6 Basic Algorithms

The aim of this chapter is to present some fundamental ideas concerning algorithms for smooth optimization problems. We start by studying the Picard iterations (introduced and discussed in the second chapter) as a method to approximate the roots of certain nonlinear equations and this discussion naturally leads us to investigate some convergence acceleration techniques for the initial sequence of iterations. We then present Newton's method for solving nonlinear equations. In the convex framework, we present the proximal point algorithm. For optimization problems without restrictions we are interested in the line search method, which we discuss in detail, while for constrained problems, we give the sequential quadratic and the interior point methods. All the theoretical elements are then discussed and verified through some Matlab-based numerical simulations.

The situations we met in the majority of the concrete examples of optimization problems (see last chapter) are very hospitable, in the sense that we are able to find the solutions exactly, by solving the systems which give the solution candidates, and using the theoretical means we have previously developed. However, in many cases, the systems are not solvable, as in Section 3.4, when we considered the nonlinear case of the least square method, and the equation for finding the Lagrange multiplier for the problem of the computation of a projection on a generalized ellipsoid. For problems of this nature, it is necessary to develop methods called algorithms, in order to approximate the respective solutions. In general, there is a clear difference between the design of the algorithms for unconstrained optimization problems and those for constrained problems.

All the algorithms require a starting point, denoted x_0. Generally speaking, it is useful that this point is itself a good approximation of the solution we are looking for (especially if the solution is not unique). For instance, the function $f : \mathbb{R} \to \mathbb{R}$,

$$f(x) = \frac{x^4}{4} - \frac{x^3}{3} - x^2$$

has two minimal points: $x = -1$ is a local minimum and $x = 2$ is a global minimum. If we start with a value x_0 close to one of these points, then (roughly speaking) it is highly possible that the algorithm will converge to that point.

In general, after the choice of x_0, the algorithm generates a sequence of iterations $(x_k)_{k\in\mathbb{N}}$ with the aim of approaching the solution. The process of generating new iterations stops when no new progress can be made in the effort to come closer to the solution (according to the internal rule of the algorithm), or when an accuracy previously established was attained. Any algorithm should generate new iterations from the existing ones. In general, every new iteration should progress towards the solution. Some algorithms are called non-monotonic, and they do not necessarily progress at every step.

The study of efficient algorithms to detect (or to approximate) the solutions of optimization problems is a huge subject, several comprehensive monographs are fully dedicated to it. We just point out here the main ideas. Many of the very efficient algorithms are implemented into the functions of scientific software as Scilab or Matlab. We illustrate this at the end of the chapter.

There are at least two problems to be studied when an algorithm is designed: we are interested in knowing if the algorithm is global (i.e., it is convergent for any initial data), and to know its speed of convergence. Therefore, for a sequence $(x_k) \subset \mathbb{R}^p$ convergent to $\overline{x} \in \mathbb{R}^p$ with $x_k \neq \overline{x}$ for every $k \in \mathbb{N}^*$, one calls order of convergence the greatest natural number q for which

$$\lim_{k\to\infty} \frac{\|x_{k+1} - \overline{x}\|}{\|x_k - \overline{x}\|^q} \in [0, \infty).$$

If $q = 1$, then one has linear convergence, and if the above limit is 0, we have superlinear convergence. If $q = 2$, we have quadratic convergence. However, the aim is to design global algorithms which have a very good speed of convergence (at least quadratic).

We start with some methods to approximate the roots of some nonlinear equations. Some of the ideas from the next section make the link between the preceding results in the case of optimization algorithms. We have already seen a nonlinear equation that cannot be solved at the end of Chapter 3. Furthermore, let us remark that Theorem 3.2.6 transforms an optimization problem into the problem of solving the equation $L(x, (\lambda, \mu)) = 0$, which, in many cases, is highly nonlinear, and therefore not solvable.

6.1 Algorithms for Nonlinear Equations

6.1.1 Picard's Algorithm

To solve $g(x) = 0$ is equivalent to looking for the fixed points of the function $f(x) = g(x) + x$. The first part of our discussion, therefore, concerns the convergence of the Picard iteration, whose theoretical study was made in Section 2.3.2. Recall that if $f : [a, b] \to [a, b]$ is a differentiable function such that its derivative is bounded on $[a, b]$ by a positive constant strictly smaller than one, then for any initial data $x_0 \in [a, b]$ the Picard iteration defined by $x_{k+1} = f(x_k)$, $k \geq 0$ is convergent to the unique fixed point $\overline{x} \in [a, b]$ of f. Moreover, if the fixed point is not attained, then

$$\frac{x_{k+1} - \overline{x}}{x_k - \overline{x}} = \frac{f(x_k) - \overline{x}}{x_k - \overline{x}} \overset{k\to\infty}{\to} f'(\overline{x}).$$

Therefore, in general, we have a linear convergence: for k big enough, the error (the absolute value of the difference between the iteration and the real value of the fixed

point) at the step $(k+1)$ is proportional to the error at the step k. This kind of convergence is not very fast.

Let us consider the contraction $f : [0, \infty) \to [0, \infty)$ given by $f(x) = \frac{1}{1+x^2}$. According to Banach Principle, f has only one fixed point in $[0, \infty)$ and this is the unique real solution of the equation $x^3 + x - 1 = 0$, whose approximate value is $\overline{x} \approx 0.6823278$. Moreover, as before, for the Picard iterations, we have

$$\frac{x_{k+1} - \overline{x}}{x_k - \overline{x}} \overset{k\to\infty}{\to} f'(\overline{x}) = \frac{-2\overline{x}}{(1+\overline{x}^2)^2} = -2\overline{x}^3 \approx -0.63534438165.$$

See Section 6.3 for the numerical implementation which gives the value above. Therefore, for k big enough, at every iterative step, the error is multiplied by the approximate value 0.6353. In contrast, if we study the restriction of $\sin x$ to the interval $[0, 1]$, which is a weak contraction with $\overline{x} = 0$ as fixed point, we deduce that

$$\frac{x_{k+1} - \overline{x}}{x_k - \overline{x}} \overset{k\to\infty}{\to} f'(\overline{x}) = 1,$$

and we do not expect a good speed of convergence. In the last section of the chapter, we illustrate these theoretical predictions.

