger$ is the Moore-Penrose generalized inverse of A. Let *f* be a scalar function of *x*, then the gradient of *f* with respect to *x* is defined by the row vector $\frac{\partial}{\partial x}f(x) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots\right)$. The gradient vector *g* is defined as $g = \left(\frac{\partial f}{\partial x}\right)^T$.
The second derivative with respect to *x* is a matrix which is called the Hessian and is defined by $\frac{\partial^2}{\partial x^2}f(x) = \frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}f(x)\right)^T = H \qquad H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$. Let *x* and *y* be $m \times 1$ and $n \times 1$ column vectors, then the derivative of *y* with respect to *x* is an $n \times m$ matrix called the Jacobian and is defined by $\frac{\partial y}{\partial x} = J \qquad J_{ij} = \frac{\partial y_i}{\partial x_j}$. Thus the Jacobian of the gradient is the Hessian.
The quadratic form $q(x)$ is defined as $q(x) = x^T Q x$. Now since $Q = \frac{Q+Q^T}{2} + \frac{Q-Q^T}{2}$ and the latter part does not contribute to $q(x)$ we will further assume that *Q* is symmetric. Quadratic forms are classified according to their sign. If $x^T Q x > 0$ for all vectors *x*, then *Q* is positive definite. (≥ 0 positive semidefinite; < 0 negative definite; ≤ 0 negative semidefinite; sign indefinite). We mention a few useful identities:

$$\frac{\partial}{\partial x}c^T x = c^T \qquad \frac{\partial}{\partial x}x^T Q x = 2x^T Q \qquad \frac{\partial^2}{\partial x^2}x^T Q x = 2Q \qquad \text{Eq. 2.93}$$

The directional derivative of a scalar function *f* at x_0 in the direction *y* ($y^T y = 1$) is defined as

$$\lim_{t \to 0}\frac{f(x_0 + ty) - f(x_0)}{t} = \frac{\partial}{\partial x}f(x_0)y = g^T(x_0)y \qquad \text{Eq. 2.94}$$

It is easily proved that this directional derivative is maximized by the gradient direction *g*. By the Schwarz inequality $(g^T y)^2 \leq g^T g y^T y = g^T g$.
A second-order Taylor series expansion of a scalar-valued function of more than one variable $f(x)$ around $x = a$ can be written as (see e.g. http://en.wikipedia.org/wiki/Taylor_series)

$$f(x) = f(a) + \frac{\partial}{\partial x}f(x)(x-a) + \frac{1}{2!}(x-a)^T H(x-a) + \ldots \qquad \text{Eq. 2.95}$$

where the gradient and Hessian have been defined above, and are to be evaluated at $x = a$.

Appendix 2.2 Orthogonality

Two functions $f(x)$ and $g(x)$ are orthogonal in an interval (x_{min}, x_{max}) when the integral of their product equals zero:

$$\int_{x_{min}}^{x_{max}} f(x)g(x)dx = 0 \qquad \text{Eq. 2.96}$$

When $x_{min} = -x_{max}$ then odd and even functions are orthogonal (recall that an even function has the property $f(x) = f(-x)$, whereas with an odd function $f(x) = -f(-x)$).
Two vectors f and g of length n are orthogonal when their inner product equals zero:

$$f \cdot g = \sum_{i=1}^{n} f_i g_i = 0 \qquad \text{Eq. 2.97}$$

An application of this is the straight line fit. When the two columns of the X matrix are orthogonal, $\sum_{i=1}^{n} 1x_i = 0$, the off diagonal elements of $X^T X$ are zero, and thus also the off diagonal elements of the autocovariance matrix $\sigma^2 (X^T X)^{-1}$ are zero. This then implies that the estimates of the intercept and slope parameters are uncorrelated. In general, orthogonality of the columns of the X matrix implies that the estimated parameters that correspond to those columns are uncorrelated. An application of this is the R-function *poly()* which computes orthogonal polynomials that can be used to describe a measurement. These orthogonal polynomials are obtained from the monomials $1, x, x^2, \ldots$ by the Gram-Schmidt process.

Appendix 2.3 Matrix decompositions

Let A be a real symmetric $n \times n$ matrix then its eigenvalue decomposition reads

$$A = V \Lambda V^T \qquad \text{Eq. 2.98}$$

with V an orthogonal matrix ($V^{-1} = V^T$) and $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$. When all $\lambda_i \neq 0$ ($\text{rank}A = n$) then $A^{-1} = V \Lambda^{-1} V^T$ where $\Lambda^{-1} = \text{diag}(\lambda_1^{-1}, \ldots, \lambda_n^{-1})$.
The trace of A is equal to the sum of its eigenvalues:

$$\text{trace}A = \text{trace}V \Lambda V^T = \text{trace}V^T V \Lambda = \text{trace}\Lambda = \sum_{i=1}^{n} \lambda_i \qquad \text{Eq. 2.99}$$

where we used

$$\mathrm{trace}AB = \sum_{i,j} A_{ij}B_{ji} = \mathrm{trace}BA \qquad \text{Eq. 2.100}$$

The determinant of A is equal to the product of its eigenvalues

$$\det A = \det V\Lambda V^T = \det V^T V\Lambda = \det\Lambda = \prod_{i=1}^{n} \lambda_i \qquad \text{Eq. 2.101}$$

If A is positive definite then the quadratic form $x^TAx > 0$ which implies

$$x^TAx = x^TV\Lambda V^Tx = \sum_{i=1}^{n} \lambda_i (x^Tv_i)^2 > 0 \forall x \qquad \text{Eq. 2.102}$$

Thus A positive definite implies $\lambda_i > 0 \forall i$. Then A can be written as

$$A = (V\Lambda^{1/2}V^T)^2 = S^2 \qquad \text{Eq. 2.103}$$

which is the square root decomposition of A: $A^{1/2} = S$.

Every regular $n \times n$ matrix A can be decomposed into

$$A = LDF = LU \qquad \text{Eq. 2.104}$$

where L and F are, respectively, lower and upper unit triangular matrices, U is an upper triangular matrix and D is a diagonal matrix. The second factorization in Eq. 2.104 is called the LU decomposition. Now if A is symmetric we have $A = LDF = F^TDL^T = A^T$, thus $L = F^T$. If furthermore A is positive definite we get the Cholesky decomposition

$$A = LD^{1/2}D^{1/2}L^T = GG^T \qquad \text{Eq. 2.105}$$

with G a lower triangular matrix. The Cholesky decomposition is useful to invert a positive definite matrix with a large condition number. When $\kappa(A)$ is the condition number of A (the ratio of the largest and smallest eigenvalue of A) then $\kappa(G) = (\kappa(A))^{1/2}$ and $A^{-1} = (G^T)^{-1}G^{-1}$.

Every $n \times m$ matrix A ($n \geq m$) can be decomposed into

$$A = QR \qquad \text{Eq. 2.106}$$

where Q is an $n \times m$ matrix with orthogonal columns and R is an upper triangular matrix. Eq. 2.106 is termed the “skinny” QR factorization.

When A is square and nonsingular then the QR-factorization of A gives for $B = A^TA = (QR)^TQR = R^TQ^TQR = R^TR$ which is the Cholesky-decomposition of B.

For completeness we also define here the full *QR* factorization:

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} \qquad \text{Eq. 2.107}$$

Here Q_1 and *R* are the *Q* and *R* of Eq. 2.106, whereas Q_2 is an $n \times (n - m)$ matrix such that $\begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ is an $n \times n$ orthogonal matrix.

Appendix 2.4 The Singular Value Decomposition

Next to the QR decomposition, SVD is a reliable method to compute the Moore-Penrose generalized inverse. Furthermore the method provides a means to decompose a matrix containing multivariate data into orthogonal pieces of information. SVD can be viewed as a natural generalization to arbitrary matrices of the eigenvector / eigenvalue decomposition of normal matrices (when $BB^T = B^TB$ then *B* is called normal). We will only consider real matrices, for complex matrices the Hermitian adjoint ($B^* = \bar{B}^T$) has to be substituted for the transpose B^T.

Theorem 2.6. Singular Value Decomposition (SVD).

Let *A* be an $m \times n$ matrix with $\operatorname{rank}A = r \le m \le n$. There exist orthogonal $m \times m$ and $n \times n$ matrices *U* and *W* and an $m \times n$ matrix $\Sigma = [\sigma_{ij}]$ which is zero except for its diagonal elements $\sigma_{11} \ge \sigma_{22} \ge \ldots \ge \sigma_{rr} > \sigma_{r+1,r+1} = \ldots = \sigma_{mm} = 0$ such that *A* may be written in the form

$$A = U\Sigma W^T = \sum_{l=1}^{r} u_l \sigma_{ll} w_l^T \qquad \text{Eq. 2.108}$$

The numbers $\sigma_{ii} \equiv \sigma_i$ (known as the singular values) are the non-negative square roots of the eigenvalues of the normal matrix AA^T, and hence are uniquely determined. The columns of *U* and *W* are eigenvectors of, respectively, AA^T and A^TA, and are known as, respectively, the left and right singular vectors of *A*. If AA^T has distinct eigenvalues, then *U* is determined up to a right diagonal factor $D = \operatorname{diag}(d_1, \ldots, d_m)$ where $d_i = \pm 1$. The first *r* columns of *W* are uniquely determined, when *U* is given.

Proof: If A is factorized according to Eq. 2.108 then $AA^T = U\Sigma W^T W\Sigma^T U^T = U\text{diag}(\sigma_1^2, \ldots, \sigma_m^2)U^T$ which is the diagonalization of the normal matrix AA^T. If $U = [u_1 u_2 \ldots u_m]$ then $AA^T u_j = \sigma_j^2 u_j$. Because the singular values are to be non-negative and are to be arranged in non-increasing order Σ is uniquely determined by AA^T. Analogously $A^T A = W\Sigma^T U^T U\Sigma W^T = W\text{diag}(\sigma_1^2, \ldots, \sigma_m^2, 0, \ldots, 0_n)W^T$ is the diagonalization of the normal matrix $A^T A$. If $W = [w_1 w_2 \ldots w_n]$ then $A^T A w_j = \sigma_j^2 w_j$. If the σ_j^2 are distinct, then all the normalized eigenvectors u_j are determined up to a scalar factor ± 1. Eigenvectors corresponding to a degenerate eigenvalue are of course not uniquely determined, but an orthogonal matrix U can always be chosen. If U has been chosen, the first r eigenvectors w_j are uniquely determined by $w_j = \sigma_j^{-1}(A^T u_j)$. These w_j are orthonormal: $w_j^T w_l = \sigma_j^{-1}(A^T u_j)^T \sigma_l^{-1} A^T u_l = \sigma_j^{-1}\sigma_l^{-1} u_j^T A A^T u_l = \sigma_j^{-1}\sigma_l^{-1} u_j^T \sigma_l^2 u_l = \delta_{jl}$ if both j and $l \le r$. There exist $n - r$ additional (but not uniquely determined) orthonormal vectors $w_{r+1}, \ldots, w_n$ such that W is an orthogonal matrix. □

We will now discuss the relation between SVD and least squares. Recall Eq. 2.30, with $V = I$: $X^T X \underline{\hat{\theta}} = X^T \underline{y}$ and let the SVD of X be $X = U\Sigma W^T$, where the matrices U, Σ and W are, respectively, $n \times n$, $n \times (k+1)$ and $(k+1) \times (k+1)$. Substitution of the SVD into Eq. 2.30 gives

$$W\Sigma^T U^T U\Sigma W^T \underline{\hat{\theta}} = W\Sigma^T U^T \underline{y} \qquad \text{Eq. 2.109}$$

which reduces to

$$\Sigma^T \Sigma W^T \underline{\hat{\theta}} = \Sigma^T U^T \underline{y} \qquad \text{Eq. 2.110}$$

When $\text{rank}X = q$ then

$$\underline{\hat{\theta}} = W\Sigma^{\dagger} U^T \underline{y} = X^{\dagger} \underline{y} \qquad \text{Eq. 2.111}$$

where $\Sigma^{\dagger}$ is an $(k+1) \times n$ matrix which is zero except for its first q diagonal elements which are equal to $\sigma^{\dagger}_{ii} = \sigma_i^{-1}$. Thus also SVD provides a straightforward method to estimate $\underline{\hat{\theta}}$, using the Moore-Penrose generalized inverse computed from $X^{\dagger} = W\Sigma^{\dagger} U^T$. The SVD method is more flexible than the QR method when the matrix X is not of full rank. Note that only the first q singular vectors of U and W are involved in the estimation of $\hat{\underline{\theta}}$. In practice, when the rank of X is unknown, one can define an effective pseudoinverse where $\sigma^{\dagger}{}_{ii} = \sigma_i^{-1}$ if $\sigma_i > \tau$, where τ represents a tolerance that reflects the errors in the data. Thus τ defines q. In this way X is approximated by the first q singular vectors of U and $W,$ together with the first q singular values.

