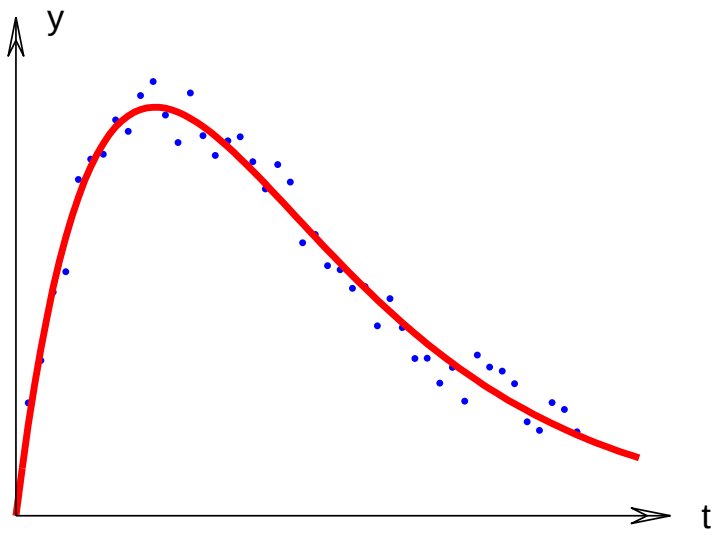
DTU Informatics
Department of Informatics and Mathematical Modeling

# Introduction to Data Fitting

**Hans Bruun Nielsen**




March 2010

# 1. DEFINITIONS AND BASIC CONCEPTS

Given *data points* $(t_1, y_1), \ldots, (t_m, y_m)$ and a *fitting model* $M(x, t)$, which depends on the independent variable $t$ and *parameters* $x = \begin{pmatrix} x_1 & \ldots & x_n \end{pmatrix}^T$. We want to "fit the model to the data", meaning that we want to determine the parameters so that $M(x, t_i) \simeq y_i$ for $i = 1, \ldots, m$.

We shall only discuss so-called *least squares fits*, where $x$ is determined such that $r_1(x)^2 + \cdots + r_m(x)^2$ is minimal. Here, the $r_i(x)$ are the *residuals*

$$r_i(x) \;=\; y_i - M(x, t_i)\;, \quad i = 1, \ldots, m\;.$$

The minimizing vector, $\hat{x}$, is called the *least squares solution*.

**Example 1.1.** The figure in the front page shows $m = 45$ data points and the *least squares fit* with the *model*

$$M(x, t) \;= x_1 e^{x_3 t} + x_2 e^{x_4 t}\;.$$

If $\hat{x}_3$ and $\hat{x}_4$ were known, then both the *free parameters* appear linearly, and we have a linear least squares problem. If all the four $x_j$ are free, then we have a nonlinear least squares problem, since at least one of the parameters appear nonlinearly.

We return to this data fitting problem in Example 3.2. ∎

The general least squares problem is defined as follows,

**Definition 1.1. Least squares problem.** Let $r : \mathbb{R}^n \mapsto \mathbb{R}^m$ with $m \geq n$ be a vector function. Find $\hat{x}$, a minimizer for the *objective function*[1)]

$$f(x) \;=\; \tfrac{1}{2}\|r(x)\|_2^2 \;=\; \tfrac{1}{2} r(x)^T r(x) \;=\; \tfrac{1}{2} \sum_{i=1}^{m} \big(r_i(x)\big)^2\;.$$

The least squares problem is said to be *linear* if the function $r(x)$ has the form

$$r(x) \;=\; y - Ax\;, \tag{1.2}$$

for a given $y \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m \times n}$. This problem is related to an overdetermined system of linear equations, which we write in the form

$$Ax \;\simeq\; y\;. \tag{1.3}$$

In Sections 8.14 – 8.15 of Eldén et al it is shown that the solution to a linear least squares problem can be found via the *normal equations*

$$\big(A^T A\big)\, \hat{x} \;=\; A^T y\;, \tag{1.4}$$

or via orthogonal transformation. If the columns of $A$ are linearly independent, then the $n \times n$ matrix $A^T A$ is nonsingular, implying that the corresponding least squares problem has a unique minimizer; cf Theorem 8.14.2 in Eldén et al.