Remark 6.1.1. *The actual speed of convergence for the Picard iterations is given by the value of $f'(\overline{x})$. In the best case where $f'(\overline{x}) = 0$ we can have better convergence than the linear case. In general, in the above framework, if $f'(\overline{x}) = 0$ and f is twice differentiable, then the double application of the L'Hôpital rule gives*

$$\lim_{x\to\overline{x}} \frac{f(x) - \overline{x}}{(x-\overline{x})^2} = \frac{f''(\overline{x})}{2}, \tag{6.1.1}$$

so, for every nonstationary Picard iteration,

$$\lim_{k\to\infty} \frac{x_{k+1} - \overline{x}}{(x_k - \overline{x})^2} = \frac{f''(\overline{x})}{2},$$

whence a quadratic convergence.

Remark 6.1.2. *In fact, for the quadratic convergence described before it is enough for $\lim_{x\to\overline{x}} \frac{f(x)-\overline{x}}{(x-\overline{x})^2}$ to exist or, more generally, $\lim_{x\to\overline{x}} \frac{|f(x)-\overline{x}|}{(x-\overline{x})^2}$, and this can happen without the twice differentiability of f : for instance, for $f : \mathbb{R} \to \mathbb{R}$,*

$$f(x) = \begin{cases} x^2, & x \geq 0 \\ -x^2, & x < 0, \end{cases}$$

there exists the limit

$$\lim_{x\to\overline{x}} \frac{|f(x) - \overline{x}|}{(x-\overline{x})^2} = 1$$

at $\overline{x} = 0$, but the function is not twice differentiable at 0. This remark will be useful later.

If $f'(\overline{x}) \neq 0$, the speed of convergence of the Picard iterations is only linear. We present in the next section a method to overcome this difficulty. This technique is called the Aitken acceleration method, and was developed by the New Zealand-born mathematician Alexander Craig Aitken in 1926.

Consider $f : [a, b] \to [a, b]$ a differentiable function such that $\left|f'(x)\right| \in [0, 1)$ for every $x \in [a, b]$. For every $x_0 \in [a, b]$, the Picard iteration defined by $x_{k+1} = f(x_k)$, $k \geq 0$ is convergent to the unique fixed point of f in $[a, b]$, denoted $\overline{x}$. Suppose that $f'(\overline{x}) \neq 0$, so $\left|f'(\overline{x})\right| \in (0, 1)$. If (x_k) is nonstationary,

$$\frac{x_{k+1} - \overline{x}}{x_k - \overline{x}} = \frac{f(x_k) - \overline{x}}{x_k - \overline{x}} \overset{k\to\infty}{\to} f'(\overline{x}). \tag{6.1.2}$$

Therefore, we can write

$$f(x_k) - \overline{x} = \rho_k(x_k - \overline{x}),$$

where $\rho_k \overset{k\to\infty}{\to} f'(\overline{x})$, which means

$$\overline{x} = \frac{f(x_k) - \rho_k x_k}{1 - \rho_k},$$

that is

$$\overline{x} = x_k + \frac{f(x_k) - x_k}{1 - \rho_k}. \tag{6.1.3}$$

Aitken's initial idea was to find another sequence (μ_k) to approximate $f'(\overline{x})$. The result is given in the next result.

Proposition 6.1.3. *If (x_k) is a nonstationary Picard sequence, then the sequence defined by*

$$\mu_k = \frac{f(f(x_k)) - f(x_k)}{f(x_k) - x_k}, \ \forall k \in \mathbb{N}$$

has the limit $f'(\overline{x})$.

Proof Since $x_{k+1} = f(x_k)$, we have that $x_{k+2} = f(f(x_k))$ and from the definition of μ_k we deduce that

$$\begin{aligned}\mu_k &= \frac{x_{k+2} - \overline{x} - (x_{k+1} - \overline{x})}{(x_{k+1} - \overline{x}) - (x_k - \overline{x})} \\ &= \frac{\frac{x_{k+2}-\overline{x}}{x_{k+1}-\overline{x}} - 1}{1 - \frac{x_k-\overline{x}}{x_{k+1}-\overline{x}}},\end{aligned}$$

and from (6.1.2), we get

$$\lim \mu_k = \frac{f'(\overline{x}) - 1}{1 - \frac{1}{f'(\overline{x})}} = f'(\overline{x}).$$

The proposition is proved. □

Now, relation (6.1.3) and the above result suggest that we should consider the sequence:

$$\begin{aligned} y_k &= x_k + \frac{f(x_k) - x_k}{1 - \frac{f(f(x_k)) - f(x_k)}{f(x_k) - x_k}} \\ &= x_k - \frac{(f(x_k) - x_k)^2}{f(f(x_k)) - 2f(x_k) + x_k} \\ &= \frac{x_k f(f(x_k)) - f(x_k)^2}{f(f(x_k)) - 2f(x_k) + x_k}. \end{aligned}$$

We prove that the sequence (y_k) also converges to the fixed point $\overline{x}$, but faster than (x_k).

