Lecture Notes

Numerical Optimization

Winter Term 2025/2026

Michael C. Lehn

Institute of Numerical Mathematics
Ulm University

Contents

List of Algorithms

Listings

Chapter 1 Introduction

The problems considered in this chapter arise in practical applications where measurements are affected by noise and models are nonlinear. In general, such problems are not solvable in a strict mathematical sense. However, they do not arise arbitrarily but have a specific structure and origin that can and must be exploited.

Numerical mathematics provides the framework to deal with such situations. By using approximation, iteration, and careful analysis, it becomes possible to obtain practically useful results even when an exact solution cannot be found.

This is not a compromise of mathematics—it is its frontier. Numerical mathematics is, in this sense, the spearhead of mathematics.

Chapter 2 Nonlinear Least Squares Problems

In the course on Numerical Linear Algebra, we studied the *linear* least squares problem, that is, minimization problems of the form

$$\|Ax - b\|_2^2 \;\to\; \min$$

for a full-rank matrix $A \in \mathbb{R}^{m\times n}$ and a vector $b \in \mathbb{R}^m$ with $m \geq n$. Defining

$$f : \mathbb{R}^n \to \mathbb{R}, \qquad f(x) = \|Ax - b\|_2^2,$$

we saw that

$$f(\widehat{x}) = \min_{x\in\mathbb{R}^n} f(x) \quad \Longleftrightarrow \quad \nabla f(\widehat{x}) = \mathbf{0}.$$

Because f is convex in x, this necessary condition is also sufficient for a global minimum. Since

$$\nabla f(x) = 2A^T(Ax - b),$$

the minimizer $\widehat{x}$ satisfies the so-called *normal equations*

$$A^T Ax = A^T b.$$

In the Numerical Linear Algebra course, the emphasis was on algorithms suitable for solving these equations efficiently and stably. For example, the Cholesky decomposition exploits the fact that $A^T A$ is symmetric and positive definite, while the QR decomposition allows one to compute the solution without explicitly forming $A^T A$, which improves numerical stability.

In the following, we will generalize the problem by allowing the objective function f to be nonlinear in x. This is highly relevant in practice, since many interesting problems fall into this category. From a theoretical perspective, however, nonlinear least squares problems are considerably more challenging.

§1 Problem Statement

In this chapter, we consider objective functions

$$f : \mathbb{R}^n \to \mathbb{R}$$

of the special form

$$f(x) = \frac{1}{2}\sum_{i=1}^{m} F_i(x)^2,$$

where the functions $F_i : \mathbb{R}^n \to \mathbb{R}$, $1 \leq i \leq m$, are assumed to be twice continuously differentiable. Equivalently, we can write

$$f(x) = \frac{1}{2}F(x)^T F(x) = \frac{1}{2}\|F(x)\|_2^2, \qquad F(x) = \begin{pmatrix} F_1(x) \\ \vdots \\ F_m(x) \end{pmatrix}.$$

To facilitate later reference, we summarize this specific class of problems in the following definition.

Definition 2.1.1 (Least Squares Problem)
Let $F_1, \ldots, F_m \in C^2(\mathbb{R}^n; \mathbb{R})$ with $m, n \geq 1$, and define

$$f : \mathbb{R}^n \to \mathbb{R}, \qquad f(x) = \frac{1}{2}\sum_{i=1}^{m} F_i(x)^2.$$

A point $\widehat{x} \in \mathbb{R}^n$ is called a (global) solution of the *least squares problem* if

$$f(\widehat{x}) = \min_{x \in \mathbb{R}^n} f(x).$$

□

Having defined the general problem, we now verify that it indeed represents a genuine generalization of the least squares problem encountered in the course on Numerical Linear Algebra.

Example 2.1.2 (Linear least squares as a special case)
Let $a_1, \ldots, a_m \in \mathbb{R}^n$ and $b_1, \ldots, b_m \in \mathbb{R}$, and define

$$F_i(x) = a_i^T x - b_i, \qquad 1 \leq i \leq m.$$

Then

$$\begin{aligned} f(x) &= \frac{1}{2}\sum_{i=1}^{m}(a_i^T x - b_i)^2 \\ &= \frac{1}{2}\|Ax - b\|_2^2, \end{aligned}$$

where

$$A = \begin{pmatrix} a_1^T \\ \vdots \\ a_m^T \end{pmatrix}, \qquad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}.$$

Clearly,

$$f(\widehat{x}) = \min_{x \in \mathbb{R}^n} f(x) \quad \Longleftrightarrow \quad \|A\widehat{x} - b\|_2^2 = \min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2.$$

Thus, this special case is equivalent to the familiar *linear least squares problem*. If $m \geq n$ and A has full column rank, the minimizer is unique.

Hence, the nonlinear least squares problem truly generalizes the classical linear one: the objective function differs only by the factor $\frac{1}{2}$, which is introduced merely for convenience, as it simplifies the expression for the necessary condition of optimality (see (2.1) below). □

Despite the rather strong assumption that the objective function f is twice continuously differentiable and, by construction, bounded below by 0, no general statement can be made about the existence (or even uniqueness) of a solution.