[1)] The factor $\frac{1}{2}$ in the definition of $f(x)$ has no effect on $\hat{x}$. It is introduced for convenience, see page 6.

For a nonlinear problem the situation is more complicated. Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be a nonlinear function. In general it is very hard to find a *global minimizer*. This is a vector $\check{x}$ such that $f(\check{x}) \leq f(x)$ for all $x \in \mathbb{R}^n$. There may be several distinct global minimizers, and there may be a number of so-called local minimizers:

**Definition 1.5. Local minimizer.** Given $f : \mathbb{R}^n \mapsto \mathbb{R}$ and a small, positive number $\delta$. The vector $\hat{x}$ is a *local minimizer* for $f$ if[2)]

$$f(\hat{x}) \leq f(x) \quad \text{for all x in the neighbourhood given by } \|x - \hat{x}\| < \delta \ .$$

**Example 1.2.** Let $f(x) = \frac{1}{2} r(x)^T r(x)$ with

$$r(x) \ = \ \begin{pmatrix} x_1^2 + x_2 - 11 \\ x_2^2 + x_1 - 7 \\ 0.2(2 - x_2) \end{pmatrix} \ .$$

Figure 1.2a shows the behaviour of $f(x)$ for $x \in \mathcal{D}$, $\mathcal{D}0[-6, 6]^2$. We have plotted $\log_{10} f(x)$ in order to dampen large (and uninteresting) values of $f(x)$ and enhance values close to zero. The figure indicates that there are four local minima.

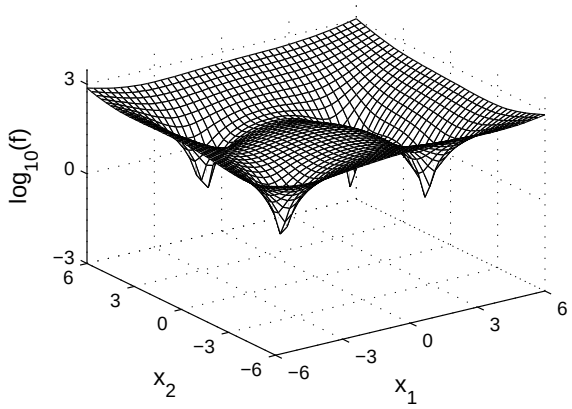



**Figure 1.2a.** *Plot of* $\log_{10} f(x)$.

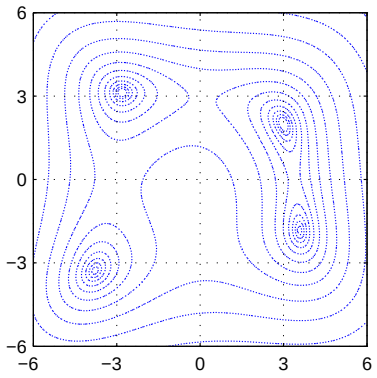



**Figure 1.2b.** *Contours,* $f(x) = 2^k, \ k = -1, 0, \ldots, 10$.

Figure 1.2b shows *contours* (also called *level curves*), ie curves along which $f(x)$ is constant. We can identify the four local minimizers. They are given below to three decimals accuracy.

| $k$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $\hat{x}_1^{[k]}$ | $-2.805$ | 3 | 3.584 | $-3.778$ |
| $\hat{x}_2^{[k]}$ | 3.130 | 2 | $-1.837$ | $-3.278$ |
| $f(\hat{x}^{[k]})$ | 0.026 | 0 | 0.295 | 0.556 |

It can be shown that all the local minimizers of $f$ are in $\mathcal{D}$, and from the function values given we see that $f$ has a unique global minimizer, $\check{x} = \begin{pmatrix} 3 & 2 \end{pmatrix}^T$.

We return to this problem in Examples 1.3, 2.1 and 2.1. ∎

We shall assume that $f$ is continuously differentiable with respect to every $x_j$. Then we have the Taylor expansion

[2)] Throughout this note $\|\cdot\|$ denotes an arbitrary norm.

$$f(x+h) \;=\; f(x) + h^T \nabla f(x) + o(\|h\|)\,, \tag{1.6}$$

where $\nabla f \in \mathbb{R}^n$ is the *gradient*

$$\nabla f \;=\; \begin{pmatrix} \dfrac{\partial f}{\partial x_1} \\ \vdots \\ \dfrac{\partial f}{\partial x_n} \end{pmatrix}\,. \tag{1.7}$$

From (1.6) we see that if $\nabla f(x) \neq 0$, then we can take a small step in the direction given by $-\nabla f(x)$ and reach a smaller $f$-value. This implies that the gradient must be zero at a minimizer,

$$\nabla f(\hat{x}) \;=\; 0\,. \tag{1.8}$$

The direction given by $-\nabla f(x)$ is known as the *steepest descent direction*.