Theorem 6.1.4. *Let (x_k) be a Picard nonstationary sequence convergent to $\overline{x}$ such that*

$$\lim \frac{x_{k+1} - \overline{x}}{x_k - \overline{x}} = f'(\overline{x}) \in (-1, 1) \setminus \{0\}.$$

If the sequence (y_k) given by

$$y_k = x_k - \frac{(f(x_k) - x_k)^2}{f(f(x_k)) - 2f(x_k) + x_k}$$

is well defined, then it converges to $\overline{x}$ and, moreover,

$$\lim \frac{y_k - \overline{x}}{x_k - \overline{x}} = 0.$$

Proof We show first that (y_k) converges to $\overline{x}$. In the definition of (y_k), we add and subtract $\overline{x}$, and then we divide both numerator and denominator by $x_{k+1} - \overline{x}$. We obtain

$$y_k = x_k + \frac{\left(1 + \frac{\overline{x} - x_k}{x_{k+1} - \overline{x}}\right)(x_{k+1} - x_k)}{\left(1 + \frac{\overline{x} - x_k}{x_{k+1} - \overline{x}}\right) - \left(\frac{x_{k+2} - \overline{x}}{x_{k+1} - \overline{x}} - 1\right)}.$$

Passing to the limit (and denoting, for simplicity, $f'(\overline{x}) := \alpha$), we deduce:

$$\lim y_k = \overline{x} + \frac{1 - \alpha^{-1}}{1 - \alpha^{-1} - \alpha + 1} \lim(x_{k+1} - x_k) = \overline{x}.$$

We show now that (y_k) converges faster than (x_k). We have

$$\begin{aligned} \frac{y_k - \overline{x}}{x_k - \overline{x}} &= \frac{x_k - \overline{x} - \frac{(f(x_k) - x_k)^2}{f(f(x_k)) - 2f(x_k) + x_k}}{x_k - \overline{x}} \\ &= 1 + \frac{\frac{(f(x_k) - x_k)^2}{(x_{k+1} - x_k) - (x_{k+2} - x_{k+1})}}{x_k - \overline{x}}. \end{aligned}$$

By the same method as above, we get

$$\frac{y_k - \overline{x}}{x_k - \overline{x}} = 1 + \frac{\left(1 + \frac{\overline{x} - x_k}{x_{k+1} - \overline{x}}\right)\left(-1 + \frac{x_{k+1} - \overline{x}}{x_k - \overline{x}}\right)}{\left(1 + \frac{\overline{x} - x_k}{x_{k+1} - \overline{x}}\right) - \left(-1 + \frac{x_{k+2} - \overline{x}}{x_{k+1} - \overline{x}}\right)},$$

so,

$$\lim \frac{y_k - \overline{x}}{x_k - \overline{x}} = 1 + \frac{(1 - \alpha^{-1})(-1 + \alpha)}{2 - \alpha - \alpha^{-1}} = 0.$$

Consequently, (y_k) converges faster than (x_k). □

Despite the fact that (y_k) produces an acceleration of the speed of convergence, the order of convergence does not change: similar calculations show that

$$\lim \frac{y_{k+1} - \overline{x}}{y_k - \overline{x}} = f'(\overline{x}),$$

that is as well linear convergence. We call this the weak Aitken acceleration method.

However, starting from this idea, we consider a Picard iteration, but for the function

$$h(x) = \frac{xf(f(x)) - f(x)^2}{f(f(x)) - 2f(x) + x}$$

(this time, $f(x)^2$ means $f(x) \cdot f(x)$). The function h cannot be formally defined at $\overline{x}$, but one can extend its definition at that point by continuity, since

$$\begin{aligned}\lim_{x \to \overline{x}} h(x) &= \frac{f(f(\overline{x})) + \overline{x}f'(f(\overline{x}))f'(\overline{x}) - 2f(\overline{x})f'(\overline{x})}{f'(f(\overline{x}))f'(\overline{x}) - 2f'(\overline{x}) + 1} \\ &= \frac{\overline{x} + \overline{x}(f'(\overline{x}))^2 - 2\overline{x}f'(\overline{x})}{(f'(\overline{x}))^2 - 2f'(\overline{x}) + 1} = \overline{x},\end{aligned}$$

(recall that $\left|f'(\overline{x})\right| < 1$). So, with this extension ($h(\overline{x}) = \overline{x}$), the fixed point of f is a fixed point of h. Suppose that there exists a neighborhood $V := [\overline{x} - \mu, \overline{x} + \mu]$ of $\overline{x}$ ($\mu > 0$) with the property that for every $x \in (V \setminus \{\overline{x}\}) \cap [a, b]$, $f(f(x)) - 2f(x) + x \neq 0$. Then the converse holds as well: if u is a fixed point of h from $V \cap [a, b]$, then the equality $h(u) = u$ leads to $(f(u) - u)^2 = 0$. Therefore, the sole fixed point of h in $V \cap [a, b]$ is $\overline{x}$. It is also clear that h is derivable on $(V \setminus \{\overline{x}\}) \cap [a, b]$.

Moreover, we suppose that f is of class C^2 on $V \cap [a, b]$. We can show that h is derivable at $\overline{x}$ and its derivative at $\overline{x}$ is $h'(\overline{x}) = 0$. For simplicity, suppose that $\overline{x} \in (a, b)$ so that we can think that $V \subset (a, b)$. However, this is not essential. Write down Taylor's Formula for f around $\overline{x}$: for every ε with $|\varepsilon| < \mu$, there exists $\theta_\varepsilon \in (0, 1)$ such that

$$\begin{aligned}f(\overline{x} + \varepsilon) &= f(\overline{x}) + f'(\overline{x})\varepsilon + \frac{f''(\overline{x} + \theta_\varepsilon \varepsilon)}{2}\varepsilon^2 \\ &= \overline{x} + f'(\overline{x})\varepsilon + \frac{f''(\overline{x} + \theta_\varepsilon \varepsilon)}{2}\varepsilon^2.\end{aligned}$$

We fix ε. For ease of computation, we denote

$$\frac{f''(\overline{x} + \theta_\varepsilon \varepsilon)}{2} =: A_\varepsilon \text{ and } f'(\overline{x})\varepsilon + \frac{f''(\overline{x} + \theta_\varepsilon \varepsilon)}{2}\varepsilon^2 =: \delta_\varepsilon.$$