Appendix 2.5 Probability theory

Some definitions and theorems from probability theory which are used in parameter estimation are summarized below.

For an $n \times 1$ random vector $\underline{y}$ (NB the underscore indicates that y is stochastic) the probability distribution function $F_{\underline{y}}$ is defined as the probability that $\underline{y}_1 \le y_1$, $\underline{y}_2 \le y_2$, $\ldots, \underline{y}_n \le y_n$, in formula

$$F_{\underline{y}}(y) = P(\underline{y}_1 \le y_1, \underline{y}_2 \le y_2, \ldots, \underline{y}_n \le y_n) \qquad \text{Eq. 2.112}$$

Under reasonable conditions the probability distribution function possesses a derivative which is called the probability density function (PDF) $f_{\underline{y}}$ and which satisfies

$$F_{\underline{y}}(y) = \int_{-\infty}^{y_n} \ldots \int_{-\infty}^{y_2} \int_{-\infty}^{y_1} f_{\underline{y}}(\psi_1, \psi_2, \ldots, \psi_n) d\psi_1 d\psi_2 \ldots d\psi_n \qquad \text{Eq. 2.113}$$

$$f_{\underline{y}}(y) = \frac{\partial^n}{\partial y_1 \partial y_2 \ldots \partial y_n} F_{\underline{y}}(y) \qquad \text{Eq. 2.114}$$

Marginal distribution functions and marginal density functions result from letting one or more of the $y_i \to \infty$, e.g. the probability distribution function for $\underline{y}_1$ is given by

$$F_{\underline{y}}(y_1, \infty, \ldots, \infty) = F_{\underline{y}_1}(y_1) \qquad \text{Eq. 2.115}$$

Saying that two random variables $\underline{y}_1$ and $\underline{y}_2$ are independent means that their joint distribution function $F_{\underline{y}_1, \underline{y}_2}(y_1, y_2)$ is equal to the product $F_{\underline{y}_1}(y_1) F_{\underline{y}_2}(y_2)$ of the marginal distributions. Generalizing: n random variables $\underline{y}_1, \underline{y}_2, \ldots, \underline{y}_n$ with joint PDF $f_{\underline{y}}(y)$ and marginal PDF’s $f_{\underline{y}_i}(y_i)$ are independent if and only if

$$f_{\underline{y}}(y) = \prod_{i=1}^{n} f_{\underline{y}_i}(y_i) \qquad \text{Eq. 2.116}$$

The ensemble average of a function of random variables is defined in terms of the expectation operation. The expected value of a (vector) function *g* of $\underline{y}$ is defined as

$$E[g(\underline{y})] = \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} g(y_1, \ldots, y_n) f_{\underline{y}}(y) dy_1 \ldots dy_n \equiv \int g(y) f_{\underline{y}}(y) dy \qquad \text{Eq. 2.117}$$

Thus expectation is a linear operation on the function *g*.

$E[\underline{y}]$ is called the mean value and shall be denoted as μ (or μ_y). The (auto) covariance matrix $D(\underline{y})$ is the $n \times n$ matrix whose ij^{th} element is $E[(\underline{y}_i - \mu_i)(\underline{y}_j - \mu_j)]$. Note that the diagonal elements of $D(\underline{y})$ are the variances of the individual random variables: $D(\underline{y}_i) = E[(\underline{y}_i - \mu_i)^2] = E[\underline{y}_i^2] - \mu_i^2$. If two random vectors $\underline{y}$ and $\underline{z}$ are independent then their covariance matrix equals zero

$$D(\underline{y}, \underline{z}) = E[(\underline{y} - \mu_y)(\underline{z} - \mu_z)^T] = E[(\underline{y} - \mu_y)]E[(\underline{z} - \mu_z)^T] = 0 \qquad \text{Eq. 2.118}$$

Vectors satisfying Eq. 2.118 are said to be uncorrelated. Uncorrelated vectors are not necessarily independent. For normally distributed random vectors uncorrelated implies independence.

Sorenson (9, p.360) states that “The general question asked in estimation theory might be phrased as ‘What is the probability that a random vector $\underline{\theta}$ has value θ given the realization y of the random measurement vector $\underline{y}$’. Assuming the existence of the required joint PDF $f_{\underline{\theta}, \underline{y}}$ this question can be answered by determining the conditional probability of $\underline{\theta}$ given the measurement y.”. The conditional PDF of $\underline{\theta}$ given $\underline{y} = y$ is defined as

$$f_{\underline{\theta}|\underline{y}}(\theta|y) = \frac{f_{\underline{\theta}, \underline{y}}(\theta, y)}{f_{\underline{y}}(y)} \qquad \text{Eq. 2.119}$$

where by definition $f_{\underline{y}}(y) = \int f_{\underline{\theta}, \underline{y}}(\theta, y) d\theta$. Eq. 2.119 leads to Bayes’ rule. From Eq. 2.119 the conditional expectation of $\underline{\theta}$ given $\underline{y} = y$ is

$$E[\underline{\theta}|y] = \int \theta \frac{f_{\underline{\theta}, \underline{y}}(\theta, y)}{f_{\underline{y}}(y)} d\theta \qquad \text{Eq. 2.120}$$

Transformation of random variables

Suppose that g is a one to one mapping of R^n into R^n and that g^{-1} is the inverse mapping such that for $\underline{y} = g(\underline{x})$ we have $\underline{x} = g^{-1}(\underline{y})$. If $\underline{x}$ possesses PDF $f_{\underline{x}}$ then the PDF of $\underline{y}$ is

$$f_{\underline{y}}(y) = f_{\underline{x}}(g^{-1}(y))|J(y)| \qquad \text{Eq. 2.121}$$

where $|J(y)|$ is the absolute value of the determinant of the Jacobian

$$|J(y)| = \left|\det\frac{\partial}{\partial y}g^{-1}(y)\right| \qquad \text{Eq. 2.122}$$

We apply Eq. 2.121 to derive the general form of the multivariate normal distribution. Assume that $\underline{x}$ is a vector of n iid normally distributed random variables $N(0, 1)$ with PDF

$$f_{\underline{x}}(x) = (2\pi)^{-n/2}e^{-\frac{1}{2}x^Tx} \qquad \text{Eq. 2.123}$$

Consider the transformation $\underline{y} = A\underline{x} + \mu$ with A a regular matrix. Then $x^Tx = (A^{-1}(y-\mu))^T(A^{-1}(y-\mu)) = (y-\mu)^TA^{-T}A^{-1}(y-\mu)$ and the Jacobian is given by $\frac{\partial}{\partial y}(A^{-1}(y-\mu)) = A^{-1}$. Thus from Eq. 2.121 we find

$$f_{\underline{y}}(y) = (2\pi)^{-n/2}|\det A^{-1}|\exp\left(-\frac{1}{2}(y-\mu)^TA^{-T}A^{-1}(y-\mu)\right) \qquad \text{Eq. 2.124}$$

Now define $A^{-T}A^{-1} = V^{-1}$ then with $\det V^{-1} = (\det A^{-1})^2 > 0$ we find (compare Eq. 2.20)

$$f_{\underline{y}}(y) = ((2\pi)^n\det V)^{-1/2}\exp\left(-\frac{1}{2}(y-\mu)^TV^{-1}(y-\mu)\right) \qquad \text{Eq. 2.125}$$

References

1 Bard, Y. (1974) Nonlinear Parameter Estimation. Academic Press, New York.

2 Bates, D.M., and Watts, D.G. (1988) Nonlinear regression and its applications. Wiley, New York.

3 Berenson, M.L., Levine, D.M., Krehbiel, T.C. (2009) Basic business statistics. Concepts and applications. 11th ed. Pearson Int. Ed., Upper Saddle River, NJ.

4 Draper, N.R., and Smith, H. (1981) Applied Regression Analysis. Wiley, New York.

5 Koch, K.-R. (1988) Parameter estimation and hypothesis testing in linear models. Springer, Berlin.

6 Sachs, L. (1982) Applied Statistics. Springer, New York.

7 Searle, S.R. (1971) Linear Models. Wiley, New York.

8 Seber, G.A.F., and Wild, C.J. (1989) Nonlinear regression. Wiley, New York.

9 Sorenson, H.W. (1980) Parameter Estimation. Dekker, New York.

Solving a separable nonlinear least squares problem: Parameter estimation of time resolved spectra.

Ivo H.M. van Stokkum
Division of Physics and Astronomy, Vrije Universiteit, Amsterdam

Spectroscopical methods are used to collect information about the structure of molecules and atoms and about their concentrations. The measurement of the concentrations of different components as a function of time is necessary to study the dynamics of chemical reactions and of structural changes. However, with a mixture of n_{comp} components whose spectra overlap concentrations can only be measured indirectly by observing the total spectrum at several time instants, for example in time-resolved absorption (5, 8) and fluorescence (9) spectroscopy.

The basic superposition model is the following:

$$\psi_{t_i\lambda_j} = \sum_{l=1}^{n_{\text{comp}}} c_{lt_i}\varepsilon_{l\lambda_j} + \xi_{t_i\lambda_j} \qquad \text{Eq. 3.1}$$

$$\Psi = CE^T + \Xi \qquad \text{Eq. 3.2}$$

where the $n \times m$ matrix Ψ denotes the time resolved spectrum, measured at n time instants t_i and m wavelengths λ_j. c_{lt_i} and $\varepsilon_{l\lambda_j}$ denote, respectively, the concentration at time t_i and spectrum at wavelength λ_j of component *l*. $\xi_{t_i\lambda_j}$ denotes an independent and identically normally distributed stochastic disturbance with zero mean and variance ς^2. The c_{lt_i} and $\varepsilon_{l\lambda_j}$ are gathered in the matrices *C* and *E*, respectively $n \times n_{\text{comp}}$ and $m \times n_{\text{comp}}$. Matrix Ξ is, like Ψ, $n \times m$. Typical values for the number of components are $1 \le n_{\text{comp}} \le 5$, whereas the number of different wavelengths and the number of different time instants may vary from n_{comp} to 1000.

Measurement of Ψ poses the inverse problem: how can the spectroscopic and kinetic properties of the components be recovered. We assume that a kinetic model for C is known, which contains n_{par} unknown kinetic parameters gathered in θ, whereas the (linear) spectral parameters are unknown. Alternatively (9), a spectral model may be known which contains n_{par} unknown spectral parameters gathered in θ, with the (linear) concentration (or amplitude) parameters being unknown. Our aim is to estimate the unknown nonlinear and linear parameters. Here we present the parameter estimation problem starting from a kinetic model, the spectral case is treated analogously.

Since the spectral parameters, E^T, appear linearly in Eq. 3.2 we deal here with a separable nonlinear least squares problem (1-8). Exploiting this separability reduces the number of explicit parameters in the NLLS fit substantially, thus reducing the computational effort. Consider first the concentration matrix for fixed kinetic parameters θ. Then Eq. 3.2 represents a multivariate Gauss-Markoff model with solution:

$$\hat{E}^T(\theta) = C^{\dagger}(\theta)\Psi \qquad \text{Eq. 3.3}$$

where $C^{\dagger}(\theta)$ is the Moore-Penrose generalized inverse of $C(\theta)$. To calculate $C^{\dagger}(\theta)$ we follow Kaufman (4) and Golub and Leveque (2) and perform a *QR* decomposition of $C(\theta)$:

$$C(\theta) = \begin{bmatrix} Q_1(\theta) \ Q_2(\theta) \end{bmatrix} \begin{bmatrix} R(\theta) \\ 0 \end{bmatrix} \qquad \text{Eq. 3.4}$$

where $Q_1(\theta)$ and $Q_2(\theta)$ are, respectively, $n \times n_{\text{comp}}$ and $n \times (n - n_{\text{comp}})$ matrices which together form the orthogonal matrix $Q(\theta)$. $R(\theta)$ represents an $n_{\text{comp}} \times n_{\text{comp}}$ upper triangular matrix. Combining Eq. 3.3 and Eq. 3.4 we have:

$$\hat{E}^T(\theta) = R^{-1}(\theta)Q_1^T(\theta)\Psi \qquad \text{Eq. 3.5}$$

Using Eq. 3.4 and Eq. 3.5 we find for the residual matrix Z:

$$Z(\theta) = \Psi - C(\theta)\hat{E}^T(\theta) = (I - Q_1(\theta)Q_1^T(\theta))\Psi = Q_2(\theta)Q_2^T(\theta)\Psi \qquad \text{Eq. 3.6}$$

where $Q_2(\theta)Q_2^T(\theta)$ is an orthogonal projection matrix. The sum of squares to be minimized as a function of the unknown parameters θ is given by

$$S(\theta) = \text{trace}(Z^T(\theta)Z(\theta)) = \text{trace}(\Psi^T Q_2(\theta)Q_2^T(\theta)\Psi) \qquad \text{Eq. 3.7}$$

Thus elimination of the $m \times n_{\text{comp}}$ linear parameters results in the $(n - n_{\text{comp}}) \times m$ residual matrix $Q_2^T(\theta)\Psi$ which leaves as degrees of freedom: $\text{df} = (n - n_{\text{comp}}) \times m - n_{\text{par}}$. Numerical minimization of $S(\theta)$ in Eq. 3.7 requires the derivative $\frac{\partial}{\partial\theta}Q_2^T(\theta)\Psi$. Following again Kaufman (4) and Golub and Leveque (2) we recall from Eq. 3.4 that $Q_2^T(\theta)C(\theta) = 0$ and thus

$$0 = \frac{\partial}{\partial\theta}(Q_2^T(\theta)C(\theta)) = \left(\frac{\partial}{\partial\theta}Q_2^T(\theta)\right)C(\theta) + Q_2^T(\theta)\left(\frac{\partial}{\partial\theta}C(\theta)\right) \qquad \text{Eq. 3.8}$$

This leads to the approximation $\frac{\partial}{\partial\theta}Q_2^T(\theta) \approx -Q_2^T(\theta)\left(\frac{\partial}{\partial\theta}C(\theta)\right)C^{\dagger}(\theta)$ which combined with Eq. 3.3 gives

$$\frac{\partial}{\partial\theta}Q_2^T(\theta)\Psi \approx -Q_2^T(\theta)\left(\frac{\partial}{\partial\theta}C(\theta)\right)\hat{E}^T(\theta) \qquad \text{Eq. 3.9}$$

We estimate the variance ς^2 from $\hat{\varsigma}^2 = S(\hat{\theta})/\text{df}$. The usual linear approximation summary statistics are based upon a Taylor expansion of the criterion function $S(\theta)$ around the maximum likelihood estimates $\hat{\theta}$ and $\text{vec}(\hat{E}^T)$.