**Example 1.3.** Equation (1.8) is a necessary condition for $\hat{x}$ to be a local minimizer. It is not sufficient, however, since the gradient is also zero at a local maximizer and at so-called saddle points.

The function from Example 1.2 has a local maximum at $x = \begin{pmatrix} -0.269 & -0.938 \end{pmatrix}^T$ and four saddle points at

$$\begin{pmatrix} 0.086 \\ 2.883 \end{pmatrix}\,, \qquad \begin{pmatrix} 3.387 \\ 0.063 \end{pmatrix}\,, \qquad \begin{pmatrix} -0.131 \\ -1.938 \end{pmatrix}\,, \qquad \begin{pmatrix} -3.074 \\ -0.085 \end{pmatrix}\,.$$

∎

## 2. ITERATIVE SOLUTION

A linear least squares problem with a full rank matrix $A$ has a unique minimizer, which can be found by solving the normal equations. A nonlinear problem may have several local minimizers (as discussed above), and we need to use an *iterative method* to find any of them. This means that we have to give a starting guess $x^{[0]}$ and use an algorithm to compute $x^{[1]}$, $x^{[2]}, \ldots$, that (hopefully) converge to a local minimizer $\hat{x}$.

The iterative process should stop when we are sufficiently close to a minimizer. The *stopping criteria* should include the following three,

$$\|\nabla f(x^{[k]})\|_\infty \;\leq\; \varepsilon_1\,, \tag{2.1a}$$

$$\|x^{[k]}) - x^{[k-1]})\|_2 \leq \delta\,. \tag{2.1b}$$

$$k \;\geq\; k_{\max}\,, \tag{2.1c}$$

where $\varepsilon_1$ and $\delta$ are small, positive numbers, and $k_{\max}$ is a positive integer, eg 100. These three numbers are chosen by the user. The first criterion expresses that the necessary condition in (1.8) is considered to be sufficiently well satisfied. Criterion (2.1b) comes into effect if the function is

very flat around the minimum, and 2.1c) is a "safe guard" against an infinite loop, which might be caused by errors in the implementation of $\nabla f(x)$ or by having chosen $\varepsilon$ and $\delta$ so small that rounding errors make it impossible to satisfy (2.1a) or (2.1b).

The criterion (2.1b) is sometimes implemented as

$$\|x^{[k]}) - x^{[k-1]})\|_2 \;\le\; \varepsilon_2 \left( \|x^{[k]}\|_2 + \varepsilon_2 \right) \;. \tag{2.1b'}$$

This gives a gradual change from desired absolute accuracy $\delta = \varepsilon_2^2$ when all components in $\hat{x}$ are close to zero, to desired relative accuracy $\delta \simeq \varepsilon_2 \|\hat{x}\|_2$ when $\hat{x}$ has large components.

In the remainder of the note we simplify notation by letting $x$ denote the current iterate, $x = x^{[k]}$, and letting $h$ denote the $k^{\text{th}}$ step, $h = x^{[k+1]} - x^{[k]}$.

It is outside the scope of this introductory note to give a general description of methods for computing minimizers of general objective functions. We restrict ourselves to discussing a robust and efficient method for finding a local minimizer for least squares problems where each $r_i(x)$ is continuously differentiable with respect to every $x_j$. Then Taylor's theorem tells us that

$$r_i(x+h) \;=\; r_i(x) + \sum_{j=1}^{n} \frac{\partial r_i}{\partial x_j}(x) h_j + O(\|h\|^2) \;, \quad i = 1, \ldots, m \;.$$