Example 2.1.3
Let $m = n = 1$.

(a) For $F(x) = F_1(x) = 1 - x^2$ we have

$$f(x) = \frac{1}{2}F(x)^2 = \frac{1}{2}(1 - x^2)^2.$$

Clearly, $f(x) = 0$ if and only if $x = \pm 1$. Hence, a solution exists, but it is not unique.

(b) For $F(x) = F_1(x) = \frac{1}{1+x^2}$ we obtain

$$f(x) = \frac{1}{2(1 + x^2)^2} > 0 \quad \text{for all } x \in \mathbb{R}.$$

Since

$$\lim_{x\to\infty} f(x) = 0,$$

no global minimizer exists.

□

If a solution $\widehat{x}$ exists, that is, if the global minimum of f is attained, then $\widehat{x}$ is in particular also a local minimum. A necessary condition for a local minimum (or, more generally, for a stationary point) is that the gradient vanishes:

$$f'(\widehat{x}) = \nabla f(\widehat{x})^T = \mathbf{0}^T. \tag{2.1}$$

A sufficient condition for a local minimum is that

$$\begin{aligned} &\nabla f(\widehat{x}) = \mathbf{0} \quad \text{and} \quad r^T H_f(\widehat{x}) r > 0 \quad \forall r \neq \mathbf{0} \\ &\Rightarrow \ \exists \varepsilon > 0 : f(x) > f(\widehat{x}) \quad \forall x : \ \|x - \widehat{x}\|_2 < \varepsilon. \end{aligned} \tag{2.2}$$

For completeness, these well-known results from analysis are reviewed in Appendix §1. That appendix also summarizes the notation and conventions used throughout this text, such as ∇f for the gradient (the $n \times 1$ Jacobian matrix) and H_f for the Hessian matrix.

However, the sufficient condition for a local minimum given in (2.2) is not always conclusive, as the following example shows.

Example 2.1.4 (Stationary points and higher derivatives)
Let $m = n = 1$.

(a) For $F_1(x) = x^2$ we have

$$f(x) = \tfrac{1}{2}F_1(x)^2 = \tfrac{1}{2}x^4,$$

and $\widehat{x} = 0$ is clearly the point of the global minimum. Here

$$f'(0) = f''(0) = f^{(3)}(0) = 0, \qquad f^{(4)}(0) = 4! > 0.$$

In general, if f is $2k$-times continuously differentiable and

$$f'(\widehat{x}) = f''(\widehat{x}) = \cdots = f^{(2k-1)}(\widehat{x}) = 0, \qquad f^{(2k)}(\widehat{x}) > 0,$$

then $\widehat{x}$ is a local minimizer.

For $n > 1$, $f''(\widehat{x})$ is a matrix (the Hessian). Checking its definiteness is numerically feasible, as known from the course on Numerical Linear Algebra. We also know the computational cost associated with this task. Higher derivatives (3rd, 4th, etc.) would be tensors. In principle, similar sufficient conditions could be formulated using them, but the computational effort grows rapidly. Since we only assume $f \in C^2(\mathbb{R}^n)$, we will not pursue this idea further, as higher derivatives may not even exist.

(b) Even if we allow f to be infinitely often differentiable, examining higher derivatives alone may not suffice. Consider the classical example

$$F_1(x) = \begin{cases} e^{-1/x^2}, & x \neq 0, \\ 0, & x = 0. \end{cases}$$

Then F_1 is infinitely differentiable, and $x = 0$ is a global minimum of F_1, but

$$F_1^{(k)}(0) = 0 \quad \text{for all } k \in \mathbb{N}.$$

The same holds for

$$f(x) = \tfrac{1}{2} F_1(x)^2,$$

where all derivatives vanish at $x = 0$, even though $x = 0$ is a strict global minimizer. Thus, higher-order derivatives alone do not necessarily provide sufficient information to identify local minima.

□

For a numerical method to be considered feasible, it is therefore reasonable to require merely that it produces stationary points of the objective function. Whether a stationary point corresponds to a local minimum can then be examined separately.

However, even this task is far from trivial. A conceivable strategy for finding the solution would be to determine all local minima and then select the one with the smallest function value. For such an approach to be theoretically feasible, it must at least be guaranteed that there exist only finitely many local minima.

To illustrate what "not feasible" may look like in practice and to appreciate the inherent difficulty of the problem, consider the following example.

Example 2.1.5 (Infinitely many non-global local minima)
Let

$$f(x) = \sin^2(x) + \sin^2(x^2).$$

Then $f(0) = 0$ and $f(x) > 0$ for all $x \neq 0$, so $\widehat{x} = 0$ is the unique global minimizer. Nevertheless, f possesses infinitely many local minima. This can already be seen intuitively from the graph of f (see Figure 2.1), whose oscillations become increasingly dense for large values of $|x|$.