We simulated a model with two components having $c_l(t) = e^{-k_l t}$. We chose overlapping spectra ε_1 and ε_2 of Gaussian shape (cf. the solid lines in Fig.2.). The rate constants differed by only 10%: $k_1 = 0.05 \text{ ns}^{-1}$ and $k_2 = 0.055 \text{ ns}^{-1}$. We added noise with $\varsigma = 0.001\Psi_{\max} = 0.534$. The simulated time resolved spectrum Ψ is shown in Fig.1. Note

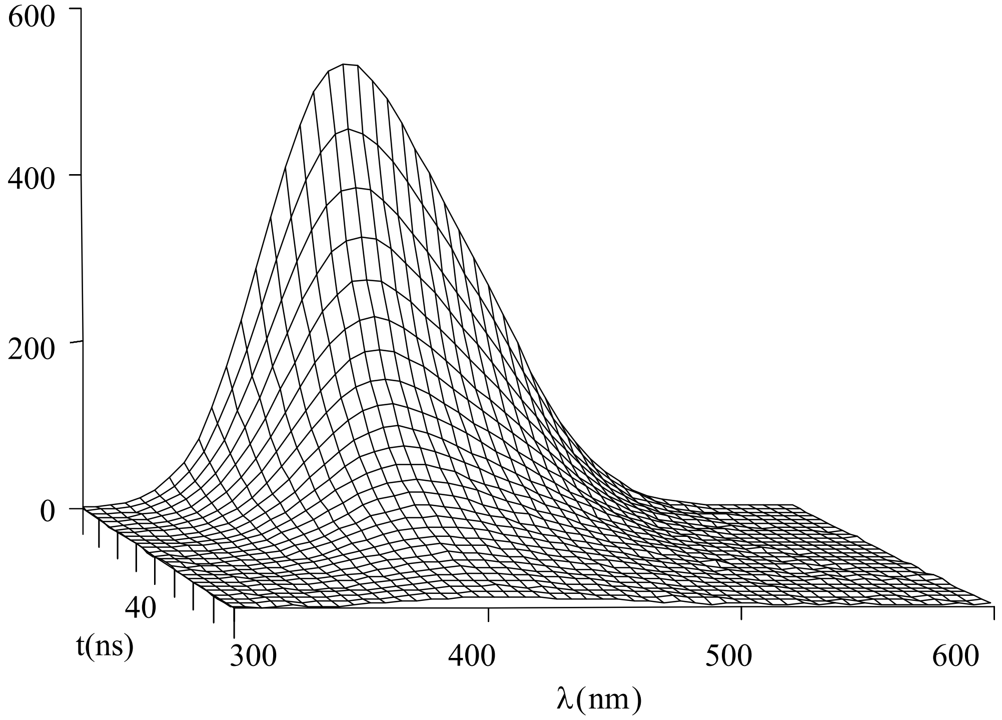


Fig.1. Simulated time resolved spectrum $\psi(t, \lambda)$ with $n = 81$ time instants between 0 and 80 nanoseconds and $m = 50$ wavelengths between 300 and 600 nanometer.

that we eliminate in this example $2m = 100$ parameters, arriving at $n_{\text{par}} = 2$. The NLLS fit resulted in $\hat{k}_1 = 0.0502(4)$, $\hat{k}_2 = 0.0549(6)$, $\hat{\varsigma} = 0.536(7)$ and the Studentized residuals behaved well. Comparing also the estimated spectral parameters (dashed lines) with their simulated values (solid lines) in Fig.2. we conclude that the fit was excellent.

References

1 Bates, D.M., and Lindstrom, M.J. (1986) Nonlinear least squares with conditionally linear parameters. In: Proc. of the Statistical Computing Section, American Statistical Association, New York, pp. 152-157.

2 Golub, G.H., and LeVeque, R.J. (1979) Extensions and uses of the variable projection algorithm for solving nonlinear least squares problems. Proc. of the 1979 Army Numerical Analysis and Comp. Conf., ARO Report 79-3, 1-12.

3 Golub, G.H., and Pereyra, V. (1973) The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. SIAM J. Numer. Anal. 10, 413-432.

4 Kaufman, L. (1975) A variable projection method for solving separable nonlinear least squares problems.

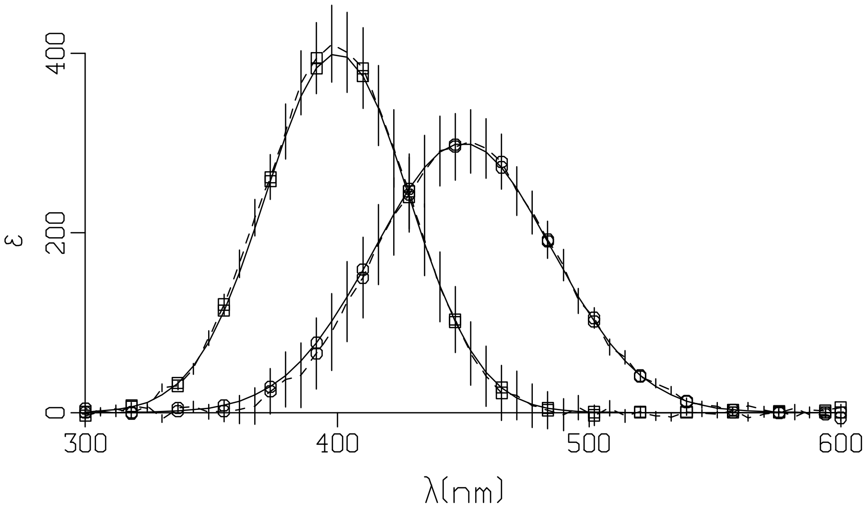


Fig.2. Simulated (solid lines) and estimated (dashed lines) spectral parameters $\varepsilon_1(\lambda)$ (squares) and $\varepsilon_2(\lambda)$ (circles). The vertical lines indicate plus or minus one standard error of the estimated spectral parameters.

BIT 15, 49-57.

5 Nagle, J.F. (1991) Solving complex photocycle kinetics. Theory and direct method. Biophys. J. 59, 476-487.

6 Ruhe, A., and Wedin, P.Å. (1980) Algorithms for separable nonlinear least squares problems. SIAM Review 22, 318-337.

7 Seber, G.A.F., and Wild, C.J. (1989) Nonlinear regression. Wiley, New York.

8 Solar, S., Solar, W., and Getoff, N. (1983) A semi-linear optimization model for resolving fast processes. J. Chem. Soc., Faraday Trans. 2, 79, 123-135.

9 Van Stokkum, I.H.M., Scherer, T., Brouwer, A.M., and Verhoeven, J.W. *(*1994) Conformational dynamics of flexibly and semirigidly bridged electron donor-acceptor systems as revealed by spectrotemporal parameterization of fluorescence. J. Phys. Chem. 98*,* 852-866.

Graphical output from a kinetic model

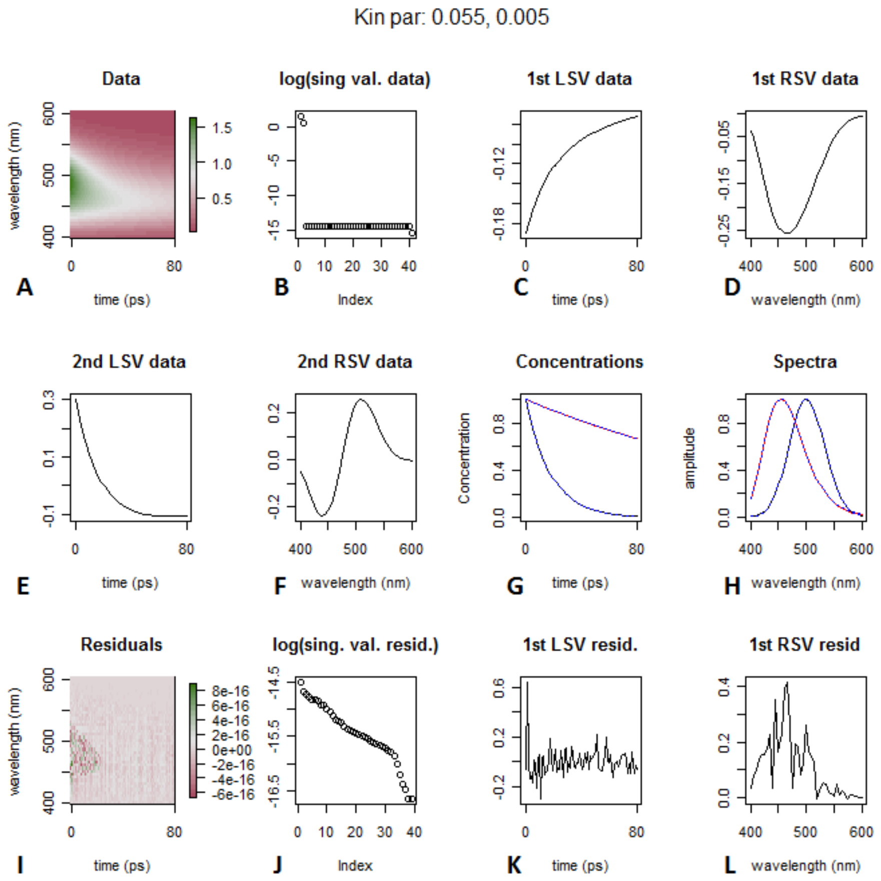


Fig.3.1. The above 12 panels summarize the analysis of default simulated data with zero additive noise. Since the data depend upon both time and wavelength, it is natural to use the actual time points and wavelengths of the measurement as abscissa and ordinate coordinates, respectively (see the Data and Residuals in panel A, I, respectively). The actual time points are also used as abscissa coordinates when plotting a concentration profile (panel G) or Left Singular Vector (LSV, panel C,E,K). Likewise, the actual wavelengths are used as abscissa coordinates when plotting a spectrum (panel H) or Right Singular Vector (RSV, panel D,F,L). Finally, the logarithm of the singular values of the matrix of Data is depicted in panel B, with the index as abscissa. Since a model with two linearly independent components was used, the rank of the data matrix was two, which is visible as the first two singular values differing significantly from the “noise level”. The estimated rate constants (Kinetic Parameters) are written in the title. After the fit the SVD of the matrix of residuals (panel I) is summarized in panels J,K,L. Both the first LSV and RSV are unstructured, indicating a satisfactory fit. The logarithm of the singular values of the residual matrix decreases from -14.5 to -16.5, which agrees with the machine error of about 10^{-16}. Black and red solid lines in Panel G correspond to the fitted concentrations of the first and second component. With this kinetic model they correspond to exp(-kt), with k the estimated Kinetic Parameters. The simulated concentrations are indicated by dashed blue lines. Black and red solid lines in Panel H correspond to the estimated spectra of the first and second component. Here, they perfectly overlap the simulated spectra indicated by dashed blue lines.