This can be written

$$r(x+h) \;\simeq\; r(x) + J(x)h \;, \tag{2.2}$$

where the $m \times n$ matrix $J(x)$ is the *Jacobian*

$$J(x) \;=\; \begin{pmatrix} \dfrac{\partial r_1}{\partial x_1}(x) & \cdots & \dfrac{\partial r_1}{\partial x_n}(x) \\ \vdots & & \vdots \\ \dfrac{\partial r_m}{\partial x_1}(x) & \cdots & \dfrac{\partial r_m}{\partial x_n}(x) \end{pmatrix} \;. \tag{2.3}$$

Equation (2.2) is the background for the so-called *Gauss–Newton method*: Given the iterate $x$, compute the vector $r(x)$ and matrix $J(x)$. The next iterate is $x+h$, where $h$ is the minimizer of $\|r(x) + J(x)\tilde{h}\|_2$, ie the least squares solution to the overdetermined linear system

$$J(x)\tilde{h} \;\simeq\; -r(x) \;,$$

cf (1.3). The step may be found via the normal equations

$$\big(J(x)^T J(x)\big)\, h \;=\; -J(x)^T r(x) \;. \tag{2.4}$$

Applying well-known rules of differentiation to the function $f$ given in Definition 1.1 we get[1)]

$$\frac{\partial f}{\partial x_j}(x) \;=\; \sum_{i=1}^{m} r_i(x) \frac{\partial r_i}{\partial x_j}(x) \;=\; J(x)^T_{\vdots,j} r(x) \;.$$

[1)] If we had not used the factor $\frac{1}{2}$ in the definition of $f$, we would have got an annoying factor of 2 in a number of expressions.

This shows that the gradient of $f$ can be expressed by

$$\nabla f(x) \;=\; J(x)^T r(x)\;, \tag{2.5}$$

so the left hand side in the stopping criterion (2.1a) does not need extra function evaluation.

**Example 2.1.** The function from Example 1.2 has the Jacobian

$$J(x) \;=\; \begin{pmatrix} 2x_1 & 1 \\ 1 & 2x_2 \\ 0 & -0.2 \end{pmatrix}\;.$$

Figure 2.1 shows the iteration paths when we take 5 steps with the Gauss–Newton method from four different starting points,

$$\begin{pmatrix} 5 \\ 5 \end{pmatrix}\;, \qquad \begin{pmatrix} -1 \\ -5 \end{pmatrix}\;, \qquad \begin{pmatrix} 1 \\ -5 \end{pmatrix}\;, \qquad \begin{pmatrix} -1 \\ 1 \end{pmatrix}\;.$$

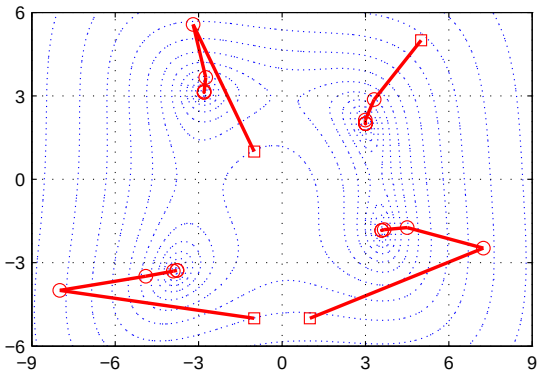



**Figure 2.1.** *Iteration paths with the Gauss-Newton method.*

We see that by proper choice of the starting point we may end in any of the four local minimizers. Often, but not always, we land in the minimizer closest to the starting point. The plot also shows that we may have to take a detour between the starting point and the minimizer found.

Inserting (2.5) in (2.4) we get

$$\big(J(x)^T J(x)\big)\, h \;=\; -\nabla f(x)\;.$$

This shows that if we start in a local maximizer or in a saddle point, then the step is $h = 0$, ie we do not move. The iterations will be stopped by criterion (2.1a). ∎

# 3. THE LEVENBERG–MARQUARDT METHOD

The Taylor expansion (2.2) is a good approximation to $r(x+h)$ when $\|h\|$ is small. This means that the Gauss–Newton method is a good choice in the final stage of the process, where we can get from the current iterate $x$ to $\hat{x}$ by taking a small step. In the initial stage, however, we want to be able to take larger steps, but not too large. This can eg be achieved by using the *Levenberg–Marquardt method*, where (2.4) is replaced by

$$\big(J(x)^T J(x) + \mu I\big)\, h \;=\; -J(x)^T r(x)\;, \tag{3.1}$$

where $I$ is the identity matrix. This is a so-called *damped method*, with the *damping parameter* $\mu > 0$. The damping parameter has several effects:

a) Even if the columns in $J(x)$ are linearly dependent, the coefficient matrix is positive definite for all $\mu > 0$, cf Appendix. This implies that the matrix is nonsingular, and $h$ is unique.