A proof of this statement is given at the end of this section in Remark 2.1.6. Before that, we discuss what such examples imply for numerical treatment. □

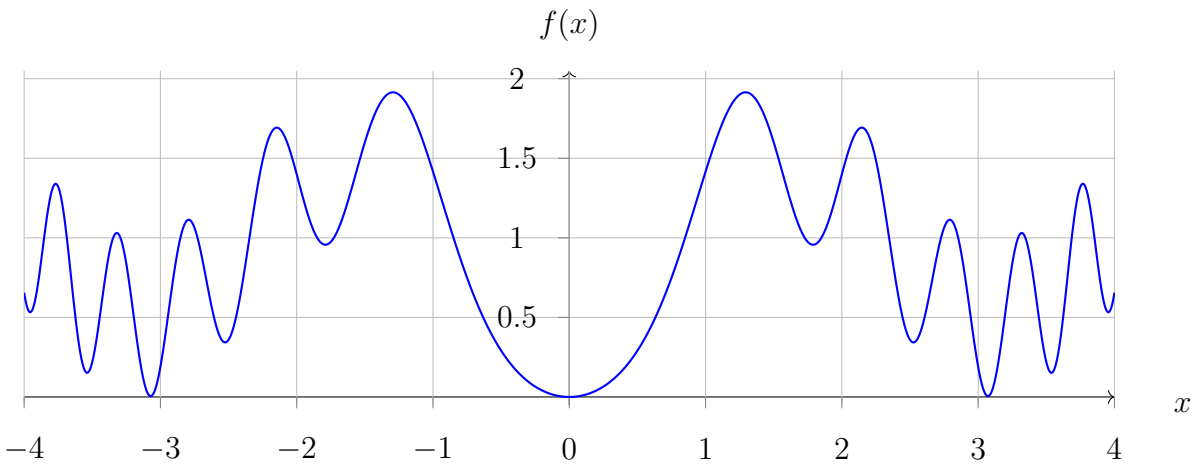


Abbildung 2.1: *Graph of* $f(x) = \sin^2(x) + \sin^2(x^2)$ *on* $[-4, 4]$.

In situations like this, a numerical optimization method can at best be expected to generate a sequence of local minimizers converging towards the global one. In general, however, the local minimizers cannot be determined analytically but only numerically. Hence, even finding a single local minimizer requires a numerical method that, under suitable assumptions, produces a sequence converging to a local stationary point. Whether such a stationary point corresponds to a local minimum, a saddle point, or a local maximum can, in general, only be determined *a posteriori*.

One should therefore not expect numerical optimization to provide a magical black box capable of solving such problems automatically. After this somewhat sobering realization, it is worth emphasizing that there are still good reasons to be optimistic: objective functions arising in applications do not fall from the sky. In practice, we usually know much more about them than merely that they are twice continuously differentiable. What we can develop is therefore not a single universal algorithm that solves all problems, but rather a toolbox of methods that—under appropriate assumptions—can handle specific subclasses of problems efficiently. Using this toolbox effectively requires understanding the underlying methods and their limitations, in particular the conditions under which they are guaranteed to work.

Before we turn to an application example in the next section, we give here the proof promised in Example 2.1.5.

Remark 2.1.6 (Proof of Example 2.1.5)

We consider

$$f(x) = \sin^2 x + \sin^2 x^2,$$

with

$$\begin{aligned} f'(x) &= 2 \sin x \cdot \cos x + 2 \sin x^2 \cdot \cos x^2 \cdot 2x \\ &= \sin 2x + 2x \sin 2x^2. \end{aligned}$$

We show the following three statements:

(a) $x = 0$ *is a global minimizer.*

Since both terms in $f(x)$ are nonnegative and vanish simultaneously only for $x = 0$,

we have

$$f(x) \geq 0 \quad \text{for all } x \in \mathbb{R}, \qquad f(0) = 0.$$

Hence $\widehat{x} = 0$ is a global minimizer with $f(\widehat{x}) = 0$.

(b) *$x = 0$ is the unique global minimizer.*
The equality $f(x) = 0$ requires $\sin x = 0$ and $\sin x^2 = 0$ simultaneously. This means

$$x = k\pi \quad \text{and} \quad x^2 = n\pi \quad \text{for some } k, n \in \mathbb{Z}.$$

Combining both gives $k^2\pi^2 = n\pi$, i.e. $n = k^2\pi$. As π is irrational, this is impossible for integer n, k unless $k = n = 0$. Therefore $x = 0$ is the only point where $f(x) = 0$, and thus the unique global minimizer.

(c) *f has infinitely many local minima.*
Define

$$a_n = \sqrt{\tfrac{\pi}{4}(4n+1)}, \qquad b_n = \sqrt{\tfrac{\pi}{4}(4n+3)}, \qquad n \in \mathbb{N}.$$

Then

$$\sin 2a_n^2 = 1, \qquad \sin 2b_n^2 = -1.$$

Because

$$b_n > a_n \geq \sqrt{\tfrac{\pi}{4}} > \tfrac{1}{2},$$

we have

$$f'(a_n) = \sin 2a_n + 2a_n > \sin 2a_n + 1 \geq 0,$$
$$f'(b_n) = \sin 2b_n - 2b_n < \sin 2b_n - 1 \leq 0.$$

And since $a_n < b_n < a_{n+1}$, the intermediate value theorem ensures at least one zero of f' in each of the intervals (a_n, b_n) and (b_n, a_{n+1}). Hence f' changes sign infinitely often, and f possesses infinitely many local minima and maxima.