It is obvious that for ε small enough, $|\delta_\varepsilon| < \mu$, so

$$f(\overline{x} + \delta_\varepsilon) = \overline{x} + f'(\overline{x})\delta_\varepsilon + A_{\delta_\varepsilon}\delta_\varepsilon^2.$$

From the expression for h and a few computations, we obtain

$$\begin{aligned} h(\overline{x}+\varepsilon) &= \frac{(\overline{x}+\varepsilon)f(\overline{x}+\delta_\varepsilon)-(\overline{x}+\delta_\varepsilon)^2}{f(\overline{x}+\delta_\varepsilon)-2(\overline{x}+\delta_\varepsilon)+(\overline{x}+\varepsilon)} \\ &= \overline{x}-\frac{\delta_\varepsilon^2-f'(\overline{x})\varepsilon\delta_\varepsilon-A_{\delta_\varepsilon}\varepsilon\delta_\varepsilon^2}{\varepsilon-2\delta_\varepsilon+f'(\overline{x})\delta_\varepsilon+A_{\delta_\varepsilon}\delta_\varepsilon^2} \\ &= \overline{x}-\varepsilon^2(f'(\overline{x})+A_\varepsilon\varepsilon)\cdot \\ &\quad \frac{A_\varepsilon-A_{\delta_\varepsilon}f'(\overline{x})-A_{\delta_\varepsilon}A_\varepsilon\varepsilon}{(1-f'(\overline{x}))^2-\varepsilon(2A_\varepsilon-A_\varepsilon f'(\overline{x})-A_{\delta_\varepsilon}f'(\overline{x})^2+2A_\varepsilon A_{\delta_\varepsilon}f'(\overline{x})\varepsilon+A_{\delta_\varepsilon}A_\varepsilon^2\varepsilon^2)}. \end{aligned}$$

We write the quotient $\frac{h(\overline{x}+\varepsilon)-\overline{x}}{\varepsilon}$, and pass to the limit as $\varepsilon\to 0$. Then, since $f'(\overline{x})\neq 1$ and

$$A_\varepsilon \overset{\varepsilon\to 0}{\to} \frac{f''(\overline{x})}{2},\ A_{\delta_\varepsilon} \overset{\varepsilon\to 0}{\to} \frac{f''(\overline{x})}{2},$$

we deduce that there exists

$$\lim_{\varepsilon\to 0}\frac{h(\overline{x}+\varepsilon)-\overline{x}}{\varepsilon}=0.$$

The claim is proved. Furthermore,

$$\lim_{x\to\overline{x}}\frac{h(x)-\overline{x}}{(x-\overline{x})^2}=\lim_{\varepsilon\to 0}\frac{h(\overline{x}+\varepsilon)-\overline{x}}{\varepsilon^2}=-\frac{f''(\overline{x})}{2}\cdot\frac{f'(\overline{x})}{1-f'(\overline{x})}\in\mathbb{R}.$$

In particular, if one changes the neighborhood V, if needed, h is a contraction from V to V. By the use of this result and Remark 6.1.1, the sequence (x_k) defined by

$$x_{k+1}=x_k-\frac{(f(x_k)-x_k)^2}{f(f(x_k))-2f(x_k)+x_k}$$

with well chosen initial data is convergent towards $\overline{x}$ quadratically, i.e.,

$$\lim\frac{|x_{k+1}-\overline{x}|}{(x_k-\overline{x})^2}\in[0,\infty).$$

We call this method the strong Aitken acceleration method.

The drawback is that x_0 should be chosen from V, so it should be close enough to $\overline{x}$ (such that the equation $f(f(x))-2f(x)+x=0$ should not have another root in V, except $\overline{x}$). Another supplementary assumption was linked to the order of smoothness of f. In fact, this is the price which must be paid in order to have such a good speed of convergence. Let us remark that, at first sight, the former requirement looks pretty heavy: it is unnatural to ask for an initial data close to the point $\overline{x}$ we want to approximate. A possible solution to this would be to generate, for some steps, the Picard iterations in order to get close to $\overline{x}$ and then to apply the strong Aitken method.

6.1.2 Newton's Method

The celebrated Newton's method is one of the most well-known iterative procedures to approximate the roots of a function with sufficient differentiability properties. We shall see that this is a local algorithm (since in order to have the desired convergence one should take as initial data a value sufficiently close to the solution), but it converges quadratically.

Let us consider a function $f : \mathbb{R}^p \to \mathbb{R}^p$ of class C^1 and let $\overline{x}$ be a nondegenerate root of f (i.e., $f(\overline{x}) = 0$, and $\nabla f(\overline{x})$ is nonsingular). We consider a value x_0 close enough to $\overline{x}$.

The sequence of Newton iterations starts from the equation

$$0 = f(x_k) + \nabla f(x_k)(x_{k+1} - x_k). \tag{6.1.4}$$

This equation, which gives the value of x_{k+1}, shows why it is necessary to have a simple solution and we should start from a point close to $\overline{x}$: we should put our initial point in a neighborhood of $\overline{x}$ where ∇f is invertible, and such neighborhood do exist exactly because ∇f is continuous and nonsingular at $\overline{x}$. In this way, we formally define the Newton iteration by:

$$x_{k+1}^t = x_k^t - \nabla f(x_k)^{-1} \cdot (f(x_k))^t. \tag{6.1.5}$$

Recall that $\nabla f(x_k)$ can be identified with the Jacobian matrix. As seen from the previous relations, as well as from the convergence result given in the sequel, the main drawbacks of the Newton's method can be summarized as follows:

- when the starting point x_0 is not close enough to the solution $\overline{x}$, the algorithm associated to (6.1.5) does not converge;
- if $\nabla f(x_k)$ is singular, one cannot define x_{k+1};
- it may be too expensive to compute exactly $\nabla f(x_k)^{-1}$ for large p;
- it may happen that $\nabla f(\overline{x})$ is singular.