b) For large values of $\mu$ we get

$$h \;\simeq\; -\frac{1}{\mu}\,\nabla f(x)\;,$$

ie a short step in the steepest descent direction. This is good if the current iterate is far from the solution.

c) If $\mu$ is very small, then the step given by (3.1) is almost identical to the Gauss–Newton step given by (2.4). This is a good step in the final stage of the iteration, when $x$ is close to $\hat{x}$.

Thus, the damping parameter influences both the direction and the size of the step. During the iterative process an assessment of the quality of the approximation in (2.2) is used to update $\mu$: More specifically, we use the *gain ratio*

$$\varrho \;=\; \frac{\Delta_a}{\Delta_p}\;, \tag{3.2}$$

where $\Delta_a$ and $\Delta_p$ are respectively the actual and the predicted gain in objective function value,

$$\begin{aligned}
\Delta_a \;&=\; f(x) - f(x+h) \;=\; \tfrac{1}{2} r(x)^T r(x) - \tfrac{1}{2} r(x+h)^T r(x+h)\;,\\
\Delta_p \;&=\; \tfrac{1}{2} r(x)^T r(x) - \tfrac{1}{2}\left(r(x) + J(x)h\right)^T \left(r(x) + J(x)h\right)\\
&=\; -\tfrac{1}{2} h^T \big(2J(x)^T r(x) + J(x)^T J(x) h\big)\\
&=\; -\tfrac{1}{2} h^T \big(2\nabla f(x) - \nabla f(x) - \mu h\big) \;=\; \tfrac{1}{2} h^T \big(\mu h - \nabla f(x)\big)\;.
\end{aligned} \tag{3.3}$$

According to remark a) above the matrix in (3.1) is positive definite. Using this and (2.5) we see that if $\nabla f(x) \neq 0$, then $h \neq 0$ and

$$-h^T \nabla f(x) \;=\; h^T \left(J(x)^T J(x) + \mu I\right) h \;>\; 0\;.$$

Also, $\mu\, h^T h > 0$, so the predicted gain $\Delta_p$ is positive. If the step is too large, then the actual gain $\Delta_a$ may be considerably smaller; it may even be negative. Therefore, a small value of the gain ratio $\varrho$ indicates that $h$ was too big and we need larger damping. In contrast, $\varrho \simeq 1$ indicates that $h$ is small enough for (2.2) to be a very good approximation, and we may decrease $\mu$. This is also done if $\varrho > 1$.

This use of adaptive damping in connection with the Gauss–Newton method has proven ti be robust and efficient. It is known as the *Levenberg–Marquardt method*. Marquardt (1963) proposed the following, widely used strategy,

$$\begin{aligned} &\textbf{if } \varrho < 0.25 \textbf{ then } \mu := \mu * 2 \\ &\textbf{elseif } \varrho > 0.75 \textbf{ then } \mu := \mu/3 \end{aligned} \tag{3.4}$$

In Algorithm 3.6 below we use a slightly modified strategy.

The choice of initial $\mu$-value should be related to the values of the elements in $M^{[0]} = J(x^{[0]})^T J(x^{[0]})$, eg by letting

$$\mu^{[0]} = \tau \cdot \max_{ij} \left| \left( M^{[0]} \right)_{ij} \right| , \tag{3.5}$$

where $\tau$ is chosen by the user.[1)]

The Levenberg–Marquardt algorithm is summarized below.