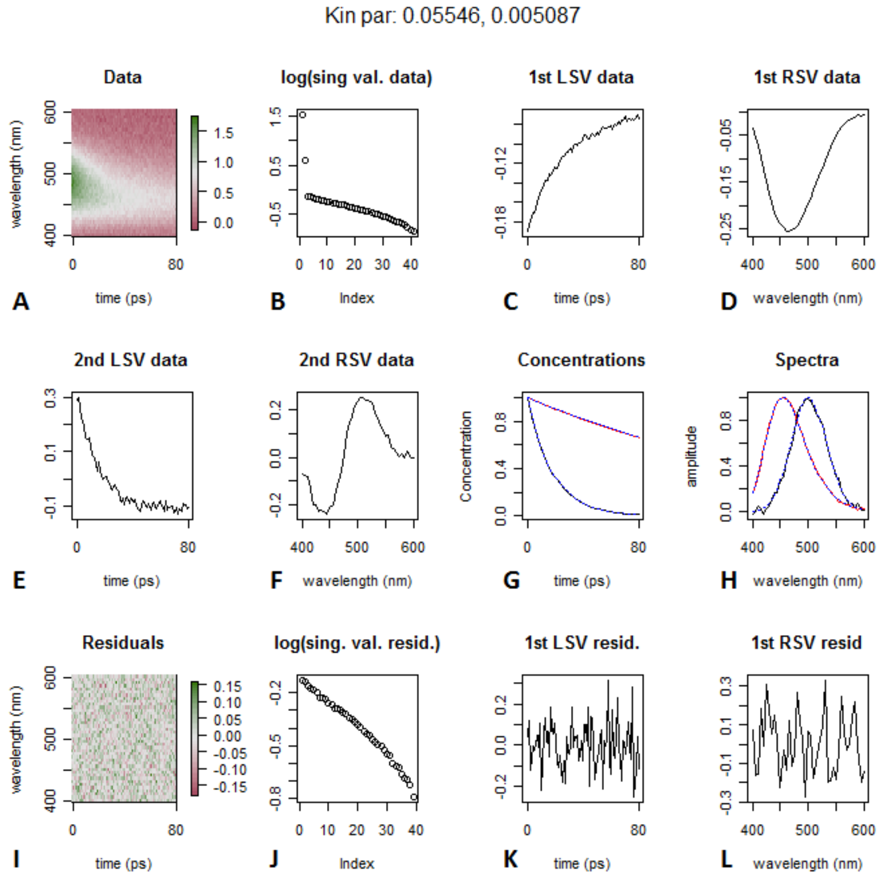


Fig.3.2. The above 12 panels summarize the analysis of default simulated data with additive noise of standard deviation 0.05. Note in panel B the huge increase of the singular values starting from index 3. A small amount of noise is mixed in with the 1st and 2nd LSV and RSV of the data (panels C-F). After the fit the SVD of the matrix of residuals (panel I) is summarized in panels J,K,L. Both the left and right first Singular Vector (LSV and RSV, panels K,L) are unstructured, as are the residuals (panel I), indicating a satisfactory fit. The logarithm of the singular values of the residual matrix decreases from -0.1 until -0.8. In a complicated way they are determined by the 0.05 standard deviation of the additive noise. The estimated spectral parameters ε_1 and ε_2 (*nl* parameters for each spectrum) are plotted as black and red solid lines in panel H. Compared to the simulated spectra (dashed blue curves), small deviations are visible. Black and red solid lines in Panel G correspond to the fitted concentrations of the first and second component, c_1 and c_2. Compared to the simulated concentrations (dashed blue curves), very small deviations are visible. These are due to the small deviations of the estimated Kinetic Parameters k_1 and k_2 (Kin par in title) from the true values (that were used in the simulation) of 0.055 and 0.005. For further information on the fit inspect Diagnostics tab of the Results.

Parameter estimation using R

Introduction

The aim of this exercise is to learn how to use ***R*** when solving a complicated parameter estimation problem. You are supposed to work individually.

You can download R on your own laptop or PC from http://cran.r-project.org/. Among the links on this page is the R home page http://www.r-project.org/

You can use a previously saved workspace, which has the extension .RData, and which contains both data and function objects. To start with a fresh .RData file, go to the bin folder.

The functions for the exercise are contained in ***Modelling.RData***

Start R by clicking on a .RData file. Look e.g. at the regression demo by typing

```
library(tcltk)
demo(tkcanvas)
```

in the command window, play with some of the points, and use the *help* function. Check out the R project home page and the on-line manuals under Help. For example:

```
help(dev.new)
```

gives information on handling multiple graphics devices. In particular

```
dev.new()
```

creates a new (active) graphics device, in which you can plot. This facilitates comparison of plots, however at the price of screen clutter. You can switch back to the previous graphics device:

```
dev.set(dev.prev())
```

If you want to save your graphics window, you can right click in the window, and select Print, and then print to a pdf file (if a pdf writer exists). If you do not have a pdf writer (at home), then you can use R's pdf function. You have to specify a unique name (as a string, between double quotes), e.g.

```
pdf(file="rate8.pdf")
simexpcorr(lellipse=T,rate=8)
dev.off(dev.cur())
```

After the pdf call, the active window shifts to the new pdf file. After you have called your function the pdf file is written, and you see no change in the now Inactive graphics window on your screen. Then you must turn off the pdf device, and the graphics window on your screen becomes Active again.

You can later manipulate your pdf file (e.g. zoom for closer inspection), or add text using a Typewriter (available in the Tools menu of Acrobat Pro).

To edit a (new) function named 'func' use the command

```
fix(func)
```

On the laptop or PC you will automatically get the R editor, which is similar to the *Notepad* editor. To attach a directory or a list or a dataframe (called *name*) to be searched use:

```
attach ("name")
```

Note that the full path name of a directory should be used.

To view the search list use the function

```
search()
```

You can also put attachments in a function named .First which you create with

```
fix(.First)
```

This function is executed when you start R. To see an object type its name, thus to see the current .First function type:

```
.First
```

History

With history(max=100) or with the arrow buttons you can see previous commands.

Example of a session

Executing this section is optional, but reading it is highly recommended. It is summarized by the function (then *kdata.txt* must be in the same folder as .Rdata)

```
examplenls()
```

The aim of this session is to fit some data using a non-linear regression model. Text in *italic* font represents output from R. The data are fitted with the model function:

$$k\,(T, K_0, A, E) = K_0 + A\,e^{\frac{-1000E}{T}}$$

The data are in matrix form (8 times 2) in a file which is read. The left column contains eight temperatures, the right column contains ‘measured’ *k*-values:

300	0.0885
290	0.0761
280	0.0587
270	0.0456
260	0.0387
250	0.0338
240	0.0330
230	0.0299

With the ‘scan’ function you can read data from file:

```
dataset <- scan(“kdata.txt”,what=list(t=0,k=0), n=16)
```

the optional ‘what’ adds a meaning to the data in the two columns (a name). The symbol ‘<-’ represents the assignment operator.

Typing the name of an object corresponds to printing the object. So:

```
dataset
$t:
[1] 300 290 280 270 260 250 240 230

$k:
[1] 0.0885 0.0761 0.0587 0.0456 0.0387 0.0338 0.0330 0.0299
```

gives the contents of the object ‘dataset’, in this case a matrix. Before we can fit a conversion to a so-called data frame is necessary

```
dataframe <- data.frame(dataset)
attributes(dataframe)
$names:
[1] "t" "k"

$row.names:
[1] "1" "2" "3" "4" "5" "6" "7" "8"

$class:
[1] "data.frame"
```

gives the attributes of the data frame. Other useful functions are *length, mode* and *dim*.

Alternatively you can use the function read.table, see

```
help(read.table)
```

We now define our fit function:

```
fitfunc <- function(TEMP,K0,A,E)
            { K0+A*exp(-E*1000/TEMP) }
```

The actual fitting is done with

```
fit <- nls(
      k~fitfunc(TEMP=dataframe$t,K0,A,E),
      dataframe,
      start=c("K0"=0.025,"A"=800,"E"=3),trace=T)
```

Note that starting values need to be given. The function ‘c()’ concatenates its arguments into a vector or list of objects.

In this case a finite difference gradient is calculated.

Now we take a closer look at the object returned by ‘nls’:

```
fit
Residual sum of squares : 3.220868e-05
parameters:
     K0          A           E
 0.0257639 1107.09 2.923585
formula: k ~ fitfunc(TEMP = dataframe$t, K0,
                  A, E)
8 observations
```

```
attributes(fit)
$names:
[1] "parameters" "formula"  "call"
[4] "residuals"   "R"           "fitted.values"
[7] "assign"

$class:
[1] "nls"
attributes(fit)
plot(dataframe$t,resid(fit))
abline(0,0)
plot(dataframe$t,dataframe$k)
lines(dataframe$t,fitted(fit))
```

A high level graphics function like ‘plot’ displays a new plot, whereas a low level function like ’ lines’ does not. Print options can be found in the ‘Graph’ and ‘Options‘ menu of the graphics window.

```
summary(fit)
Formula: k ~ fitfunc(TEMP = dataframe$t, K0, A, E)

Parameters:
        Value        Std. Error  t value
K0 2.57639e-02 3.03168e-03 8.498220
 A 1.10709e+03 1.44396e+03 0.766706
 E 2.92359e+00 3.97733e-01 7.350630

Residual standard error: 0.00253806 on 5 degrees of freedom

Correlation of Parameter Estimates:
    K0     A
A 0.872
E 0.884 0.999
```

gives a concise summary, with the estimated errors in the parameters.

The information from the (linear approximation) covariance matrix of the estimated parameters is given in two parts. First the Standard Errors of the parameters (square roots of the diagonal elements). Second the correlation coefficients (scaled off diagonal elements).

Thus, the richness of an object of class *nls* has been illustrated.

Usage of R in experimental design

The function *lmdemo* simulates two experiments: with independent variables equidistant between *xmin=0* and *xmax=range,* and between *xmin=-range/2* and *xmax=xmin+range*. Plots are made, and a list is returned containing summaries of the output of the function *lm.* Note that the last line of the function contains the returned object, in this case a list containing four objects. Different experimental designs can be simulated using e.g. the input arguments *step* and *nrep* (the number of independent repeats).

lmdemo

```
function(xmin=0, step=1,range=30,xmax = xmin+range,sigma = 0.1, nresplot=1,
intercept=2,slope=1,iseed=123,nrep=1)
{
x <- seq(xmin, xmax,step)
if (nrep>1) {x<-rep(x,nrep)}
set.seed(iseed)
noise <- rnorm(length(x)) * sigma
y <- intercept+slope*x + noise
n <- ceiling(xmax)
x2 <- seq(-range/2,range/2,step)
if (nrep>1) {x2<-rep(x2,nrep)}
y2 <- intercept+slope*x2 + noise
olm1 <- lm(y ~ x)
par(mfrow = c(2., nresplot+1))
plot(x,y,type="p")
lines(x,fitted(olm1))
plot(olm1,which=c(1:nresplot))
sum1 <- summary(olm1,correlation=T)
olm2 <- lm(y2 ~ x2)
plot(x2,y2,type="p")
lines(x2,fitted(olm2))
plot(olm2,which=c(1:nresplot))
sum2 <- summary(olm2,correlation=T)
list(olm1 = olm1, olm2 = olm2,sum1 = sum1, sum2 = sum2)
}
```

Collect the output of functions in new objects, eg:

olmdemo <- lmdemo()

Print an object by typing its name (the $ picks the element from the list):

```
> olmdemo$sum1

Call:
lm(formula = y ~ x)
Residuals:
      Min        1Q    Median        3Q       Max
-0.189593 -0.066725 -0.006956  0.064944  0.181874


Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.025964   0.033958   59.66   <2e-16 ***
x           0.998057   0.001944  513.30   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.09683 on 29 degrees of freedom
```

```
Multiple R-squared: 0.9999,     Adjusted R-squared: 0.9999
F-statistic: 2.635e+05 on 1 and 29 DF,  p-value: < 2.2e-16

Correlation of Coefficients:
  (Intercept)
x -0.86
```

The function *parcorrtdemo()* illustrates the stochastic properties of the estimated parameters. It repeatedly simulates and fits the same straight line with independent variables equidistant between *xmin=-15* and *xmax=xmin+range*. Its input arguments are similar to those of *lmdemo.* Before the argument *lellipse* can be used, a package has to be installed.

Installing package ellipse

After you have downloaded the latest version of **R**, you should choose a mirror nearby (you should have a fast internet connection to download necessary packages).

Choose to Install package: *ellipse*
Choose to Install package: *plotrix*

Usage of RStudio

Load a workspace via the Session menu.

You can also Clear your workspace via the Session menu, and then load another workspace.

Load a package by clicking the checkbox on the packages tab, or *Install* if it is not in that list.