Theorem 6.1.5. *Suppose f is Lipschitz continuously differentiable on an open convex set $D \subset \mathbb{R}^p$. Let $\overline{x}$ be a nondegenerate root of the equation $f(x) = 0$, and let (x_k) be a sequence of iterates generated by (6.1.5). Then when $x_0 \in D$ is sufficiently close to $\overline{x}$, one has*

$$\lim_{k\to\infty} \frac{\|x_{k+1} - \overline{x}\|}{\|x_k - \overline{x}\|^2} \in [0, \infty), \tag{6.1.6}$$

i.e., we have local quadratic convergence.

Proof Since $f(\overline{x}) = 0$, we have from Theorem 1.4.13 that

$$f(x_k) = f(x_k) - f(\overline{x}) = \nabla f(x_k)(x_k - \overline{x}) + w(x_k, \overline{x}), \tag{6.1.7}$$

where

$$w(x_k, \overline{x}) = \int_0^1 \left[\nabla f(\overline{x} + t(x_k - \overline{x})) - \nabla f(x_k)\right](x_k - \overline{x})\, dt.$$

We have then

$$\left\|w(x_k,\overline{x})\right\| = \left\|\int_0^1 \left[\nabla f(\overline{x}+t(x_k-\overline{x}))-\nabla f(x_k)\right](x_k-\overline{x})\,dt\right\| \tag{6.1.8}$$
$$\leq \int_0^1 \left\|\nabla f(\overline{x}+t(x_k-\overline{x}))-\nabla f(x_k)\right\| \|x_k-\overline{x}\|\,dt,$$

hence by Lagrange Theorem there is $c_k \in [0,1]$ such that

$$\left\|w(x_k,\overline{x})\right\| \leq \left\|\nabla f(\overline{x}+c_k(x_k-\overline{x}))-\nabla f(x_k)\right\| \|x_k-\overline{x}\|.$$

This gives, due to the Lipschitz continuity of ∇f, that

$$\left\|w(x_k,\overline{x})\right\| \leq L\|x_k-\overline{x}\|^2, \tag{6.1.9}$$

where by L we have denoted the Lipschitz constant of ∇f.

Moreover, since $\nabla f(\overline{x})$ is nonsingular, there is a $\delta > 0$ sufficiently small and an $M > 0$ such that $\nabla f(x)$ is nonsingular on $D(\overline{x},\delta)$ and

$$\left\|\nabla f(x)^{-1}\right\| \leq M, \quad \forall x \in D(\overline{x},\delta).$$

One has, from (6.1.7) and (6.1.5), that

$$\begin{aligned} x_{k+1} &= x_k - \nabla f(x_k)^{-1}(f(x_k)) \\ &= x_k - \nabla f(x_k)^{-1}\left(\nabla f(x_k)(x_k-\overline{x}) + w(x_k,\overline{x})\right) \\ &= \overline{x} + \nabla f(x_k)^{-1}(w(x_k,\overline{x})), \end{aligned}$$

hence

$$\begin{aligned} \|x_{k+1}-\overline{x}\| &\leq \left\|\nabla f(x_k)^{-1}\right\| \cdot \left\|w(x_k,\overline{x})\right\| \\ &\leq \left\|\nabla f(x_k)^{-1}\right\| \cdot L\|x_k-\overline{x}\|^2. \end{aligned} \tag{6.1.10}$$

Take x_0 such that $x_0 \in D(\overline{x},\delta)$ and $ML\|x_0-\overline{x}\| := \rho < 1$. It follows inductively from (6.1.10) that $\|x_k-\overline{x}\| \leq \rho^k\|x_0-\overline{x}\|$ for every $k \geq 1$, hence $(x_k) \subset D(\overline{x},\delta)$ and $x_k \to \overline{x}$. Moreover, since

$$\|x_{k+1}-\overline{x}\| \leq ML\|x_k-\overline{x}\|^2, \ \forall k,$$

we get (6.1.6). □

In case $p = 1$, i.e., $f : \mathbb{R} \to \mathbb{R}$, the equation (6.1.5) becomes

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

In this case, the iterate x_{k+1} is exactly the one where the tangent to the graph of f at $(x_k, f(x_k))$ intersects Ox.

As in the case of Aitken method of acceleration, we discuss in this context ($p = 1$) some possibilities to choose the point x_0 sufficiently close to the solution such that the Newton iteration converges to it. An empirical possibility would be to study the graph of the function and to choose an x_0 value which seems to be close to the solution. Another possibility is to apply the method of halving interval. Let us suppose that we have a continuous function f and two real numbers $a < b$ for which $f(a)f(b) < 0$. Then f has a root in (a, b). We then generate two sequences (a_k) and (b_k) as follows: $a_0 = a$, $b_0 = b$. Let $x_0 = 2^{-1}(a_0 + b_0)$. If $f(x_0) = 0$ then x_0 is the solution we are looking for and the process stops. If $f(a_0)f(x_0) < 0$ then we choose $a_1 = a_0$ and $b_1 = x_0$, and if $f(x_0)f(b_0) < 0$ we choose $a_1 = x_0$ and $b_1 = b_0$. In the same way, we take $x_1 = 2^{-1}(a_1 + b_1)$. Going further, we get close to the solution with (x_k) by halving at every step the interval which contains the solution. In general, this convergence is not very rapid but is good enough to be used for some of the iterations in order to find initial data for the Newton method.