In particular, $x = 0$ is the unique global minimizer, while all other local minima satisfy $f(x) > 0$. □

§2 Test Problems

In the following, three related problems are introduced that will later serve as test cases for numerical methods designed to solve least squares problems. Each problem is constructed such that the first and second derivatives of the objective function can be computed analytically — which, of course, is not generally possible in practical applications.

In all three cases, we are given points $(x_i, y_i)^T \in \mathbb{R}^2$ for $1 \leq i \leq m$, which are assumed to lie approximately on the boundary of a circle with true center $(\overline{c}_x, \overline{c}_y)^T$ and true radius $\overline{r}$, according to an underlying model, but are affected by stochastic noise.

Formally, each observed data point can be expressed as

$$\begin{aligned} x_i &= \overline{x}_i + \varepsilon_{x,i}, \\ y_i &= \overline{y}_i + \varepsilon_{y,i}, \end{aligned} \qquad 1 \leq i \leq m,$$

where the ideal (noise-free) points $(\overline{x}_i, \overline{y}_i)$ are located on the circumference of a circle with true parameters $(\overline{c}_x, \overline{c}_y, \overline{r})$ and satisfy

$$\begin{aligned} \overline{x}_i &= \overline{c}_x + \overline{r}\cos\varphi_i, \\ \overline{y}_i &= \overline{c}_y + \overline{r}\sin\varphi_i, \end{aligned} \qquad 1 \le i \le m.$$

The angles φ_i are independently and uniformly distributed as

$$\varphi_i \overset{\text{iid}}{\sim} \mathcal{U}(0, 2\pi),$$

and the noise terms are independent and identically distributed as

$$\varepsilon_{x,i}, \varepsilon_{y,i} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2).$$

For a single data point $(x_i, y_i)^T$ we define the residual

$$F_i(r, c_x, c_y) \;=\; \sqrt{(x_i - c_x)^2 + (y_i - c_y)^2} \;-\; r, \tag{2.3}$$

and the least-squares objective

$$f(r, c_x, c_y) \;=\; \frac{1}{2}\sum_{i=1}^{m} F_i(r, c_x, c_y)^2. \tag{2.4}$$

If the data are noise-free (formally, $\sigma = 0$), we have

$$f(\overline{r}, \overline{c}_x, \overline{c}_y) = 0.$$

For noisy data with $\sigma > 0$, this equality no longer holds exactly. The expected value of $f(\overline{r}, \overline{c}_x, \overline{c}_y)$ is strictly positive due to the bias introduced by the noise. Nevertheless, for small noise levels, the global minimizer of f will typically be close to the true parameters $\overline{r}, \overline{c}_x$, and $\overline{c}_y$, and can therefore serve as a useful estimator of the unknown true parameters.

We now consider the following three test problems, with objectives $f_{\text{ex1}}, f_{\text{ex2}}, f_{\text{ex3}}$.

- **Example 1 (known center at the origin)**

  The center of the circle is known to be at $(c_x, c_y) = (0, 0)$, while the radius r is unknown. Although negative values of r have no physical meaning, we deliberately allow $r < 0$ so that the problem remains unconstrained. This leads to

  $$f_{\text{ex1}} \colon \mathbb{R} \to \mathbb{R}, \qquad f_{\text{ex1}}(r) := f(r, 0, 0). \tag{2.5}$$

  A particular advantage of this setting is that both f and its derivative f' can be easily visualized.

- **Example 2 (partially known center)**

  The circle center is partially known: $c_y = 0$ is fixed, whereas c_x and r are unknown. This leads to

  $$f_{\text{ex2}} \colon \mathbb{R}^2 \to \mathbb{R}, \qquad f_{\text{ex2}}(r, c_x) := f(r, c_x, 0). \tag{2.6}$$

  In this case, a contour plot of $f(c_x, r)$ can be generated, giving intuitive insight into the structure of the objective function and the distribution of its extrema.

- **Example 3 (unknown center and radius)**
  The most general case, where r, c_x, and c_y are all unknown. This leads to

$$f_{\text{ex3}}\colon \mathbb{R}^3 \to \mathbb{R}, \qquad f_{\text{ex3}}(r, c_x, c_y) := f(r, c_x, c_y). \tag{2.7}$$

  Here no parameters are fixed, making the problem fully nonlinear and three-dimensional.

Listing 2.1 shows the implementation of the function

```
function [x, y] = create_circ_data(m, r_, cx_, cy_, sigma)
```

which generates synthetic test data for the problems described above. Its parameters have the following meaning:

- `m` – number of data points,
- `r_` – true radius, corresponding to $\overline{r}$,
- `cx_`, `cy_` – true circle center, corresponding to $\overline{c}_x$ and $\overline{c}_y$,
- `sigma` – standard deviation of the normally distributed noise.

The function returns two column vectors

$$x = (x_1, \ldots, x_m)^T, \qquad y = (y_1, \ldots, y_m)^T,$$

representing the observed data points.