Alternatively, type in the console

```
install.packages(“ellipse”)
```

Exercises with R

Exercise 1: Experimental design: *t*-value and correlation coefficient *r*

The function *lmdemo* was written to experiment with a simple experimental design for a straight line (that crosses at the intercept and possesses a certain slope). See §2.1 in the lecture notes, in particular Fig.2.2. For help on arguments of the function that you are calling, press the F1 function key.

```
> lmdemo
function(xmin=0, step=1,range=30,xmax = xmin+range,sigma = 0.1, nresplot=1,
intercept=2,slope=1,iseed=123,nrep=1)
```

The x vector extends from *xmin* to *xmax = xmin+range*, with steps of *step*. Optionally the x vector can be repeated *nrep* times.

The simulated linear model is *y=intercept+slope*x+noise.* The noise has standard deviation *sigma* and the randomgenerator is controlled via its starting value *iseed*.

Two designs are compared, from *xmin* to *xmax* (summary sum1), and a symmetric design from *–range*/2 till *range*/2 (summary sum2). More advanced residual diagnostic plots can be drawn with *nresplot* up to 6.

You can collect the output of functions in new objects, eg:

```
olmdemo <- lmdemo()
```

and inspect the contents of the elements of the output list, eg:

```
attributes(olmdemo)
olmdemo$sum1
```

Experiment with this function (in particular investigate the effect of *xmin* and *step,* try different starting values *iseed* for the random number generator), and describe clearly the behavior of the *t*-value of the estimated parameters (intercept and slope) and of the correlation coefficient between the estimated parameters.

In particular attention should be paid to the (auto)covariance matrix of the estimated parameters. In the output of the function *summary t*-values and correlation coefficients are given. The *t*-value of θ is defined as $\hat{\theta}/\hat{\sigma}_\theta$ (which is in the R-summary the ratio Estimate / Std. Error). A larger *t*-value implies a larger precision in the estimated parameter.

Report a table with relevant input arguments of *lmdemo*, estimates $\hat{\theta}$, $\hat{\sigma}_\theta$, $\hat{\theta}/\hat{\sigma}_\theta$ for the different designs.

E.g. to investigate the effect of xmin (using the default values for all the other parameters) on olmdemo$sum1 we choose values of xmin between -60 and 30 in steps of 15. Note that $xmin = -15$ in the default case corresponds to olmdemo$sum2. Fill the table and draw a conclusion:

Dependence of t-values and correlation coefficient upon xmin

xmin	$\hat{\theta}_0$	$\hat{\sigma}_{\hat{\theta}_0}$	t_{θ_0}	$\hat{\theta}_1$	$\hat{\sigma}_{\hat{\theta}_1}$	t_{θ_1}	$\rho_{\theta_0, \theta_1}$
0							
-15							
-30							
-45							
-60							
15							
30							

Since filling a table by hand is a rather tedious procedure, more functions (with the same arguments as *lmdemo*) have been written that automate this process, called *lmdemorange, lmdemosigma* and *lmdemoiseed.* Note that with *lmdemorange* both *xmin* and *xmax* are automatically computed.

Use these functions with the following arguments, and draw conclusions:

lmdemoiseed()

lmdemoiseed(iseed=789)

lmdemorange()

lmdemorange(range=2,step=0.1)

lmdemorange(range=2,step=0.1,sigma=1)

lmdemorange(range=2,step=0.1,sigma=1,iseed=789)

lmdemosigma()

lmdemosigma(iseed=789)

Compare the following simulations

```
olmdemo1 <- lmdemo(range=29)
olmdemo2 <- lmdemo(range=29,step=29,nrep=15)
```

comment and discuss the results and relate them to the lecture notes, in particular Example 2.3. Collate the sum1 and sum2 results in these two cases in a table and draw a conclusion.

Exercise 2: Repeated simulation of a straight line to illustrate the *t*-value and *r*

Obviously each realization of the noise, determined by *iseed*, is different. The function *parcorrtdemo()* repeatedly simulates (with different values of *iseed*) and fits a straight line. It illustrates the stochastic properties of the estimated parameters.

```
> parcorrtdemo
function(xmin=-15,step=1,range=30,xmax = xmin+range, sigma = 0.1,nsim=100,
iseed=4123,intercept=2,slope=1,lellipse=F,nrep=1)
Estimated parameters of nsim realizations of the design from xmin to xmax (see lmdemo).
When the logical variable lellipse=T theoretical results are drawn (package ellipse must be
installed).
oparcorrtdemo<-parcorrtdemo()
oparcorrtdemo$sumbest
```

Describe what happens, and explain the plots. Interpret the scatterplot of the estimated parameters, cf. Fig.2.9.A. In particular, you can compare the estimated correlation coefficient and estimated standard errors for the parameters from a single experiment with the sample properties (estimated correlation coefficient of the scatterplot and rms deviation of the estimated parameters). Experiment with this function (in particular investigate the effect of *xmin* and *step, iseed, nsim, sigma*), and describe clearly the behavior of the histograms of the estimated parameters (intercept and slope, cf. Fig.2.9.B and C), of the mean square error (*mse*, cf. Fig.2.9.D, note that in the summary statistics the root mean square error *rmse* is reported as Residual standard error), and of the *t*-values of the estimated parameters (more precisely, of $(\hat{\theta} - \theta_{\mathrm{true}}) / \hat{\sigma}_{\hat{\theta}}$ where θ_{true} is the parameter value used for the simulation, cf. Fig.2.9.E and F).

Confidence regions

The package *ellipse* draws confidence regions for a pair of parameters. With linear regression these are ellipsoidal (see e.g. Fig.2.9.A). Here are the commands to draw an ellipse for the linear model of a straight line with two parameters:

```
parcorrtdemo(lellipse=T,iseed=12345)
```

The ellipse that is drawn corresponds to the 90% confidence region of the best simulation outcome (as determined in the loop).

In addition, the logical *lellipse* is used to draw theoretical results of the histograms in Fig.2.9.B-F. What are the underlying theoretical distributions of the histograms? Relate the scatterplot and all five histograms to equations in the lecture notes.

Repeat this with a few different experimental designs of your own choice, and comment on the outcome(s) that you consider most informative.

Exercise 3: Simulation of a quadratic curve

The function *mlmdemo()* simulates and fits a quadratic curve.

```
> mlmdemo
function(xmin=0, step=1,range=30,xmax = xmin+range,sigma = 0.1, nresplot=1,
intercept=3,slope1=1,slope2=2,iseed=123,power=2,nrep=1)
```

(see lmdemo) Linear model *y=intercept+slope1*x+slope2*x^power+noise*.

```
mlmdemo()
```

Describe what happens, interpret the outcomes, and explain the plots.

The function *mlmdemorange()* investigates how the *t*-values and correlation coefficients depend upon the experimental design.

Answer the same questions as with exercise 1. Discuss the correlation coefficients, *t*-values, and the efficiency of the different experimental designs.

Compare also the following simulations

```
olmdemo1 <- mlmdemo()
olmdemo2 <- mlmdemo(step=15,nrep=10)
```

comment and discuss the results and relate them to the lecture notes.

Hint: with a symmetric design, odd and even functions are orthogonal, e.g.

$\sum_{i=-xmax}^{xmax} 1 \cdot x_i = 0$ and $\sum_{i=-xmax}^{xmax} x_i^2 \cdot x_i = 0$. Then what do you expect for the correlation coefficients of the corresponding parameters?

The inverse problem

Until now you have first chosen the parameters for your simulation, and then used linear models to estimate parameters that describe the data. Thus the correct model was known in advance. However, with real data, the correct model is often unknown, but a suitable class of models is available. We can approach the real data situation better when you use data that have been simulated in an unknown way.

Exercise 4: Fitting unknown simulated data

Use the function *mlmfit*

```
> mlmfit
function (formula=y ~ x + I(x^power), nresplot = 1,  power = 2,
datamlm=mlmdatacrit,lpoly=F)
```

to fit the unknown simulated data *mlmdatacrit.* Logical *lpoly=T* uses orthogonal polynomials upto order *power*. Compare the results with *lpoly=F* (which uses a formula instead), and do this also for a third order polynomial using the formula *y ~ x + I(x^2)+ I(x^3)*. Collate the results in these four cases (*lpoly=F* or *T,* both with second or third order polynomial) in a table and draw a conclusion.

Exercise 5: Multiple linear regression

A function *powerfitplot* has been written which performs the regression Eq. 2.37.

```
> powerfitplot
function (data = powerLH2, spec = spectraLH2, imin = 1, imax = data$nt,errbar=F)
```

> Linear model explained in example 2.5 of the lecture notes. Here *ncomp=4*, and default all powers are shown (data$nt=16). Different consecutive powers can be selected between 1 and 16 using the indices *imin* and *imax*. When the logical variable *errbar=T* also the error bars of the estimated amplitudes are drawn (package plotrix must be installed).

powerfitplot(imax=1)

fits one spectrum as a linear combination of four basis spectra. Perform and interpret this fit. In total, sixteen spectra were measured as a function of laser power. Since each measured spectrum can be fitted in this way, the power dependence of each of the concentration parameters can be estimated by:

powerfitplot()

To look at a subset use e.g.:

powerfitplot(imin=5, imax=7)

Perform and interpret the fit of all sixteen spectra.

Compare this with the results where the last component has been omitted:

```
> powerfitplotomit
function (data = powerLH2, select=c(1:4),spec = spectraLH2[,select], imin = 1, imax =
data$nt,errbar=F)
```

> As in *powerfitplot*, but now selecting spectra via *select*. This function also allows you to choose the spectra that you want to use in your model. E.g.
> *powerfitplotomit(select=c(1,3,5))*
> uses only the second and fourth basis spectrum. Note that the 1 is compulsory for the wavelengths, and the numbers 3 and 5 correspond to the second and fourth basis spectrum.

Comment on the necessity of each of the four basis spectra. Demonstrate what happens when you leave out one of the necessary basis spectra.

Exercise 6: Simulation and fit of an exponential decay

The function *simexp()* simulates and fits an exponential decay. Two different experimental designs are compared: with equidistant time points, and with logarithmically equidistant time points (option *llogt=T* below).

```
> simexp
function(rate=1, tmax = 3, deltat=0.1, sigma = 0.1,iseed=123,nrep=1,tmind=0.1)
```

The t vector extends from 0 to *tmax*, with steps of *deltat*. Optionally the t vector can be repeated *nrep* times. Nonlinear model *y(t)=exp(-rate*t)+noise*. The noise has standard deviation *sigma* and the randomgenerator is controlled via its starting value *iseed*. *tmind* is the minimum time for the logarithmic timescale.

With a larger decay rate the parameter precision will be different:

```
simexp(rate=3)
```

Describe what happens, and explain the plots. Try also different decay rates.

The function *simexpcorr()* repeatedly simulates and fits an exponential decay. It illustrates the stochastic properties of the estimated parameters.

```
> simexpcorr
function(rate=1, tmax = 3, deltat=0.1, sigma = 0.1,iseed=123,nsim=100,amp=1,
lellipse=F,nrep=1,tmind=0.1,llogt=F)
```

Estimated parameters of nsim realizations of simexp. Logical llogt is used for the logarithmic timescale.

Nonlinear model y(t)=amp*exp(-rate*t)+noise. When the logical variable lellipse=T linear approximation theoretical results are drawn (package ellipse must be installed).

To compare the two different experimental designs:

```
simexpcorr(rate=3)
dev.new()
simexpcorr(rate=3,llogt=T)
```

Describe what happens, and explain the plots. Experiment with *rate* between 0.1 and 10.

Analogous to the function *parcorrtdemo* from Exercise 2, also *t*-values $(\hat{\theta} - \theta_{\mathrm{true}}) / \hat{\sigma}_{\hat{\theta}}$ are calculated for the deviation of the parameters, and histograms are made. With nonlinear regression these are of course based upon the linear approximation covariance matrix.

For each of the histograms of the parameters also the rms value is computed. Comment on the outcome of these, and experiment with different values of *iseed*. What do you conclude on the differences between the linear and the nonlinear model, in particular regarding the adequacy of the standard error ? What can you say about the adequacy of the linear approximation standard error, how does it depend upon the particular choice of the simulation parameters?

Confidence regions

Here is the command to draw an approximate 90% confidence ellipse (see e.g. Fig.2.15.A) for the simulation of a nonlinear model of an exponential decay with two parameters:

```
osimexpcorr<-simexpcorr(lellipse=T, iseed=12345)
osimexpcorr$sumbest
```

Note that with nonlinear regression the ellipse (drawn around the best estimate) is of course based upon the linear approximation covariance matrix.

In addition, the logical *lellipse* is used to draw linear approximation theoretical results of the histograms in Fig.2.15.B-F. What are the underlying theoretical distributions of the histograms? Relate the scatterplot and all five histograms to equations in the lecture notes.

Repeat this with a few different experimental designs of your own choice, and comment on the outcome(s) that you consider most informative.

Installing package paramGUI

For all the following exercises you will need a home-written package called paramGUI, which depends upon the package TIMP. On the packages tab of R Studio select **paramGUI**

Then automatically dependent packages like TIMP will be installed.