In the general case ($p \geq 1$), we remark that some methods, known as quasi-Newton methods, do not require the calculation of the Jacobian $\nabla f(x)$. Instead, they use an approximation of this matrix, updating it at each iteration in such a way that it mimics the behavior of the Jacobian over the current step. If we denote this approximation matrix by J_k, then equation (6.1.4) becomes

$$0 = (f(x_k))^t + J_k \cdot (x_{k+1} - x_k)^t,$$

which gives, when J_k is nonsingular, the explicit formula

$$x_{k+1}^t = x_k^t - J_k^{-1} \cdot (f(x_k))^t. \tag{6.1.11}$$

If we denote

$$s_k := x_{k+1} - x_k \quad \text{and} \quad y_k := f(x_{k+1}) - f(x_k),$$

then by using Theorem 1.4.13 we get that

$$y_k = \int_0^1 \nabla f(x_k + ts_k)s_k \, dt \approx \nabla f(x_{k+1})(s_k) + r(\|s_k\|),$$

where

$$\lim_{k\to\infty} \frac{r(\|s_k\|)}{\|s_k\|} = 0.$$

So, in order that J_k mimics the behavior of the Jacobian $\nabla f(x_k)$, one asks that J_{k+1} satisfies the so called secant equation:

$$y_k^t = J_{k+1}s_k^t, \tag{6.1.12}$$

which ensures that J_{k+1} and $\nabla f(x_{k+1})$ have similar behavior along s_k. In fact, (6.1.12) can be seen as a system of p equations with p^2 unknowns, where the unknowns are the elements of J_{k+1}, so for $p > 1$ the components of J_{k+1} are not uniquely determined.

One of the best ways to find J_{k+1} is described by Broyden's method, where the matrix is given by the recurrence

$$J_{k+1} = J_k + \frac{(y_k^t - J_k s_k^t) \cdot s_k}{\langle s_k, s_k \rangle}. \tag{6.1.13}$$

As shown by the next result, the Broyden update makes the smallest change to J_k (measured by the Euclidean norm) that is consistent to (6.1.12).

Proposition 6.1.6. *The matrix J_{k+1} given by (6.1.13) satisfies:*

$$\|J_{k+1} - J_k\| = \min \{\|J - J_k\| \mid y_k = Js_k\}.$$

Proof Take J any matrix which satisfies $y_k^t = Js_k^t$. We have then

$$\begin{aligned} \|J_{k+1} - J_k\| &= \left\| \frac{(y_k^t - J_k s_k^t) \cdot s_k}{\langle s_k, s_k \rangle} \right\| = \left\| \frac{((J - J_k) s_k^t) \cdot s_k}{\langle s_k, s_k \rangle} \right\| \\ &\leq \|J - J_k\| \left\| \frac{s_k^t \cdot s_k}{s_k^t \cdot s_k} \right\| = \|J - J_k\|, \end{aligned}$$

which finishes the proof. □

6.2 Algorithms for Optimization Problems

6.2.1 The Case of Unconstrained Problems

There exist several general methods to design algorithms for unconstrained optimization, but we concentrate on the line search method. The aim of this algorithm is to realize at every step a decrease of the value of the objective function $f : \mathbb{R}^p \to \mathbb{R}$, which is considered to be of class C^2. One asks that $f(x_{k+1}) < f(x_k)$. The algorithm computes for every term k a direction p_k (a vector of norm 1) and a step $\alpha_k > 0$ to move on the direction p_k. Therefore, starting from x_k, the new iteration will be

$$x_{k+1} = x_k + \alpha_k p_k. \tag{6.2.1}$$

With this approach, the choice of both the direction and the step are very important. From Taylor's Formula, for fixed α, p, there exists $t \in (0, \alpha)$

$$f(x_k + \alpha p) = f(x_k) + \alpha \nabla f(x_k)(p) + \frac{1}{2}\alpha^2 \nabla^2 f(x_k + tp)(p, p). \tag{6.2.2}$$

Putting aside the second order term (which for small α is small), the direction on which f decreases most is in fact the solution of the minimization on the unit ball of the

function (of p), $p \mapsto \nabla f(x_k)(p)$. Since

$$\nabla f(x_k)(p) = \|p\| \left\|\nabla f(x_k)\right\| \cos\theta = \left\|\nabla f(x_k)\right\| \cos\theta,$$

where θ is the angle between p and $\nabla f(x_k)$, it is clear that the minimum is attained for

$$p = -\frac{\nabla f(x_k)}{\left\|\nabla f(x_k)\right\|}$$

if $\left\|\nabla f(x_k)\right\| \neq 0$. So, if a critical point is attained, then we cannot go further. If is not the case, the choice of p_k as above is called the steepest descent method. On the other hand, every other direction for which the angle with $\nabla f(x_k)$ is greater that $\frac{\pi}{2}$ (i.e., $\cos\theta < 0$) produces a decrease of f if α is sufficiently small, since the second order term in (6.2.2) contains a factor of α^2. Such a direction (for which $\nabla f(x_k)(p_k) < 0$) is called a decrease direction. Now, one has the problem of the choice of α_k. The ideal choice would be the minimum for $\alpha > 0$, of the function $\alpha \mapsto f(x_k + \alpha p_k)$, but, again, this problem is not necessarily a simple one. Another possibility is to choose a number $\alpha > 0$ such that $f(x_k + \alpha p_k) < f(x_k)$, but this choice could be insufficient, since the function may have a not very important decrease. In order to avoid both the problem of exact solvability of the optimization problem and the latter mentioned difficulty, a compromise is to choose an α_k which satisfies

$$f(x_k + \alpha p_k) < f(x_k) + c_1 \alpha \nabla f(x_k)(p_k), \tag{6.2.3}$$

where $c_1 \in (0, 1)$. The above inequality is called the Armijo condition (and was introduced by the American mathematician Larry Armijo in 1966), and the possibility of choosing α to fulfill (6.2.3) is ensured by (6.2.2) and by $\nabla f(x_k)(p_k) < 0$. In general, in order that some values of α are sufficiently large to satisfy the condition (6.2.3), c_1 is taken to be small. Even so, there is a risk of choosing a value α that is too small, such that, usually, one needs a second condition of the type

$$c_2 \nabla f(x_k)(p_k) \leq \nabla f(x_k + \alpha p_k)(p_k), \tag{6.2.4}$$

where $c_2 \in (c_1, 1)$. This condition is called the curvature condition and says that the derivative of the mapping $\alpha \mapsto f(x_k + \alpha p_k)$ at α is bigger than the product between c_2 and the derivative of the same function at 0. It is clear that a lower value of $\nabla f(x_k)(p_k)$ implies a bigger decrease of f, so in condition (6.2.4) it is preferable that c_2 to be taken close to 1. The conditions (6.2.3) and (6.2.4) are called the Wolfe conditions (after the American mathematician Philip Wolfe who introduced them in 1968). If instead of (6.2.4) one takes

$$\left|\nabla f(x_k + \alpha p_k)(p_k)\right| \leq \left|c_2 \nabla f(x_k)(p_k)\right|,$$

then one talks about strong Wolfe conditions.