To build intuition for the three test problems, it is instructive to visualize the noisy data together with the "true" and "estimated" circles. Listing 2.2 shows the implementation of the function

```
function draw_circle(x, y, r_, cx_, cy_, r, cx, cy)
```

Its arguments are:

- `x`, `y` – data vectors as returned by `create_circ_data`,
- `r_`, `cx_`, `cy_` – true circle parameters $(\overline{r}, \overline{c}_x, \overline{c}_y)$,
- `r`, `cx`, `cy` – estimated circle parameters.

The function plots the data points (x_i, y_i), the true circle (in blue, with radius $\overline{r}$ and center $(\overline{c}_x, \overline{c}_y)$), and the estimated circle (in red, with radius r and center (c_x, c_y)). This visualization provides an immediate geometric impression of how well the estimated parameters fit the noisy data.
Figure 2.2 illustrates characteristic aspects of the test problems introduced above. The three subfigures do not visualize all cases (f_{ex3} would be three-dimensional and is therefore difficult to display), but they provide geometric intuition for the first two examples and for the general nature of the least squares objectives.

1. Subfigure 2.2a shows how the noisy data points are scattered around the "true" circle. The data were generated with `[x, y] = create_circ_data(500, 10, 0, 0, 1)` and visualized using `draw_circle(x, y, 10, 0, 0, 10, 0, 0)`.

Listing 2.1: *Function* `create_circ_data`*: Generate synthetic circle data.*

```
function [x, y] = create_circ_data(m, r_, cx_, cy_, sigma)

    arguments
        m (1,1) double {mustBeInteger, mustBeNonnegative}
        r_ (1,1) double {mustBeFinite}
        cx_ (1,1) double {mustBeFinite}
        cy_ (1,1) double {mustBeFinite}
        sigma (1,1) double {mustBeNonnegative} = 1
    end

    if m == 0
        x = zeros(0,1);
        y = zeros(0,1);
        return
    end

    phi   = 2 * pi * rand(m,1);
    noise = sigma * randn(m,2);

    x = cx_ + r_ * cos(phi) + noise(:, 1);
    y = cy_ + r_ * sin(phi) + noise(:, 2);
end
```

Listing 2.2: *Function* `draw_circle`*: Visualize data and fitted circles.*

```
function draw_circle(x, y, r_, cx_, cy_, r, cx, cy)

    plot(x, y, 'bx');
    hold on; axis equal; grid on

    theta = linspace(0, 2 * pi, 400);
    plot(cx_ + r_*cos(theta), cy_ + r * sin(theta), ...
         'b-', 'LineWidth', 1.2);

    theta = linspace(0, 2 * pi, 400);
    plot(cx + r_est*cos(theta), cy + r * sin(theta), ...
         'r-', 'LineWidth', 1.2);

    xlabel('x'), ylabel('y')
    title('Noisy circle data: r = 10, (c_x, c_y) = (0, 0)');

    hold off
end
```

2. Subfigure 2.2b displays the objective $f_{\text{ex1}}(r)$ for $r \in [9.5, 10.5]$. The function is quadratic and therefore convex in r, which can also be shown analytically. The minimum is slightly shifted from $r = 10$, reflecting the influence of the measurement noise.

3. Subfigure 2.2c presents a contour plot of $f_{\text{ex2}}(r, c_x)$ for $r \in [0, 30]$ and $c_x \in [0, 30]$. The contour lines reveal that f_{ex2} is not convex (if it were, all level sets would be convex as well). This nonconvexity can also be established formally. For f_{ex3}, visualization would require plotting two-dimensional manifolds in $\mathbb{R}^3$, which is considerably more challenging.

§3 Derivatives of the Objective Function

In order to formulate numerical methods such as Newton–Raphson, Gauss–Newton, or Levenberg–Marquardt, we require the first and second derivatives of the objective function f. The goal is to express the partial derivatives of f in terms of the derivatives of the residuals $F_1, \dots, F_m$, so that the resulting expressions can be easily adapted to specific least-squares problems.

We use i and j as row and column indices, respectively, and denote the summation index by ℓ. Let

$$f(x) \;=\; \frac{1}{2}\sum_{\ell=1}^{m} F_\ell(x)^2, \qquad x \in \mathbb{R}^n.$$

For the partial derivatives of f, the chain rule gives

$$\frac{\partial f}{\partial x_j}(x) \;=\; \frac{1}{2}\sum_{\ell=1}^{m} 2\,F_\ell(x)\,\frac{\partial F_\ell}{\partial x_j}(x) \;=\; \sum_{\ell=1}^{m} F_\ell(x)\,\frac{\partial F_\ell}{\partial x_j}(x). \tag{2.8}$$