**Algorithm 3.6. The Levenberg–Marquardt method**

**begin**
  $k := 0; \quad x := x^{[0]}; \quad \nu = 2;$ $\{1^\circ\}$
  $M := J(x)^T J(x); \quad g := J(x)^T r(x)$
  Compute $\mu$ by (3.5); STOP := ( $\|g\|_\infty \le \varepsilon_1$ ) $\{2^\circ\}$
  **while** ( **not** STOP ) **and** ( $k < k_{\max}$ )
    $k := k+1;$ Solve $(M + \mu I)h = -g$
    **if** $\|h\|_2 \le \varepsilon_2(\|\boldsymbol{x}\|_2 + \varepsilon_2)$ **then** $\{2^\circ\}$
      STOP := **true**
    **else**
      $\varrho := \Delta_a/\Delta_p$ $\{3^\circ\}$
      **if** $\varrho > 0$ **then** $\{4^\circ\}$
        $x := x + h; \quad M := J(x)^T J(x); \quad g := J(x)^T r(x)$
        STOP := $(\|g\|_\infty \le \varepsilon_1)$
        $\mu := \mu \cdot \max\{1 - (2\varrho - 1)^3, \frac{1}{3}\}; \quad \nu := 2$ $\{6^\circ\}$
      **else** $\{5^\circ\}$
        $\mu := \mu \cdot \nu; \quad \nu := 2 \cdot \nu$ $\{6^\circ\}$
      **end**
    **end**
  **end**
**end**

We have the following remarks

$1^\circ$ Initialization. See $6^\circ$ about $\nu$,

$2^\circ$ We use the stopping criteria (2.1a), (2.1b') and (2.1c).

[1)] The algorithm is not very sensitive to the choice of $\tau$, but as a rule of thumb, one should use a small value, eg $\tau = 10^{-6}$, if $x^{[0]}$ is believed to be a good approximation to $\hat{x}$. Otherwise, use $\tau = 10^{-3}$ or even $\tau = 1$.

3° Gain ratio, cf (3.2). In order to reduce the effect of cancellation errors, the actual gain should be computed as $\frac{1}{2}\big(r(\boldsymbol{x}) - r(x+h)\big)^T\big(r(\boldsymbol{x}) + r(x+h)\big)$, and we use (3.3) to compute the predicted gain.

4° $f(x+h) < f(x)$. Update $x$ and $\mu$.

5° Uphill step, $f(x+h) \geq f(x)$. Don't change $x$, but increase $\mu$.

6° Update $\mu$. In Nielsen (1999) we showed that the jumps in the scheme given in (3.4) may give rise to a "flutter" that slows down convergence. The continuous change when the gain factor is positive generally gives a better performance. The factor $\nu$ speeds up the growth of the damping coefficient in case of consecutive uphill steps. This may eventually lead to a faster stopping of the process by means of criterion (2.1b').

**Example 3.1.** We have used Algorithm 3.6 on the problem from Example 1.2, with the same starting points as in Example 2.1. We used $\tau = 10^{-3}$ and the stopping criteria (2.1) with

$$\varepsilon_1 \;=\; 10^{-8}\,, \quad \varepsilon_2 \;=\; 10^{-12}\,, \quad k_{\max} \;=\; 100\,.$$

Figure 3.1 shows the iteration paths. An uphill step is indicated by a dashed line with a star at $x+h$

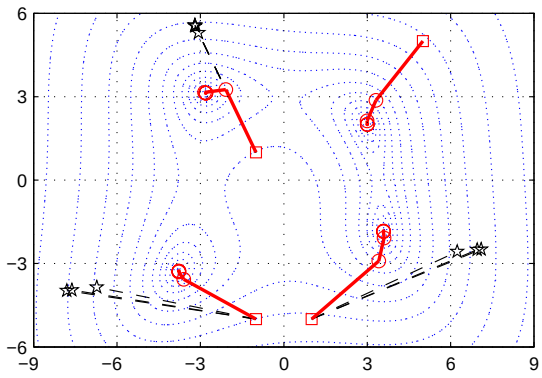



**Figure 3.1.** *Iteration paths with the Levenberg–Marquardt method.*

It is easy to reach the minimizer $\begin{pmatrix} 3 & 2 \end{pmatrix}^T$ from the starting point $\begin{pmatrix} 5 & 5 \end{pmatrix}^T$, and the criterion (2.1b') stopped the process after 5 iterations. The other starting points are rather poor, and we use 9 – 10 iterations. If we use a larger value for $\tau$, we can avoid some of the initial uphill steps. ∎

**Example 3.2.** We look at the data fitting problem from Example 1.1. The front page figure shows $m = 45$ data points and the least squares fit with the model

$$M(x,t) \;=\; x_1 e^{x_3 t} + x_2 e^{x_4 t}\,.$$

The $i$th residual and the $i$th row in the Jacobian are

$$r_i(x) = y_i - \big(x_1 e^{x_3 t_i} + x_2 e^{x_4 t_i}\big)\,, \qquad J(x)_{i,:} = \begin{pmatrix} e^{x_3 t_i} & e^{x_4 t_i} & x_1 t_i e^{x_3 t_i} & x_2 t_i e^{x_4 t_i} \end{pmatrix}\,.$$