The consistency of these conditions is rigorously shown in the next sections.

Proposition 6.2.1. *Let $f : \mathbb{R}^p \to \mathbb{R}$ be of class C^1, p_k a decrease direction at x_k, and $0 < c_1 < c_2 < 1$. If f is lower bounded on the set $\{x_k + \lambda p_k \mid \lambda > 0\}$, then there exists $\alpha > 0$ which satisfies the Wolfe conditions and the strong Wolfe conditions.*

Proof According to the assumption, the function $\alpha \mapsto f(x_k + \alpha p_k)$ is lower bounded on $(0, \infty)$. Since $c_1 > 0$ and $\nabla f(x_k)(p_k) < 0$, for α small enough,

$$f(x_k + \alpha p_k) < f(x_k) + c_1 \alpha \nabla f(x_k)(p_k)$$

whence, taking into account the boundedness property, the equation (in α)

$$f(x_k + \alpha p_k) = f(x_k) + c_1 \alpha \nabla f(x_k)(p_k)$$

has at least a strictly positive solution. From the continuity (in α) of the functions involved, there exists a smallest strictly positive solution which we denote by α'. Obviously, for every $\alpha \in (0, \alpha')$, condition (6.2.3) holds. We apply again Taylor's Formula, and there exists $\alpha'' \in (0, \alpha')$ such that

$$f(x_k + \alpha' p_k) = f(x_k) + \alpha' \nabla f(x_k + \alpha'' p_k)(p_k),$$

hence

$$\nabla f(x_k + \alpha'' p_k)(p_k) = c_1 \nabla f(x_k)(p_k) > c_2 \nabla f(x_k)(p_k).$$

Therefore, for α'' condition (6.2.4) holds. Since for α'' the inequalities in both (6.2.3) and (6.2.4) are strict, there exists an interval around this point where these conditions are fulfilled. By the fact that $\nabla f(x_k + \alpha'' p_k)(p_k) < 0$, we infer that the strong Wolfe conditions hold in a whole interval around α''. □

We discuss now the convergence of the algorithm of the line search method.

Theorem 6.2.2. *Let us consider the iteration (6.2.1), where (p_k) are decrease directions, and (α_k) satisfy the Wolfe conditions. Suppose that f is of class C^1 and lower bounded, and that ∇f is Lipschitz. Then the series*

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \left\| \nabla f(x_k) \right\|^2$$

(where θ_k denotes the angle between $\nabla f(x_k)$ and p_k) is convergent.

Proof From the Wolfe conditions, for each $k \in \mathbb{N}^*$,

$$\nabla f(x_k + \alpha_k p_k)(p_k) - \nabla f(x_k)(p_k) \geq (c_2 - 1) \nabla f(x_k)(p_k),$$

that is

$$\nabla f(x_{k+1})(p_k) - \nabla f(x_k)(p_k) \geq (c_2 - 1) \nabla f(x_k)(p_k),$$

and the Lipschitz condition on the differential gives a positive constant L such that

$$\left\|\nabla f(x_{k+1}) - \nabla f(x_k)\right\| \leq L \left\|x_{k+1} - x_k\right\|,$$

from where,

$$\left(\nabla f(x_{k+1}) - \nabla f(x_k)\right)(p_k) \leq \alpha_k L \left\|p_k\right\|^2.$$

We infer that

$$\alpha_k \geq \frac{c_2 - 1}{L} \frac{\nabla f(x_k)(p_k)}{\left\|p_k\right\|^2}.$$

From (6.2.3), taking again into account the inequality $\nabla f(x_k)(p_k) < 0$, we get

$$f(x_{k+1}) \leq f(x_k) + c_1 \frac{c_2 - 1}{L} \frac{\left(\nabla f(x_k)(p_k)\right)^2}{\left\|p_k\right\|^2}.$$

But

$$\frac{\left(\nabla f(x_k)(p_k)\right)^2}{\left\|p_k\right\|^2} = \cos^2\theta_k \left\|\nabla f(x_k)\right\|^2,$$

so

$$f(x_{k+1}) - f(x_k) \leq c_1 \frac{c_2 - 1}{L} \cos^2\theta_k \left\|\nabla f(x_k)\right\|^2.$$

Summing up, we deduce

$$f(x_{k+1}) \leq f(x_0) + c_1 \frac{c_2 - 1}{L} \sum_{i=0}^{k} \cos^2\theta_i \left\|\nabla f(x_i)\right\|^2.$$

Since $c_2 - 1 < 0$, the lower boundedness of f yields the convergence of the series. □

The above theorem ensures that

$$\cos^2\theta_k \left\|\nabla f(x_k)\right\|^2 \to 0.$$

If the choice of p_k is made in such a way that $\cos\theta_k > \varepsilon$ for every k and for a fixed $\varepsilon > 0$, then $\nabla f(x_k) \to 0$. Such a situation is called the steepest descent method where $\cos\theta_k = -1$. The algorithm does not guarantee the convergence toward a minimum point. The fact that one gets a sequence of points where the norm of the gradient is smaller and smaller, however, gives us hope that we progress towards a critical point. Clearly, it can be a saddle point, hence not a minimum. Moreover, the speed of convergence is slow.

