$$= \quad f(\mathbf{x}_i|\mathbf{a}) = a_1 + \sum_{j=2}^{M} a_j \cdot \tilde{x}^{j-1} \cdot (x_i')^{j-1} + \varepsilon_i$$

$$= \quad f(\mathbf{x}_i|\mathbf{a}) = b_1 + \sum_{j=2}^{M} b_j \cdot (x_i')^{j-1} + \varepsilon_i \quad \text{with} \quad a_j = \frac{b_j}{\tilde{x}^{j-1}},\ x_i' = \frac{x_i}{\tilde{x}} \ .$$

The values x_i' are within the range of $[-1, 1]$ and the matrix inversion should be more stable. The true model parameters a_j must be derived from the intermediate values b_j.

Other model functions can be treated in the same manner, as long as each x is multiplicatively combined with one model parameter. Subsection B.7.1 demonstrates the effectiveness of this normalisation technique.

6.4 Fitting of nonlinear model functions

6.4.1 Error-surface approximation

As already described in Chapter 2, the fitting of models that are nonlinear for at least one model parameter has to be performed iteratively. The set of M parameters in $\mathbf{a}$ spans an M-dimensional space. Each point in this space is characterised by a corresponding value of $\chi^2(\mathbf{a})$ according to equation (6.4). The entity of all points in that space is called the *hyper-surface* (see also Section 2.1). Starting from an initial position $\mathbf{a}$, the fitting algorithm is expected to walk towards the global minimum of $\chi^2(\mathbf{a})$.

In order to determine the right direction, the principle of expansion by Taylor series [Gel68] is used. Any function $f(x)$ that is defined in the vicinity of $f(x_0)$ and has infinite continuous derivatives $f^{(\nu)}(x)$ can be expanded in the following fashion

$$f(x) = \sum_{\nu=0}^{\infty} \frac{f^{(\nu)}(x_0)}{\nu!} \cdot (x - x_0)^{\nu}$$

or, when expressing x as sum of the point x_0 and an offset h by $x = x_0 + h$,

$$f(x_0 + h) = \sum_{\nu=0}^{\infty} \frac{f^{(\nu)}(x_0)}{\nu!} \cdot h^{\nu} \ .$$

For small offsets h, the first three terms including the first and second derivative of $f(x)$ at x_0 provide a fairly good approximation

$$f(x_0 + h) \simeq f(x_0) + \frac{f'(x_0)}{1!} \cdot h + \frac{f''(x_0)}{2!} \cdot h^2 \ . \tag{6.14}$$

Now let us analyse how this approximation can be utilised to determine the steps on the hyper-surface, i.e. the improvement of the parameter vector $\mathbf{a}$.

6.4.2 Gauss-Newton method

The Gauss-Newton method is an approach for finding the minimum in an one- or multi-dimensional signal. With respect to data fitting via least squares, the average quadratic error between the observations and the model function has to be minimised, i.e., the minimum of the hyper-surface $\chi^2(\mathbf{a})$ has to be found. This is achieved by approximating the target function by a Taylor series of second order.

The parameter vector $\mathbf{a} = (a_1 \; a_2 \; \dots \; a_j \; \dots \; a_M)^{\mathrm{T}}$ determines the dimensionality of the minimisation problem. The variables from equation (6.14) must be substituted as follows. The independent variable we are looking for is $\mathbf{a}$, thus

$$x_0 \longrightarrow \mathbf{a}$$

and the corresponding step size is

$$h \longrightarrow \mathbf{\Delta a} \; .$$

The function that has to be approximated via Taylor series is

$$f(x_0) \longrightarrow \chi^2(\mathbf{a})$$

and its derivatives are

$$f^{'}(x_0) \longrightarrow \frac{\mathrm{d}\chi^2(\mathbf{a})}{\mathrm{d}\mathbf{a}} = \sum_{j=1}^{M} \frac{\partial\chi^2(\mathbf{a})}{\partial a_j}$$

and

$$f^{''}(x_0) \longrightarrow \frac{\mathrm{d}^2\chi^2(\mathbf{a})}{\mathrm{d}\mathbf{a}^2} = \sum_{j=1}^{M}\sum_{k=1}^{M} \frac{\partial^2\chi^2(\mathbf{a})}{\partial a_j \partial a_k} \; .$$