Then type in the R Studio console:

```
library(paramGUI)
```

To start the GUI in the same R session type in the console:

```
startGUI()
```

Exponential decay

The solution of the homogeneous linear differential equation $\frac{d}{dt}c(t) = -kc(t)$ with initial condition $c(0) = c_0$ is given by $c(t) = c_0\exp(-kt)$. The solution of the inhomogeneous linear differential equation $\frac{d}{dt}c(t) = -kc(t) + i(t)$ with $c(-\infty) = 0$ is given by the convolution of the homogeneous solution $\exp(-kt)$ with the input function $i(t)$: $c(t) = \int_0^{\infty}\exp(-ks)i(t-s)ds = \int_{-\infty}^{t}\exp(-k(t-s))i(s)ds$. Check this by differentiating the second representation.

Exercise 7: Fitting a decay rate in combination with an IRF

Any real measurement will have a finite time resolution. When the aim is to study ultrafast processes an Instrument Response Function (IRF) has to be taken into account. The GUI contains a Gaussian shaped IRF $i(t)$ with two parameters for the location (mean) μ and the full width at half maximum (FWHM) Δ:

$$i(t) = \frac{1}{\tilde{\Delta}\sqrt{2\pi}}\exp(-\log(2)(2(t-\mu)/\Delta)^2)$$

where $\tilde{\Delta} = \Delta/(2\sqrt{2\log(2)}) \approx \Delta/2.35$. The above convolution yields

$$c(t;k,\mu,\Delta) = \frac{1}{2}\exp(-kt)\exp\left(k\left(\mu+\frac{k\tilde{\Delta}^2}{2}\right)\right)\left\{1+\operatorname{erf}\left(\frac{t-(\mu+k\tilde{\Delta}^2)}{\sqrt{2}\tilde{\Delta}}\right)\right\}$$

where the error function is defined as $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp(-t^2)dt$.

First select the *Simulate* tab. We simulate a kinetic model with a single trace, by choosing Maximum wavelength 400. By clicking the checkbox we can add a Gaussian shaped IRF with parameters for location $\mu = 10$ and width $\tilde{\Delta} = 5$. The default model contains a sum of two decay rates, both with amplitude 1, and increase the maximum of timepoints to 110. Change the Decay rate box $k = 0.05$ and Amplitude box to 1, which corresponds to a simulation of a single exponential decay. Likewise, take care that you have a single parameter for the spectral location, width, and skewness. Now when you press the *Simulate* button a realization of the additive noise is generated (where you can adjust the seed and the standard deviation).

Next go to the Fitting tab, and select model type Kinetic, and change the starting value of the decay rates box to 0.04. By clicking the checkbox we can add a Gaussian shaped IRF to the model that is used in the analysis of the data. Press *Fit model*. Increase the number of iterations until the fit converges, which you can judge from the Fit progression tab of the Results. Note that in the Concentrations panel zero time now corresponds to the maximum of the IRF. With an IRF it is sometimes advisable to use a time scale linear around this maximum, and logarithmic thereafter. You can change the default linear part by inserting e.g. 10 in the Linear-Log axis box.

Notice the rise time with decay rate $k = 0.05$. Now increase and decrease this simulation decay rate and simulate and fit the new data. Fix the Stdev. noise at 0.01. Interpret the resulting *Fit diagnostics* tab. What decay rates can reliably be estimated? Give both upper and lower bound. Explain why.

The inverse problem

Until now you have first chosen the parameters for your simulation, and then used kinetic models to estimate parameters that describe the data. Thus the correct model was known in advance. However, with real data, the correct model is often unknown, but a suitable class of models is available. We can approach the real data situation better when you use data that have been simulated in an unknown way. For this you can select the **I/O tab** and press Choose File.

Exercise 8: Fitting unknown simulated kinetic data

The folder **unknownkinetic** contains six sets of simulated data. Copy this folder, press Choose File and browse to it. Now try to fit each data set using a kinetic model, possibly with an IRF, that you consider most appropriate, and motivate the choice of the model. Describe the line of reasoning that you followed. When you encounter fitting errors return to the most simple model that still works, write down the estimated parameters, and choose new starting parameters of a slightly more complicated model based upon these parameters and upon careful investigation of the residuals. Comment on the parameter summary on the Diagnostics tab of the Results and on the graphical output of your best fit.

Fill the following table:

Summary of the fit of unknown simulated kinetic data

dataset	rms error	number of components

Exercise 9: Fitting a skewed Gaussian shape to a spectrum

Spectra are often parameterized with a skewed Gaussian in the energy domain, which depends upon three parameters: location $\bar{\nu}_{\max}$, width $\Delta\bar{\nu}$, and skewness *b:*

$$\varepsilon(\bar{\nu}) = \exp(-\ln 2[\ln(1+2b(\bar{\nu}-\bar{\nu}_{\max})/\Delta\bar{\nu})/b]^2)$$

Note that in the limit of skewness parameter *b* equal to zero $\varepsilon(\bar{\nu})$ simplifies to a normal Gaussian (since $\lim_{b\to 0}\frac{\ln(1+bx)}{b} = x$):

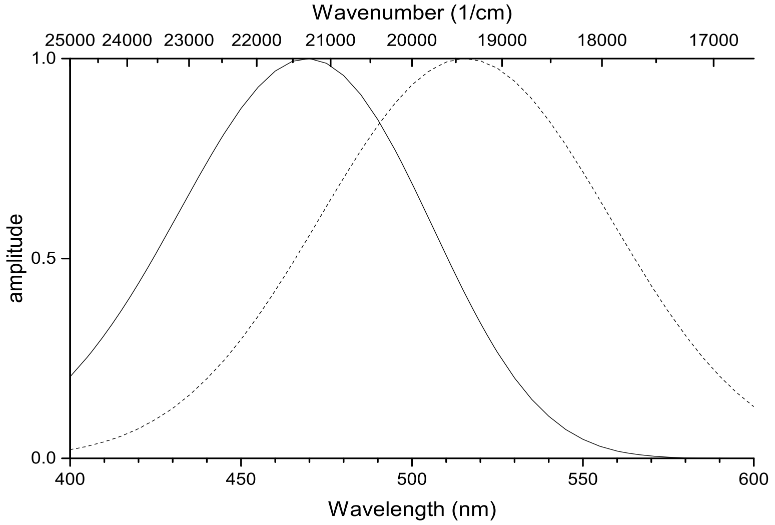
$$\varepsilon(\bar{\nu}) = \exp(-\ln 2[(2(\bar{\nu}-\bar{\nu}_{\max})/\Delta\bar{\nu})]^2)$$

where $\Delta\bar{\nu}$ is now the full width at half maximum (FWHM). Compared to the more common definition $\exp(-(x-\mu)^2/(2\sigma^2))$ we have $\sigma = \Delta/(2\sqrt{2\ln(2)}) = \Delta/2.35$.

The unit of wavenumber $\bar{\nu}$ is reciprocal centimeter (also called wavenumbers), abbreviated cm^{-1}. Since the unit of wavelength λ is usually nanometer (nm), the conversion reads

$$1\,\mathrm{nm} = 10^{-7}\mathrm{cm} \qquad \bar{\nu}_{\max} = \frac{10^7}{\lambda_{\max}} \qquad \lambda_{\max} = \frac{10^7}{\bar{\nu}_{\max}}$$

e.g. 500 nm corresponds to $20000\,\mathrm{cm}^{-1}$ and 454.5 nm corresponds to $22000\,\mathrm{cm}^{-1}$.



Examples of skewed Gaussians. Solid and dashed spectrum have spectral parameters $(\bar{\nu}_{\max}, \Delta\bar{\nu}, b)$ equal to (21307, 3930, 0.277) and (19396, 3743, 0.197)

First, we simulate a single spectrum, by choosing Maximum time 0 (if necessary change

Maximum wavelength back to 600, and uncheck the IRF box). Change the spectral parameters $(\bar{\nu}_{\max}, \Delta\bar{\nu}, b)$ to (20000,3000,0.1) and amplitude to 1, thus simulating a single band. Set the Stdev. noise at 0.01. Press the *Simulate* button. Next go to the Fitting tab, and select model type *Spectral*. Press *Fit model*. Increase the number of iterations until the fit converges, which you can judge from the Fit progression tab of the Results. Interpret the resulting *Fit diagnostics* tab.

Use the Reload button to retrieve the default settings to simulate a sum of two skewed Gaussian bands, again choose Maximum time 0. Investigate whether you can estimate the parameters with the Stdev. noise doubling from 0.01 till 0.32. What do you conclude? When the spectral parameters can no longer be precisely estimated, you can analyse the sum with a single band. Does this work? How much does the fit quality decrease? Can you judge from the residuals whether one of two bands are needed?

Exercise 10: Fitting unknown simulated spectral data

The folder **unknownspectra** contains six sets of simulated data. Copy this folder, press Choose File and browse to it. Now try to fit each data set using a spectral model, that you consider most appropriate, and motivate the choice of the model. Describe the line of reasoning that you followed. When you encounter fitting errors return to the most simple model that still works, write down the estimated parameters, and choose new starting parameters of a slightly more complicated model based upon these parameters and upon careful investigation of the residuals. Comment on the parameter summary on the Diagnostics tab of the Results and on the graphical output of your best fit.

Fill the following table:

Summary of the fit of unknown simulated spectral data

dataset	rms error	number of components

Analysis of time resolved spectra

In Exercise 5: we have seen that the concentration parameters of a mixture of components can be estimated from a measured spectrum by means of linear regression when the component spectra are known. In Exercise 6: we have seen that the parameters of an exponential decay can be estimated by means of nonlinear regression. We are now combining these techniques to address a more complicated modelling problem: how to estimate the spectral and temporal (=kinetic) properties of a mixture of components.

Read the paper *Solving a separable nonlinear least squares problem: Parameter estimation of time resolved spectra*.

In this parameter estimation problem both linear and non-linear regression are used. The functions can simulate models with exponentially decaying components and skewed Gaussian spectral shapes. For example we give here a kinetic model with two components, giving rise to a simulated time resolved spectrum (TRS) $\psi(t, \lambda)$, which contains additive noise $\xi(t, \lambda)$:

$$\psi(t, \lambda) = \varepsilon_1(\lambda)e^{-k_1 t} + \varepsilon_2(\lambda)e^{-k_2 t} + \xi(t, \lambda)$$

The functions implemented in the paramGUI can estimate the non-linear kinetic parameters $\theta = \begin{bmatrix} k_1 & k_2 \end{bmatrix}^T$ and the (conditionally) linear spectral parameters $\varepsilon_1(\lambda)$ and $\varepsilon_2(\lambda)$ (assuming the kinetic model from the above paper is used).

Rank of the matrices involved

Simulations are done at *nt* time points and *nl* wavelengths. In matrix notation we have

$$\underline{\Psi} = \sum_{j=1}^{ncomp} c_j \varepsilon_j^T + \underline{\Xi} = CE^T + \underline{\Xi}$$

where the $nt \times nl$ matrix Ψ denotes the time resolved spectrum, the columns c_j of the $nt \times n_{\text{comp}}$ matrix C contain the concentration profiles of the components, and the columns ε_j of the $nl \times n_{\text{comp}}$ matrix E contain the spectra of the components. The $nt \times nl$ matrix $\underline{\Xi}$ contains the additive noise. When the concentration profiles of the components are linearly independent, the rank of the matrix C equals n_{comp}. Likewise, when the spectra of the components are linearly independent, the rank of the matrix E equals n_{comp}. Thus with noise free data and matrices C and E of full rank, the rank of Ψ equals n_{comp} (for an example, see Fig.3.1. on page 53). For example with two components:

$$C = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \qquad E = \begin{bmatrix} \varepsilon_1 & \varepsilon_2 \end{bmatrix} \qquad E^T = \begin{bmatrix} \varepsilon_1^T \\ \varepsilon_2^T \end{bmatrix}$$

$$c_1\varepsilon_1^T + c_2\varepsilon_2^T = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} \varepsilon_1^T \\ \varepsilon_2^T \end{bmatrix} = CE^T$$

If c_1 and c_2 are linearly independent, the rank of the matrix C equals 2, otherwise it equals 1. Likewise, if ε_1 and ε_2 are linearly independent, the rank of the matrix E equals 2, otherwise it equals 1. By definition, a product of column c_j and row ε_j^T is a matrix of rank 1. Thus the rank of the noiseless matrix Ψ equals the **minimum** of *rank(C)* and *rank(E).*

For example with two components, and a linear relation $c_2 = \alpha c_1$ we have:

$c_1\varepsilon_1^T + c_2\varepsilon_2^T = c_1\varepsilon_1^T + \alpha c_1\varepsilon_2^T = c_1(\varepsilon_1^T + \alpha\varepsilon_2^T)$, which is a matrix of rank 1 !