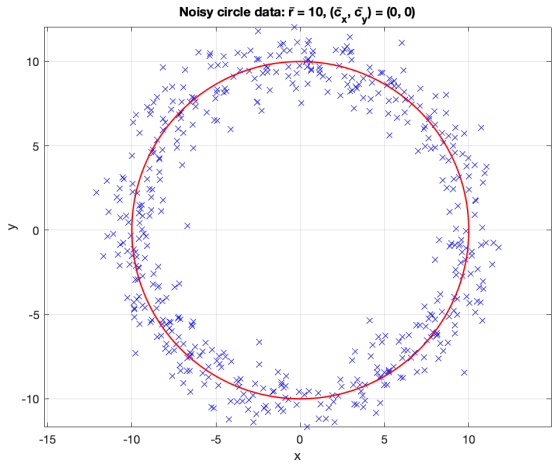
Hence, the gradient is

$$\nabla f(x) \;=\; \begin{pmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{pmatrix} \;=\; \sum_{\ell=1}^{m} F_\ell(x)\,\nabla F_\ell(x) \quad \text{with} \quad \nabla F_\ell(x) := \begin{pmatrix} \frac{\partial F_\ell}{\partial x_1}(x) \\ \vdots \\ \frac{\partial F_\ell}{\partial x_n}(x) \end{pmatrix}. \tag{2.9}$$

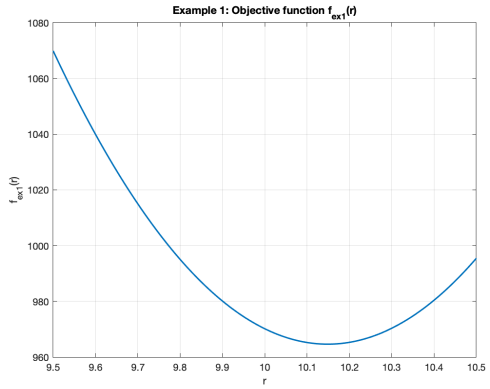
For the second partial derivatives, applying the product rule yields

$$\frac{\partial^2 f}{\partial x_i\,\partial x_j}(x) \;=\; \sum_{\ell=1}^{m}\left(\frac{\partial F_\ell}{\partial x_i}(x)\,\frac{\partial F_\ell}{\partial x_j}(x) \;+\; F_\ell(x)\,\frac{\partial^2 F_\ell}{\partial x_i\,\partial x_j}(x)\right). \tag{2.10}$$

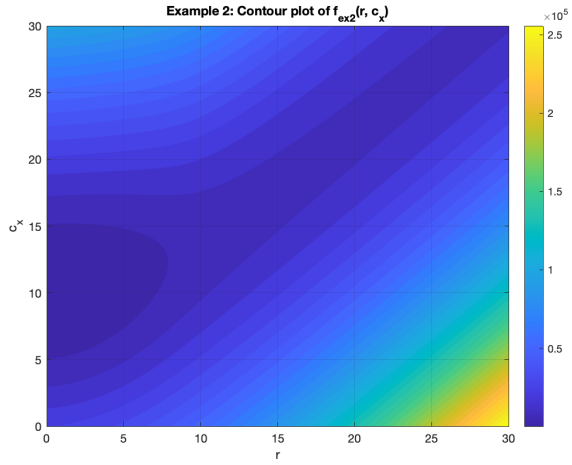
This is the entry of the Hessian matrix $H_f(x)$ in row i and column j. To write this more compactly, it is helpful to recognize how the collection of all products $\frac{\partial F_\ell}{\partial x_i}(x)\,\frac{\partial F_\ell}{\partial x_j}(x)$ can



(a) *Generated data using* `[x, y] = create_circ_data(500, 10, 0, 0, 1)` *and visualized with* `draw_circle(x, y, 10, 0, 0, 10, 0, 0)`*. The noisy points (black) are distributed around the true circle (blue).*



(b) *Plot of* $f_{ex1}(r)$ *for* $r \in [9.5, 10.5]$*. The objective is convex in* r*, though the minimum is slightly shifted from the true radius* $r = 10$ *due to noise.*



(c) *Contour plot of* $f_{ex2}(r, c_x)$ *for* $r \in [0, 30]$ *and* $c_x \in [0, 30]$*. The objective is non-convex, showing multiple local minima.*

Abbildung 2.2: *Illustrations of the test problems. (a) Generated circle data with noise, (b) convex objective* $f_{ex1}(r)$*, and (c) non-convex objective* $f_{ex2}(r, c_x)$*.*

be represented as a single matrix expression. Consider

$$\begin{aligned}\begin{pmatrix}\nabla F_1 & \dots & \nabla F_m\end{pmatrix}\begin{pmatrix}\nabla F_1^\top \\ \vdots \\ \nabla F_m^\top\end{pmatrix} &= \sum_{\ell=1}^{m} \nabla F_\ell \, \nabla F_\ell^\top = \sum_{\ell=1}^{m} \begin{pmatrix}\frac{\partial F_\ell}{\partial x_1}(x) \\ \vdots \\ \frac{\partial F_\ell}{\partial x_n}(x)\end{pmatrix}\begin{pmatrix}\frac{\partial F_\ell}{\partial x_1}(x) & \dots & \frac{\partial F_\ell}{\partial x_n}(x)\end{pmatrix} \\ &= \sum_{\ell=1}^{m} \begin{pmatrix}\frac{\partial F_\ell}{\partial x_1}(x)\frac{\partial F_\ell}{\partial x_1}(x) & \dots & \frac{\partial F_\ell}{\partial x_1}(x)\frac{\partial F_\ell}{\partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial F_\ell}{\partial x_n}(x)\frac{\partial F_\ell}{\partial x_1}(x) & \dots & \frac{\partial F_\ell}{\partial x_n}(x)\frac{\partial F_\ell}{\partial x_n}(x)\end{pmatrix} \\ &= \begin{pmatrix} & \vdots & \\ \dots & \sum_{\ell=1}^{m} \frac{\partial F_\ell}{\partial x_i}(x)\frac{\partial F_\ell}{\partial x_j}(x) & \dots \\ & \vdots & \end{pmatrix}_{1\le i,j\le n}.\end{aligned}$$