Corresponding to equation (6.14), this leads to an approximation of the error surface at the new position $\mathbf{a} + \mathbf{\Delta a}$

$$\begin{aligned} \chi^2(\mathbf{a} + \mathbf{\Delta a}) &\simeq \chi^2(\mathbf{a}) + \sum_{j=1}^{M} \frac{\partial\chi^2(\mathbf{a})}{\partial a_j} \cdot \Delta a_j + \frac{1}{2} \cdot \sum_{j=1}^{M}\sum_{k=1}^{M} \frac{\partial^2\chi^2(\mathbf{a})}{\partial a_j \partial a_k} \cdot \Delta a_j \cdot \Delta a_k \\ &\simeq F(\mathbf{a} + \mathbf{\Delta a}) \; . \end{aligned}$$

Using matrices, this reads as

$$F(\mathbf{a}+\mathbf{\Delta a}) = \chi^2(\mathbf{a}) + (\mathbf{\Delta a})^{\mathrm{T}} \cdot \mathbf{g} + \frac{1}{2} \cdot (\mathbf{\Delta a})^{\mathrm{T}} \cdot \mathbf{H} \cdot \mathbf{\Delta a} \tag{6.15}$$

with a gradient vector

$$\mathbf{g} = \left(\frac{\partial \chi^2(\mathbf{a})}{\partial a_1} \; \frac{\partial \chi^2(\mathbf{a})}{\partial a_2} \; \cdots \; \frac{\partial \chi^2(\mathbf{a})}{\partial a_M} \right)^{\mathrm{T}} . \tag{6.16}$$

The elements g_j of $\mathbf{g}$ are, using the equation (6.4) for χ^2 and $w_i = 1/\sigma_i^2$, equal to

$$g_j = \frac{\partial \chi^2(\mathbf{a})}{\partial a_j} = \sum_{i=1}^{N} w_i \cdot \frac{\partial f(\mathbf{x}_i|\mathbf{a})}{\partial a_j} \cdot [f(\mathbf{x}_i|\mathbf{a}) - y_i] \; . \tag{6.17}$$

$\mathbf{H}$ in eq.(6.15) is the so-called *Hessian* matrix[1]

$$\mathbf{H} = \begin{pmatrix} \frac{\partial^2 \chi^2(\mathbf{a})}{\partial a_1 \partial a_1} & \frac{\partial^2 \chi^2(\mathbf{a})}{\partial a_1 \partial a_2} & \cdots & \frac{\partial^2 \chi^2(\mathbf{a})}{\partial a_1 \partial a_M} \\ \frac{\partial^2 \chi^2(\mathbf{a})}{\partial a_2 \partial a_1} & \frac{\partial^2 \chi^2(\mathbf{a})}{\partial a_2 \partial a_2} & \cdots & \frac{\partial^2 \chi^2(\mathbf{a})}{\partial a_2 \partial a_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \chi^2(\mathbf{a})}{\partial a_M \partial a_1} & \frac{\partial^2 \chi^2(\mathbf{a})}{\partial a_M \partial a_2} & \cdots & \frac{\partial^2 \chi^2(\mathbf{a})}{\partial a_M \partial a_M} \end{pmatrix} . \tag{6.18}$$

The elements of $\mathbf{H}$ are computed using the product rule for derivations $(u \cdot v)' = u' \cdot v + u \cdot v'$

$$\begin{aligned} H_{jk} &= \frac{\partial^2 \chi^2(\mathbf{a})}{\partial a_j \partial a_k} = \frac{\partial g_j}{\partial a_k} \\ &= \sum_{i=1}^{N} w_i \cdot \left[\frac{\partial^2 f(\mathbf{x}_i|\mathbf{a})}{\partial a_j \partial a_k} \cdot [f(\mathbf{x}_i|\mathbf{a}) - y_i] + \frac{\partial f(\mathbf{x}_i|\mathbf{a})}{\partial a_j} \cdot \frac{\partial f(\mathbf{x}_i|\mathbf{a})}{\partial a_k} \right] \qquad (6.19) \\ &= \sum_{i=1}^{N} w_i \cdot \frac{\partial^2 f(\mathbf{x}_i|\mathbf{a})}{\partial a_j \partial a_k} \cdot [f(\mathbf{x}_i|\mathbf{a}) - y_i] + \sum_{i=1}^{N} w_i \cdot \frac{\partial f(\mathbf{x}_i|\mathbf{a})}{\partial a_j} \cdot \frac{\partial f(\mathbf{x}_i|\mathbf{a})}{\partial a_k} . \end{aligned}$$