We use the Levenberg–Marquardt method with $\tau = 10^{-2}$, $\varepsilon_1 = 10^{-8}$, $\varepsilon_2 = 10^{-12}$, $k_{\max} = 100$, and a rather poor initial guess, $x^{[0]} = \begin{pmatrix} 0 & 0 & -1 & -2 \end{pmatrix}^T$. It takes 67 iterations to find the solution. To three decimals accuracy this is $\hat{x} = \begin{pmatrix} 4 & -4 & -4 & -5 \end{pmatrix}^T$. Figure 3.2 illustrates the performance

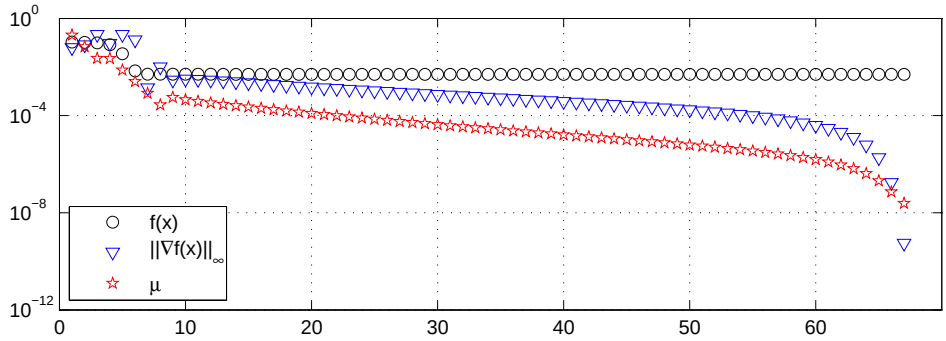



**Figure 3.2.** *Performance parameters.*

In iterations 10 – 55 the $\nabla f$-values show a steady (linear) convergence, and in the last about 10 iterations there is faster (superlinear) convergence. The value of $f(x)$ is almost constant after the $10^{th}$ iteration, although $x^{[10]} \simeq \begin{pmatrix} 1.9 & -1.9 & -3.6 & -5.7 \end{pmatrix}^T$ is quite far from $\hat{x}$. This means that the minimum is very flat.

The front page figure suggests that the data fitting model should satisfy $M(x, 0) = 0$. Therefore, it seems a good idea to replace the model by $M(x, t) = x_1 \left(e^{x_2 t} - e^{x_3 t}\right)$. We use the same stopping criteria as above and the equivalent poor starting guess $x^{[0]} = \begin{pmatrix} 0 & -1 & -2 \end{pmatrix}^T$, and we get the equivalent solution $\hat{x} = \begin{pmatrix} 4 & -4 & -5 \end{pmatrix}^T$ after 53 iterations. Thus, we save approximately 20% by using the simpler model. ∎

**Example 3.3.** The Gauss-Newton step is defined by (2.4). This is the normal equations for the overdetermined system of equations $r(x) + J(x)h \simeq 0$. Similarly,the L-M equations (3.1) are the normal equations for the system

$$\begin{pmatrix} r(x) \\ 0 \end{pmatrix} + \begin{pmatrix} J(x) \\ \sqrt{\mu} I \end{pmatrix} h \simeq 0 \,.$$

In Section 8.15 of Eldén et al (2004) it is discussed why the most accurate solution of the system is computed via orthogonal transformation applied to the overdetermined system. However, the solution $h$ is just a step in an iterative process, and needs not be computed very accurately. The solution via the normal equations is "cheaper", and therefore this method is normally employed. ∎

## APPENDIX. SYMMETRIC, POSITIVE DEFINITE MATRICES

The matrix $A \in \mathbb{R}^{n\times n}$ is symmetric if $A = AT$, ie if $a_{ij} = a_{ji}$ for all $i, j$.

**Definition A.1.** The symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is

*positive definite* if $x^T A\, x > 0$ for all $x \in \mathbb{R}^n,\ x \neq \mathbf{0}$ ,

*positive semidefinite* if $x^T A\, x \geq 0$ for all $x \in \mathbb{R}^n,\ x \neq \mathbf{0}$ .