Thus, the (i,j)-entry of this matrix is precisely $\sum_{\ell=1}^{m} \frac{\partial F_\ell}{\partial x_i}(x)\,\frac{\partial F_\ell}{\partial x_j}(x)$, which corresponds exactly to the first term in (2.10).

The Hessian of a single residual function F_ℓ is

$$H_{F_\ell}(x) \;=\; \Big(\, \tfrac{\partial^2 F_\ell}{\partial x_i\,\partial x_j}(x) \,\Big)_{1\le i,j\le n}.$$

Combining these results with (2.10), we obtain the compact form

$$H_f(x) \;=\; \sum_{\ell=1}^{m}\Big(\nabla F_\ell(x)\,\nabla F_\ell(x)^T \;+\; F_\ell(x)\,H_{F_\ell}(x)\Big). \tag{2.11}$$

Jacobian notation (for later reference). Let

$$F\colon \mathbb{R}^n \to \mathbb{R}^m, \qquad F(x) = \begin{pmatrix}F_1(x) \\ \vdots \\ F_m(x)\end{pmatrix}.$$

For the least-squares problem introduced in Definition 2.1.1, the mapping F is assumed to be twice continuously differentiable. Its derivative $F'(x)$ is the *Jacobian matrix* (or Jacobian)

$$F'(x) = J(x) = \begin{pmatrix}\frac{\partial F_1}{\partial x_1}(x) & \dots & \frac{\partial F_1}{\partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial F_m}{\partial x_1}(x) & \dots & \frac{\partial F_m}{\partial x_n}(x)\end{pmatrix} = \begin{pmatrix}\nabla F_1(x)^T \\ \vdots \\ \nabla F_m(x)^T\end{pmatrix} \in \mathbb{R}^{m\times n}. \tag{2.12}$$

With this notation, the results obtained above can be expressed compactly in matrix form. From (2.9), the gradient of f can be written as

$$\nabla f(x) = J(x)^T F(x), \tag{2.13}$$

and from (2.11), the Hessian follows as

$$H_f(x) = J(x)^T J(x) + \sum_{\ell=1}^{m} F_\ell(x)\,H_{F_\ell}(x). \tag{2.14}$$

The first term $J(x)^T J(x)$ corresponds to $\sum_{\ell=1}^{m} \nabla F_\ell(x)\,\nabla F_\ell(x)^T$ and is often referred to as the *Gauss–Newton term*. The second term collects all contributions involving the Hessians $H_{F_\ell}(x)$ of the residuals F_ℓ. It vanishes when the residuals are small, which will motivate the *Gauss–Newton approximation*.

§4 Newton–Raphson Method

The idea is to determine stationary points $\widehat{x}$, that is, points for which

$$\nabla f(\widehat{x}) = \mathbf{0}.$$

As already mentioned, one may then examine whether such points correspond to local minima.

Define

$$g : \mathbb{R}^n \to \mathbb{R}^n, \quad g(x) = \nabla f(x).$$

Finding stationary points is therefore equivalent to solving a nonlinear system of equations, a problem known from the course *Numerical Analysis*. Since f is twice continuously differentiable, g is continuously differentiable as well. Hence, the Newton method is a natural candidate for computing such points:

$$x_{k+1} = x_k - g'(x_k)^{-1} g(x_k), \quad \text{starting from some } x_0.$$

For the method to be applicable, $g'(x_k)$ must be invertible at every step. Convergence can only be guaranteed under additional assumptions. Under certain conditions, the convergence is even quadratic if the initial guess is sufficiently close to the solution.

For reference, the following theorem was proven in the lecture.

Theorem 2.4.1 (Quadratic convergence of the Newton method for systems)
Let $M \subseteq \mathbb{R}^n$ be open and convex, $\|\cdot\|$ a norm in $\mathbb{R}^n$, $|||\cdot|||$ a compatible matrix norm in $\mathbb{R}^{n\times n}$, and $g : M \to \mathbb{R}^n$ a continuously differentiable function whose Jacobian $g'(x)$ is invertible for all $x \in M$, such that

$$|||(g'(x))^{-1}||| \leq \beta \quad \forall x \in M. \tag{2.15}$$

Furthermore, assume that $g'(x)$ is Lipschitz continuous on M with constant γ, i.e.

$$|||g'(x) - g'(y)||| \leq \gamma \|x - y\| \quad \forall x, y \in M. \tag{2.16}$$

Suppose there exists a solution $\widehat{x} \in M$ of $g(x) = \mathbf{0}$. The initial guess x_0 satisfies

$$x_0 \in U_\omega := \{x \in \mathbb{R}^n : \|x - \widehat{x}\| < \omega\},$$

where ω is sufficiently small such that $U_\omega \subseteq M$, and there exists a constant α satisfying

$$\frac{\beta\gamma\omega}{2} \leq \alpha < 1. \tag{2.17}$$

Then the Newton method is well defined and converges quadratically to $\widehat{x}$.