In order to minimise $\chi^2(\mathbf{a}+\mathbf{\Delta a}) \simeq F(\mathbf{a}+\mathbf{\Delta a})$, we have to look for a point where the gradient of $F(\mathbf{a}+\mathbf{\Delta a})$ (eq. (6.15)) is zero

$$\frac{\mathrm{d}\, F(\mathbf{a}+\mathbf{\Delta a})}{\mathrm{d}\, (\mathbf{\Delta a})^{\mathrm{T}}} = F'(\mathbf{a}+\mathbf{\Delta a}) = \mathbf{g} + \mathbf{H} \cdot \mathbf{\Delta a} = 0$$

[1] named after the German mathematician Ludwig Otto Hesse (1811-1874)

with the result of

$$\mathbf{\Delta a} = -\mathbf{H}^{-1} \cdot \mathbf{g} \,. \tag{6.20}$$

Voilá! The step $\mathbf{\Delta a}$ can be computed. However, compared to (2.6), equation (6.20) seems to be very different. Let us take a closer look at the elements of $\mathbf{g}$ and $\mathbf{H}$ (eqs. 6.17 and 6.20). It turns out that

$$\mathbf{g} = -\mathbf{J}^{\mathrm{T}} \cdot \mathbf{W} \cdot \mathbf{r} \qquad \text{and} \qquad \mathbf{H} = \mathbf{Q} + \mathbf{J}^{\mathrm{T}} \cdot \mathbf{W} \cdot \mathbf{J} \tag{6.21}$$

with $\mathbf{J}$, $\mathbf{r}$, and $\mathbf{W}$, already defined in equations (2.3), (2.4), and (2.5), respectively

$$\mathbf{J} = \begin{pmatrix} \frac{\partial f(\mathbf{x}_1|\mathbf{a})}{\partial a_1} & \frac{\partial f(\mathbf{x}_1|\mathbf{a})}{\partial a_2} & \cdots & \frac{\partial f(\mathbf{x}_1|\mathbf{a})}{\partial a_M} \\ \frac{\partial f(\mathbf{x}_2|\mathbf{a})}{\partial a_1} & \frac{\partial f(\mathbf{x}_2|\mathbf{a})}{\partial a_2} & \cdots & \frac{\partial f(\mathbf{x}_2|\mathbf{a})}{\partial a_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(\mathbf{x}_N|\mathbf{a})}{\partial a_1} & \frac{\partial f(\mathbf{x}_N|\mathbf{a})}{\partial a_2} & \cdots & \frac{\partial f(\mathbf{x}_N|\mathbf{a})}{\partial a_M} \end{pmatrix} ,$$

$$\mathbf{r} = \begin{pmatrix} y_1 - f(\mathbf{x}_1|\mathbf{a}) \\ y_2 - f(\mathbf{x}_2|\mathbf{a}) \\ y_3 - f(\mathbf{x}_3|\mathbf{a}) \\ \vdots \\ y_N - f(\mathbf{x}_N|\mathbf{a}) \end{pmatrix} ,$$

and

$$\mathbf{W} = \begin{pmatrix} w_1 & 0 & 0 & \cdots & 0 \\ 0 & w_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & w_N \end{pmatrix} .$$

The term $\mathbf{Q}$ contains the second-order derivatives in equation (6.20), which are ignored in many texts – on the one hand because it is zero for linear problems, and on the other hand because the multiplying term $[f(\mathbf{x}_i|\mathbf{a}) - y_i]$ in equation (6.20) is merely a random measurement error of each point. This error can be either positive or negative as soon as $\mathbf{a}$ is close to its optimum. If the multiplying

This text was taken from the book:

T.Strutz: *Data Fitting and Uncertainty – An introduction to weighted least squares and beyond*. ISBN-10: 3834810223, ISBN-13: 978-3834810229

„A text guiding you step-by-step from the basics to the details of the method of data fitting“