We can invert this reasoning, and use the rank of Ψ as a guide for the minimal number of components needed to model a time resolved spectrum of a system under study. For this we use the Singular Value Decomposition (SVD), which we write here as

$$\Psi = \sum_{l=1}^{\min(nt,\, nl)} LSV_l SV_l RSV_l^T$$

where LSV_l is a left singular vector of length *nt*, RSV_l is a right singular vector of length *nl,* and SV_l is the *l*-th singular value. With noise free data and matrices C and E of full rank

$$\Psi = \sum_{l=1}^{n_{\mathrm{comp}}} LSV_l SV_l RSV_l^T$$

The additive noise will always make the matrix Ψ of full rank, $\min(nt, nl)$. However, with a large signal to noise ratio, the singular values SV_l, $l > n_{\mathrm{comp}}$ will be much smaller than the first n_{comp} singular values. Then the number of singular values significantly different from the noise singular values indicates the rank of Ψ (for an example, see Fig.3.2. on page 54). When the smallest singular value of the noise free data matrix is about as large as the largest singular value of the noise matrix, this becomes visible as noisy n_{comp}-singular vectors, and it becomes difficult to judge the rank of the data matrix. Thus it may become impossible to resolve all components from the data.

The SVD is also used a tool to investigate the residual matrix. Unstructured left and right first singular vectors, and smoothly decreasing singular values indicate an adequate fit.

Exercise 11: Fitting with a kinetic model

First look at a model with a single component. Investigate the way in which the fit depends upon the standard deviation of the additive (normally distributed) noise (see the Diagnostics tab of the Results). Take Stdev. noise 0, 1e-3,1e-2,1e-1, and 1. What do you conclude?

Next reload, and investigate what happens to the parameter precision in a model with two components. Again investigate the way in which the fit depends upon the standard deviation of the additive noise. How much noise can be added, while still two components can be estimated (keep the default kinetic and spectral parameters)? Take Stdev. noise 0, 1e-3,1e-2,1e-1, and 1. *Discuss also the SVD panels of the data and of the residual matrix.* What do you conclude? In particular, when can data simulated with two components be fitted with a single component? How much does the fit quality decrease? Can you judge from the residuals whether one of two components are needed?

Try also (with a low noise level) a simulation where two components have either identical concentration profiles, or identical spectra. How does this affect the rank of the data matrix ?

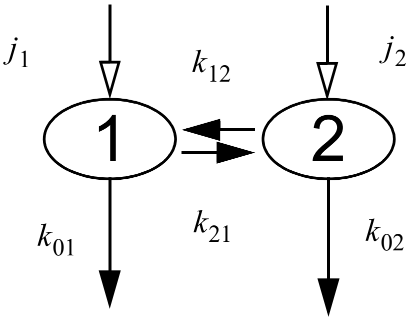
The desired result of this exercise is a specification of the signal to noise level needed to **just** resolve three components (below this level two components are sufficient). Of course this depends upon the differences between the parameters describing these components. Assume spectral parameters $(\bar{\nu}_{\max}, \Delta\bar{\nu}, b)$ of (22000,3000,0.1), (21000,3000,0.1), (20000,3000,0.1), all amplitudes equal to 1, and decay rates 0.05,0.03 and 0.01, respectively. What is the estimated rank of the data matrix at the critical noise level? Verify this with three realizations of the noise generated by changing the *Seed* on the *Simulate* tab.

Compartmental models

Compartmental systems consist of a finite number of subsystems, called compartments, which exchange with each other and with the environment, so that the concentration of material within each compartment can be described by a first-order differential equation.

Compartmental models are intensively used to describe pharmacokinetics, but also in epidemiology, chemical kinetics, medical physics and biophysics[1].

The general kinetic scheme with two compartments



can be described by two coupled linear differential equations:

$$\frac{d}{dt}\begin{bmatrix} c_1(t) \\ c_2(t) \end{bmatrix} = \begin{bmatrix} -k_{01}-k_{21} & k_{12} \\ k_{21} & -k_{02}-k_{12} \end{bmatrix}\begin{bmatrix} c_1(t) \\ c_2(t) \end{bmatrix} + \begin{bmatrix} j_1 \\ j_2 \end{bmatrix} i(t)$$

$$\frac{dc}{dt} = Kc + j(t) \qquad c = \exp(Kt) \oplus j(t)$$

The solution is a linear combination of exponential decays convolved with the IRF $i(t)$

	K	j	c	$\Psi = CE^T$	spectral relation
parallel, decay associated	$\begin{bmatrix} -k_1 & 0 \\ 0 & -k_2 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} \exp(-k_1 t) \\ \exp(-k_2 t) \end{bmatrix} \equiv C_I^T$	$C_P DAS^T$	
sequential, evolution associated, unbranched unidirectional	$\begin{bmatrix} -k_1 & 0 \\ k_1 & -k_2 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$C_S = C_P B$	$C_S EAS^T$	$EAS\ B^T = DAS$
general, species associated			$C_{III} = C_P A_{III}$	$C_{III} SAS^T$	$SAS\ A_{III}^T = DAS$

[1] http://en.wikipedia.org/wiki/Pharmacokinetics, wiki/ Chemical_kinetics, wiki/Reaction_rates, wiki/First_order_reaction, wiki/ Fluorescein_angiography, wiki/Angiogram, wiki/Compartmental_models_in_epidemiology

In paramGUI we can simulate **parallel** and **sequential** kinetic models. Below we will detail the relations between these two models, and specify the right triangular matrix *B*.

Parallel and sequential kinetic model

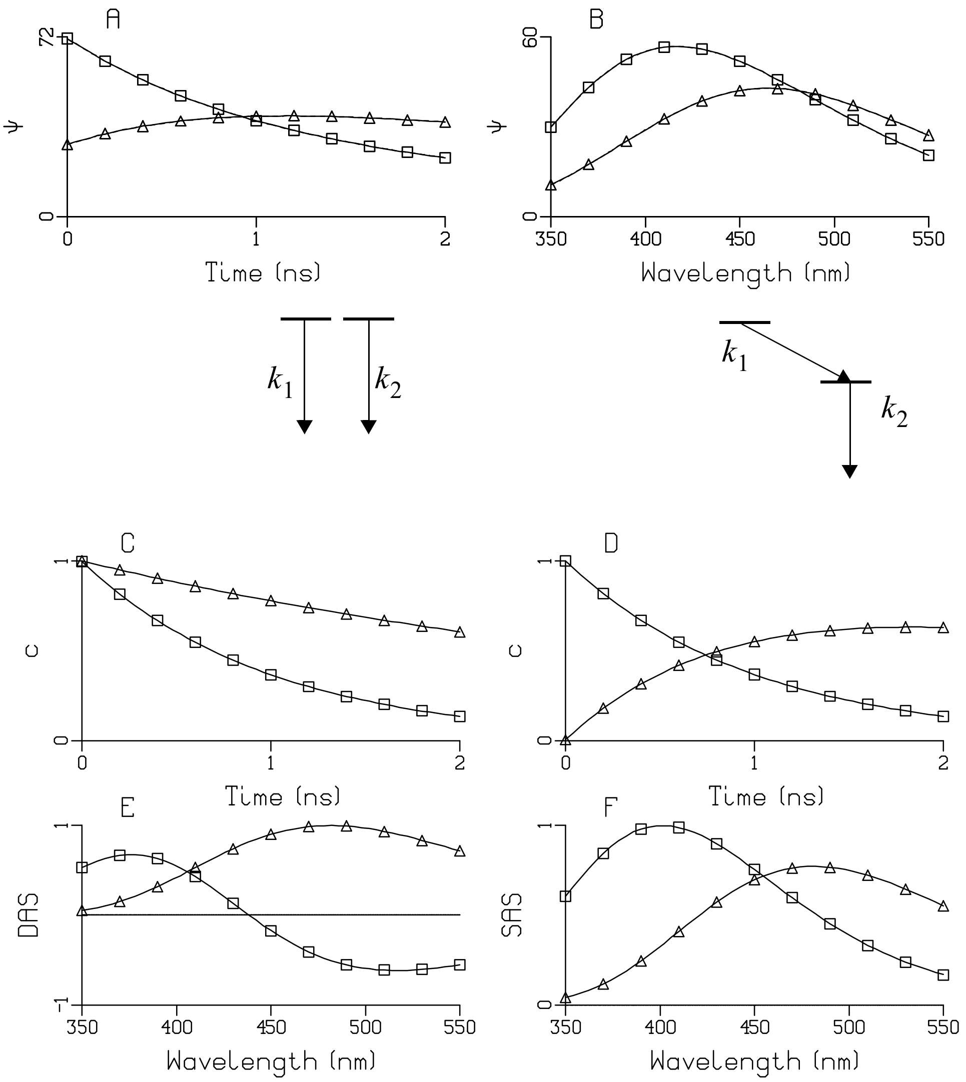


Fig.4. Global analysis of simulated data from a two-compartment model with kinetic scheme $1 \to 2$ (right inset). The first component (indicated by squares in panel D,F) decays in 1 ns, thereby forming the second component (indicated by triangles, life time 4 ns). (A) Data traces at 400 and 500 nm (indicated by squares and triangles). (B) Time gated spectra at 0.4 ns (squares) and 1.6 ns (triangles). (C,E) $c(t)$ and estimated DAS using the incorrect parallel scheme $1|2$ (left inset). (D,F) $c(t)$ and estimated EAS (or SAS) using the correct sequential scheme $1 \to 2 \to$.

When $i(t) = \delta(t)$, the concentration of component *l* in a kinetic model with parallel decays is defined as $c_l^P = \exp(-k_l t)$. For a Gaussian IRF the solution for $c_l^P(k_l)$ was given in Exercise 7. The concentration of component *l* in a sequential kinetic model is given by a linear combination $c_l^S = \sum_{j=1}^{l} b_{jl} c_j^P$ with $b_{11} = 1$ and for $j \le l$:

$b_{jl} = \prod_{m=1}^{l-1} k_m / \prod_{\substack{n=1 \\ n \neq j}}^{l} (k_n - k_j)$. For example with two components the differential equations are

$$\frac{d}{dt}\begin{bmatrix} c_1(t) \\ c_2(t) \end{bmatrix} = \begin{bmatrix} -k_1 & 0 \\ k_1 & -k_2 \end{bmatrix}\begin{bmatrix} c_1(t) \\ c_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} i(t) \qquad \begin{aligned} \frac{d}{dt}c_1(t) &= -k_1 c_1(t) + i(t) \\ \frac{d}{dt}c_2(t) &= -k_2 c_2(t) + k_1 c_1(t) \end{aligned}$$

Check that c_1^S and c_2^S defined below are indeed the solutions to these differential equations.

$$b_{12} = \frac{k_1}{k_2 - k_1} \qquad b_{22} = \frac{k_1}{k_1 - k_2} \qquad B = \begin{bmatrix} 1 & \dfrac{k_1}{k_2 - k_1} \\ 0 & \dfrac{k_1}{k_1 - k_2} \end{bmatrix} \qquad B^T = \begin{bmatrix} 1 & 0 \\ \dfrac{k_1}{k_2 - k_1} & \dfrac{k_1}{k_1 - k_2} \end{bmatrix}$$

$c_1^S = c_1^P \qquad c_2^S = \frac{k_1}{k_1 - k_2}(c_2^P - c_1^P)$. And in matrix notation $C_S = C_P B$, where the matrices C_S and C_P are both $nt \times 2$, and *B* is 2×2. In transposed form

$$C_S^T = B^T C_P^T$$

As a consequence of the linear relation $C_S = C_P B$, also the spectra estimated with the two different types of models are linearly related: $C_S \cdot EAS^T = C_P B \cdot EAS^T = C_P \cdot DAS^T$ and thus $DAS = EAS \cdot B^T$. DAS stands for Decay Associated Spectra, and EAS for Evolution Associated Spectra. The matrices DAS and EAS are both $nl \times 2$. In particular, inserting the above we find $DAS = EAS \cdot B^T$ or $\begin{bmatrix} \varepsilon_1^P & \varepsilon_2^P \end{bmatrix} = \begin{bmatrix} \varepsilon_1^S & \varepsilon_2^S \end{bmatrix}\begin{bmatrix} 1 & 0 \\ \dfrac{k_1}{k_2 - k_1} & \dfrac{k_1}{k_1 - k_2} \end{bmatrix}$ or

$\varepsilon_1^P = \varepsilon_1^S + \frac{k_1}{k_2 - k_1}\varepsilon_2^S$ and $\varepsilon_2^P = \frac{k_1}{k_1 - k_2}\varepsilon_2^S$. Conversely, $\varepsilon_1^S = \varepsilon_1^P + \varepsilon_2^P$ and $\frac{k_1}{k_1 - k_2}\varepsilon_2^S = \varepsilon_2^P$.