Some useful properties of such matrices are listed in Theorem A.2 below. The proof can be found by combining theorems in almost any textbook on linear algebra and on numerical linear algebra.

**Theorem A.2.** Let $A \in \mathbb{R}^{n\times n}$ be symmetric and let $A = \boldsymbol{LU}$, where $\boldsymbol{L}$ is a unit lower triangular matrix and $\boldsymbol{U}$ is an upper triangular matrix. Further, let $\{(\lambda_j, \boldsymbol{v}_j)\}_{j=1}^n$ denote the eigensolutions of A,

$$A\boldsymbol{v}_j \ = \ \lambda_j \boldsymbol{v}_j \ , \quad j = 1, \ldots, n \ . \tag{A.3}$$

Then

1° The eigenvalues are real, $\lambda_j \in \mathbb{R}$, and the eigenvectors $\{\boldsymbol{v}_j\}$ form an orthonormal basis of $\mathbb{R}^n$.

2° The following statements are equivalent

a) A is positive definite (positive semidefinite)

b) All $\lambda_j > 0$ ( $\lambda_j \geq 0$ )

c) All $u_{ii} > 0$ ( $u_{ii} \geq 0$ ) .

If A is positive definite, then

3° The LU-factorization is numerically stable.

4° $A = \boldsymbol{C}^T \boldsymbol{C}$, the *Cholesky factorization.* $\boldsymbol{C} \in \mathbb{R}^{n\times n}$ is upper triangular.

Now, let $J \in \mathbb{R}^{m\times n}$ be given, and let

$$A = J^T J \ .$$

Then $AT = J^T (J^T)^T = A$, ie A is symmetric. Further, for any nonzero $x \in \mathbb{R}^n$ let $y = Jx$. Then

$$x^T A\, x = x^T J^T J\, x = y^T y \geq 0 \ ,$$

showing that A is positive semidefinite. If $m \geq n$ and the columns in J are linearly independent, then $x \neq \mathbf{0} \ \Rightarrow \ y \neq \mathbf{0}$ and $y^T y > 0$. Thus, in this case A is positive definite.

From (A.3) it immediately follows that

$$(A + \mu \boldsymbol{I})\boldsymbol{v}_j = (\lambda_j + \mu)\boldsymbol{v}_j \ , \quad j = 1, \ldots, n$$

for any $\mu \in \mathbb{R}$. Now, let A be symmetric, but not necessarily positive (semi)definite, and let the eigenvalues be numbered so that

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \ .$$

Further, let $\alpha = \max\{-\lambda_1, 0\}$. Then it follows that $h = A + \mu \boldsymbol{I}$ is also symmetric, and 2° in Theorem A.2 shows that h is guaranteed to be positive definite for all $\mu > \alpha$.

The 2-norm *condition number* of a symmetric matrix A is

$$\kappa_2(A) = \max\{|\lambda_j|\} / \min\{|\lambda_j|\} \ .$$

If A is positive (semi)definite and $\mu > 0$, then

$$\kappa_2(A+\mu \boldsymbol{I}) \;=\; \frac{\max\{\lambda_j\}+\mu}{\min\{\lambda_j\}+\mu} \;=\; \frac{\lambda_n+\mu}{\lambda_1+\mu} \;\leq\; \frac{\max\{\lambda_j\}+\mu}{\mu} \;\;,$$

and this is a decreasing function of $\mu$.

## REFERENCES


1. L. Eldén, L. Wittmeyer-Koch and H.B. Nielsen (2004): *Introduction to Numerical Computation – analysis and* MATLAB *illustrations*, Studentlitteratur.
2. K. Levenberg (1944): *A Method for the Solution of Certain Problems in Least Squares.* Quart. Appl. Math. **2**, pp 164–168.
3. D. Marquardt (1963): *An Algorithm for Least Squares Estimation on Nonlinear Parameters.* SIAM J. APPL. MATH.**11**, pp 431–441.
4. H.B. Nielsen (1999): *Damping Parameter in Marquardt's Method*, IMM, DTU, Report IMM-REP-1999-05. Available as http://www.imm.dtu.dk/∼hbn/publ/TR9905.ps