The assumptions of the theorem are often difficult to verify in practice, and the behavior of the Newton method strongly depends on the specific problem at hand. To illustrate its application and to gain some intuition, we now consider the test problems introduced in Section §2.

Example 2.4.2 (Application to f_{ex1})
We first consider the simplest case,

$$f_{\text{ex1}}(r) := f(r, 0, 0),$$

that is, we assume the circle is centered at the origin and only the radius r is unknown. Hence, the center coordinates c_x and c_y are fixed, and all partial derivatives of F_i with respect to c_x and c_y vanish. The residuals simplify to

$$F_i(r) = \sqrt{x_i^2 + y_i^2} - r,$$

with

$$\frac{dF_i}{dr} = -1, \qquad \frac{d^2F_i}{dr^2} = 0.$$

Therefore, the gradient and Hessian of f_{ex1} are

$$\begin{aligned} g(r) = \nabla f_{\text{ex1}}(r) &= \sum_{i=1}^{m} F_i(r)\,\frac{dF_i}{dr} = -\sum_{i=1}^{m} F_i(r) = -\sum_{i=1}^{m}\left(\sqrt{x_i^2 + y_i^2} - r\right) \\ &= m\,r - \sum_{i=1}^{m}\sqrt{x_i^2 + y_i^2}, \end{aligned}$$

and

$$g'(r) = H_{f_{\text{ex1}}}(r) = \sum_{i=1}^{m}\left[\left(\frac{dF_i}{dr}\right)^2 + F_i(r)\,\frac{d^2F_i}{dr^2}\right] = \sum_{i=1}^{m} 1 = m,$$

where m denotes the number of data points. Consequently, the Newton step becomes particularly simple:

$$r_{k+1} = r_k - \frac{g(r_k)}{g'(r_k)} = r_k + \frac{1}{m}\sum_{i=1}^{m} F_i(r_k) = \frac{1}{m}\sum_{i=1}^{m}\sqrt{x_i^2 + y_i^2}.$$

Thus, in a single iteration the method already yields the exact solution, namely the average distance of the data points from the origin. □

Example 2.4.3 (Application to f_{ex2}) We now consider the case

$$f_{\text{ex2}}(r, c_x) := f(r, c_x, 0),$$

where the circle center lies on the x-axis. The y-coordinate c_y is fixed to zero, while r and c_x are unknown. The residuals are

$$F_i(r, c_x) = \sqrt{(x_i - c_x)^2 + y_i^2} - r.$$

The partial derivatives of F_i are

$$\frac{\partial F_i}{\partial r} = -1, \qquad \frac{\partial F_i}{\partial c_x} = \frac{c_x - x_i}{\sqrt{(x_i - c_x)^2 + y_i^2}},$$

and the nonzero second derivatives are

$$\frac{\partial^2 F_i}{\partial c_x^2} = \frac{y_i^2}{\left((x_i - c_x)^2 + y_i^2\right)^{3/2}}.$$

With $x = (r, c_x)^T$, the gradient and Hessian of f_{ex2} follow from

$$g(x) = \nabla f_{\text{ex2}}(x) = \sum_{i=1}^{m} F_i(x)\, \nabla F_i(x), \qquad g'(x) = \sum_{i=1}^{m} \Big(\nabla F_i(x)\, \nabla F_i(x)^T + F_i(x)\, H_{F_i}(x) \Big).$$

Explicitly,

$$g(x) = \sum_{i=1}^{m} \begin{pmatrix} -F_i(x) \\ F_i(x)\, \dfrac{c_x - x_i}{\sqrt{(x_i - c_x)^2 + y_i^2}} \end{pmatrix},$$

and

$$g'(x) = \sum_{i=1}^{m} \begin{pmatrix} 1 & -\dfrac{c_x - x_i}{\sqrt{(x_i - c_x)^2 + y_i^2}} \\ -\dfrac{c_x - x_i}{\sqrt{(x_i - c_x)^2 + y_i^2}} & \dfrac{(c_x - x_i)^2}{(x_i - c_x)^2 + y_i^2} + F_i(x)\, \dfrac{y_i^2}{\Big((x_i - c_x)^2 + y_i^2\Big)^{3/2}} \end{pmatrix}.$$

In this two-dimensional setting, the Newton update

$$x_{k+1} = x_k - g'(x_k)^{-1} g(x_k)$$

can be computed iteratively until convergence. For noise-free data and a starting value close to the solution, the method converges quadratically to the true parameters (r, c_x). □

Anhang A Supplementary Notes to Chapters

§1 Least Square

Bibliography

[1] Horn, Roger A. und Charles R. Johnson: *Matrix Analysis*, 2. Auflage, Cambridge University Press, 2001.