Check that $c_1^S\varepsilon_1^S + c_2^S\varepsilon_2^S = c_1^P\varepsilon_1^P + c_2^P\varepsilon_2^P$. See also **Streak.pdf.**

Exercise 12: Fitting with a sequential kinetic model

To understand better the difference between DAS and EAS, simulate a model using a sequential kinetic scheme with two components, and spectra which slightly overlap. Lower the *Stdev. noise* to 0.01. First analyse these data with the correct, sequential scheme. Then begin a second RStudio session, and startGUI(). Simulate exactly the same data, but this time analyse these same data with the parallel scheme. Describe the differences that you notice between the estimated concentrations and spectra using the sequential and the parallel kinetic scheme. Express in your own words the relations between the concentration profiles using the sequential or the parallel kinetic scheme. Likewise, explain the relations between the estimated DAS and EAS. What are the elements of the B matrix with the chosen decay rates? Check that you understand the relations also numerically.

Exercise 13: Fitting unknown simulated data

The folder **unknown** contains sets of simulated data. Copy this folder, press Choose File and browse to it. Now try to fit the data using a model that you consider most appropriate, and motivate the choice of the model. Describe the line of reasoning that you followed. When you encounter fitting errors return to the most simple model that still works, write down the estimated parameters, and choose new starting parameters of a slightly more complicated model based upon these parameters and upon careful investigation of the residuals (in particular the first singular vectors of the residual matrix). In this exercise, try using kinetic models, possibly with an IRF. Comment on the parameter summary on the Diagnostics tab of the Results and on the graphical output of your best fit.

For each of the **three** simulated data sets, try to fit find the most appropriate model. Consider using a parallel or a sequential kinetic model. Comment on the parameter summary and on the graphical output of your best fit.

Exercise 14: Fitting real data with a kinetic model

To analyse real data paramGUI can also be used. There are two sets of real data: the default data are time resolved difference spectra. Proper starting values for the IRF are 0.1 and 0.2.

Plot linear until 0.5 ps. Then try to add more rate constants based upon analysis of the residuals. How many rate constants are needed to describe these data? Interpret the results. After you have obtained a satisfactory fit, try to interpret the spectra estimated with the parallel model, called Decay Associated Difference Spectra (DADS, see **Streak.pdf**). Compare with the sequential model which estimates Evolution Associated Difference Spectra (EADS). In the papers **PCP.pdf** and **PCP_SI.pdf** (fig. SI6, p.22) you can read more about energy transfer in the PCP chromophore protein complex.

The second set of data are called *streakdata.txt*. Press Choose File and browse to it. Good starting values for the IRF are -84 and 1.5. Plot linear until 20 ps. Then try to add more rate constants based upon analysis of the residuals.

These data are also described in **Streak.pdf**, together with the definitions of the Gaussian shaped Instrument Response Function (IRF), Decay Associated Spectra (DAS), and Evolution Associated Spectra. (EAS). Try to reproduce Fig.4 on p.234 and interpret the results. As explained in **Streak.pdf**, a backsweep is needed with the synchroscan instrument. Click the *Backsweep?* check box, and see how it improves the fit.

Exercise 15: Fitting with a spectral model

The spectral model with two components reads:

$$\psi(\lambda, t) = \varepsilon_1(\lambda;\bar{\nu}_{\max,1}, \Delta\bar{\nu}_1, b_1)c_1(t) + \varepsilon_2(\lambda;\bar{\nu}_{\max,2}, \Delta\bar{\nu}_2, b_2)c_2(t) + \xi(\lambda, t)$$

Here $c_l(t)$ is the concentration profile of component *l*.

Again investigate the parameter precision with one or two components. Are there differences between fitting the data with a kinetic or spectral model? Is one of these types of model more noise sensitive ? How does this depend upon differences between spectra or decay rates? Experiment with some parameters of your own choice.

Now you can return to the unknown data of Exercise 13: and try to fit these data using a spectral model. Are there differences between fitting the data with a kinetic or spectral model? Comment on the parameter summary and on the graphical output of your best fit.

Exercise 16: Fitting with a spectrotemporal model

The GUI contains two different implementations of a spectrotemporal model. In the simplest case it is assumed that for each component the chosen kinetic and spectral model function apply, and only an additional amplitude parameter a_l is needed. Then with two components the model function reads:

$$\psi(\lambda, t) = a_1\varepsilon_1(\lambda;\bar{\nu}_{\mathrm{max},\,1}, \Delta\bar{\nu}_1, b_1)c_1(k_1, t) + a_2\varepsilon_2(\lambda;\bar{\nu}_{\mathrm{max},\,2}, \Delta\bar{\nu}_2, b_2)c_2(k_2, t) + \xi(\lambda, t)$$

In the more general case each component is described by a linear combination of the chosen kinetic and spectral model functions. Then with n_{comp} components the model function reads:

$$\psi(\lambda, t) = \sum_{i=1}^{n_{\mathrm{comp}}}\sum_{j=1}^{n_{\mathrm{comp}}} c_i(k_i, t)a_{ij}\varepsilon_j(\lambda;\bar{\nu}_{\mathrm{max},j}, \Delta\bar{\nu}_j, b_j) + \xi(\lambda, t)$$

Thus this model contains n^2_{comp} amplitude parameters. Again the amplitude parameters are conditionally linear, and can be estimated using the variable projection algorithm.
Investigate the parameter precision from the parameter summary on the Diagnostics tab of the Results. Interpret again the *t*-values and the correlation coefficients.
Start again with the one component model, and then look at two (or more) components.
Simulate your favorite model with two components. Compare the results of these two implementations (in particular the parameter precision and correlation coefficients) with results analysing the same data with the help of either the kinetic or the spectral model. Which model is most suited with which data ? Discuss the advantages and disadvantages of the different types of model.
NB. There are limits to the dimensions of the datasets which **R** can deal with.
Now you can again return to the unknown data of Exercise 13: and try to fit these data using a spectrotemporal model. Comment on the parameter summary and on the graphical output of your best fit.

Overview of the different ways of analysis

Data simulated using e.g. a two component sequential kinetic model $\Psi = C_S E^T$ can be analysed in seven different ways:

Seven different ways to analyse time resolved spectra

			intrinsically nonlinear parameters	conditionally linear parameters
kinetic	parallel	$C_P(\hat{\theta})\hat{E}_P^T$	$\theta_c = (k_1, k_2)$	$E_P \equiv DAS$
	sequential	$C_S(\hat{\theta})\hat{E}_S^T$	$\theta_c = (k_1, k_2)$	$E_S \equiv EAS$
spectral		$E(\hat{\theta})\hat{C}^T$	$\theta_\varepsilon = \begin{pmatrix} \bar{\nu}_{\max,\,1}, \Delta\bar{\nu}_1, b_1 \\ \bar{\nu}_{\max,\,2}, \Delta\bar{\nu}_2, b_2 \end{pmatrix}$	C
spectrotemporal single	parallel	$\sum_{l=1}^{n_{\text{comp}}} c_l^P(\hat{\theta}_c)a_l\varepsilon_l(\hat{\theta}_\varepsilon)$	$\hat{\theta}_c, \hat{\theta}_\varepsilon$	a_l
	sequential	$\sum_{l=1}^{n_{\text{comp}}} c_l^S(\hat{\theta}_c)a_l\varepsilon_l(\hat{\theta}_\varepsilon)$	$\hat{\theta}_c, \hat{\theta}_\varepsilon$	a_l
spectrotemporal multiple	parallel	$\sum_{i=1}^{n_{\text{comp}}^{\text{kin}}}\sum_{j=1}^{n_{\text{comp}}^{\text{spec}}} c_i^P(\hat{\theta}_c)a_{ij}\varepsilon_j(\hat{\theta}_\varepsilon)$	$\hat{\theta}_c, \hat{\theta}_\varepsilon$	a_{ij}
	sequential	$\sum_{i=1}^{n_{\text{comp}}^{\text{kin}}}\sum_{j=1}^{n_{\text{comp}}^{\text{spec}}} c_i^S(\hat{\theta}_c)a_{ij}\varepsilon_j(\hat{\theta}_\varepsilon)$	$\hat{\theta}_c, \hat{\theta}_\varepsilon$	a_{ij}

Exercise 17: Fitting with a sequential kinetic model revisited

To understand better the difference between DAS and EAS, simulate a model using a sequential kinetic scheme with two components, and spectra which slightly overlap. Lower the *Stdev. noise* to 0.01. Then analyse these data two times: first with the correct, sequential scheme. Then with the parallel scheme. Describe the differences that you notice between the estimated concentrations and spectra using the sequential and the parallel kinetic scheme.

Next, use a spectrotemporal model with 2×2 amplitude parameters. The .lin parameters on the Diagnostics tab of the Results are the elements of the above defined *B* matrix. Write down this *B* matrix, both theoretically (using k_1, k_2) and with the estimated parameter values. Now change k_2 to 0.0275 and estimate the new amplitude parameters, and write down this new B matrix.

Here is the B matrix for the three component sequential kinetic model:

$$B(k_1, k_2, k_3) = \begin{bmatrix} 1 & \dfrac{k_1}{k_2 - k_1} & \dfrac{k_1}{k_1 - k_2} \cdot \dfrac{k_2}{k_1 - k_3} \\ 0 & \dfrac{k_1}{k_1 - k_2} & \dfrac{k_1}{k_1 - k_2} \cdot \dfrac{k_2}{k_3 - k_2} \\ 0 & 0 & \dfrac{k_1}{k_1 - k_3} \cdot \dfrac{k_2}{k_2 - k_3} \end{bmatrix}$$

Simulate a three component sequential kinetic model with different rate constants, and high signal to noise ratio. Analyse these data with a spectrotemporal model that uses C_P, and check that the linear parameters on the Diagnostics tab of the Results agree with the above B matrix.

Now you can again return to the unknown data of Exercise 13: and try to fit these data using a sequential kinetic model. Also try a spectrotemporal model with and without the sequential option. Comment on the parameter summary and on the graphical output of your best fit. With the spectrotemporal model, again comment on the .lin parameters with and without the sequential option.

Arguments of the R functions used in exercises 1-5

> lmdemo

function(xmin=0, step=1,range=30,xmax = xmin+range,sigma = 0.1, nresplot=1, intercept=2,slope=1,iseed=123,nrep=1)

The *x* vector extends from *xmin* to *xmax = xmin+range*, with steps of *step.* Optionally the *x* vector can be repeated *nrep* times.

Linear model *y=intercept+slope*x+noise.* The noise has standard deviation *sigma* and the randomgenerator is controlled via its starting value *iseed.*

Two designs are compared, from *xmin* to *xmax* (summary *sum1*), and a symmetric design from *–range*/2 till *range*/2 (summary *sum2*). More advanced residual diagnostic plots can be drawn with *nresplot* up to 6.

> parcorrtdemo

function(xmin=-15,step=1,range=30,xmax = xmin+range, sigma = 0.1,nsim=100, iseed=4123,intercept=2,slope=1,lellipse=F,nrep=1)

Estimated parameters of *nsim* realizations of the design from *xmin* to *xmax* (see *lmdemo*). When the logical variable *lellipse=T* theoretical results are drawn (package *ellipse* must be installed).

> mlmdemo

function(xmin=0, step=1,range=30,xmax = xmin+range,sigma = 0.1, nresplot=1, intercept=3,slope1=1,slope2=2,iseed=123,power=2,nrep=1)

(see *lmdemo*) Linear model *y=intercept+slope1*x+slope2*x^power+noise.*

> powerfitplot

function (data = powerLH2, spec = spectraLH2, imin = 1, imax = data$nt,errbar=F)

Linear model explained in example 2.5 of the lecture notes. Here *ncomp*=4, and default all powers are shown (data$nt=16). Different consecutive powers can be selected between 1 and 16 using the indices imin and imax. When the logical variable errbar=T also the error bars of the estimated amplitudes are drawn (package *plotrix* must be installed).

> powerfitplotomit

function (data = powerLH2, select=c(1:4),spec = spectraLH2[,select], imin = 1, imax = data$nt,errbar=F)

As in powerfitplot, but now selecting spectra via *select.* See exercise 4.

> simexp

function(rate=1, tmax = 3, deltat=0.1, sigma = 0.1,iseed=123,nrep=1,tmind=0.1)

The *t* vector extends from *0* to *tmax*, with steps of *deltat.*Optionally the *t* vector can be repeated *nrep* times. Nonlinear model *y(t)=exp(-rate*t)+noise.* The noise has standard deviation *sigma* and the randomgenerator is controlled via its starting value *iseed. tmind* is the minimum time for the logarithmic timescale. See exercise 5.

> simexpcorr

function(rate=1, tmax = 3, deltat=0.1, sigma = 0.1,iseed=123,nsim=100,amp=1, lellipse=F,nrep=1,tmind=0.1,llogt=F)

Estimated parameters of *nsim* realizations of *simexp*. Logical *llogt* is used for the logarithmic timescale.

Nonlinear model *y(t)=amp*exp(-rate*t)+noise.* When the logical variable *lellipse=T* linear approximation theoretical results are drawn (package *ellipse* must be installed).