However, in particular situations, some versions of the line search algorithm can be analyzed more accurately, as the next example shows.

Example 6.2.3. *Let us consider the case of the function $f : \mathbb{R}^p \to \mathbb{R}$ given by $f(x) = \frac{1}{2}\left\langle (Ax^t)^t, x\right\rangle + \langle b, x\rangle$ where A is a symmetric, positive definite square matrix of dimension p, and $b \in \mathbb{R}^p$. Clearly, f is strictly convex and its level sets are bounded, so there exists*

a unique minimum point given by the equation $\nabla f(x) = 0$. *Therefore,* $\overline{x} = -(A^{-1}b^t)^t$. *Let us study the behaviour of the algorithm given by the relations* $x_{k+1} = x_k + \alpha_k d_k$, *where* $d_k = -\nabla f(x_k) = -(Ax_k^t)^t - b$, *and* α_k *is the minimum of the function* $\alpha \mapsto f(x_k + \alpha d_k)$. *Suppose that the gradient does not vanish at this iteration points, which is equivalent (taking into account the convexity of the problem) to* $f(x_k) > \overline{f}$, *where* $\overline{f}$ *is the minimum value of the function, that is* $\overline{f} := f(\overline{x}) = -\frac{1}{2}\left\langle (A^{-1}b^t)^t, b\right\rangle$.

Thus,

$$f(x_k + \alpha d_k) = f(x_k) + \frac{1}{2}\alpha^2 \left\langle (Ad_k^t)^t, d_k \right\rangle + \alpha \left\langle (Ad_k^t)^t + b, d_k \right\rangle.$$

Since $d_k \neq 0$, *the minimum of this function is attained at*

$$\alpha_k = \frac{\|d_k\|^2}{\left\langle (Ad_k^t)^t, d_k \right\rangle},$$

and

$$d_{k+1} = -(Ax_{k+1}^t)^t - b = -(Ax_k^t)^t - \alpha_k (Ad_k^t)^t - b = d_k - \alpha_k (Ad_k^t)^t,$$

whence

$$\langle d_{k+1}, d_k\rangle = \langle d_k, d_k\rangle - \alpha_k \left\langle (Ad_k^t)^t, d_k \right\rangle = 0.$$

We infer that

$$f(x_{k+1}) = f(x_k) - \frac{1}{2}\frac{\|d_k\|^4}{\left\langle (Ad_k^t)^t, d_k \right\rangle},$$

so

$$f(x_{k+1}) - \overline{f} = \left(f(x_k) - \overline{f}\right)\left[1 - \frac{\|d_k\|^4}{2\left(f(x_k) - \overline{f}\right)\left\langle (Ad_k^t)^t, d_k \right\rangle}\right].$$

Therefore,

$$\begin{aligned}\left\langle (A^{-1}d_k^t)^t, d_k \right\rangle &= \left\langle (A^{-1}((Ax_k^t)^t + b)^t)^t, (Ax_k^t)^t + b \right\rangle \\ &= 2\left[\frac{1}{2}\left\langle (Ax_k^t)^t, x_k \right\rangle + \langle b, x_k\rangle + \left\langle (A^{-1}b^t)^t, b \right\rangle\right] \\ &= 2\left(f(x_k) - \overline{f}\right).\end{aligned}$$

So,

$$f(x_{k+1}) - \overline{f} = \left(f(x_k) - \overline{f}\right)\left[1 - \frac{\|d_k\|^4}{\left\langle (A^{-1}d_k^t)^t, d_k \right\rangle \cdot \left\langle (Ad_k^t)^t, d_k \right\rangle}\right].$$

From the inequality of Kantorovici (Theorem 2.2.31), we have

$$\frac{\|d_k\|^4}{\left\langle (A^{-1}d_k^t)^t, d_k \right\rangle \cdot \left\langle (Ad_k^t)^t, d_k \right\rangle} \geq 4\left[\sqrt{\frac{\lambda_1}{\lambda_p}} + \sqrt{\frac{\lambda_p}{\lambda_1}}\right]^{-2} = 4\frac{\lambda_1\lambda_p^{-1}}{(\lambda_1\lambda_p^{-1} + 1)^2},$$

where λ_1 and λ_p are the greatest and the smallest eigenvalue of A, respectively. We denote $c := \lambda_1\lambda_p^{-1}$. We put together the above relations and get

$$f(x_{k+1}) - \overline{f} \leq \left(f(x_k) - \overline{f}\right)\left[1 - 4\frac{c}{(c+1)^2}\right].$$

It follows that

$$f(x_k) - \overline{f} \leq \left(f(x_0) - \overline{f}\right)\left(\frac{c-1}{c+1}\right)^{2k}, \ \forall k \in \mathbb{N}.$$

From Example 7.72,

$$\begin{aligned} f(x_k) - \overline{f} &= \frac{1}{2}\left\langle (Ax_k^t)^t, x_k\right\rangle + \langle b, x_k\rangle - \overline{f} \\ &= \frac{1}{2}\left\langle (A(x_k - \overline{x})^t)^t, x_k - \overline{x}\right\rangle \geq \frac{1}{2}\lambda_p \|x_k - \overline{x}\|^2, \end{aligned}$$

so

$$\|x_k - \overline{x}\| \leq \sqrt{\left[\frac{2(f(x_0) - \overline{f})}{\lambda_p}\right]}\left(\frac{c-1}{c+1}\right)^k, \ \forall k \in \mathbb{N},$$

and this relation allows us to conclude that the approximation of the minimum point depends on the value of c : if the difference between the greater and the smaller eigenvalues of A is small, then the convergence is rapid. At the limit, for $c = 1$, the first iteration already attains the minimum point.

On the other hand, there are several possible improvements of the general line search method and one of these possibilities would be to consider a second-order decrease direction (of Newton type). Therefore, let us suppose that we have a function $f : \mathbb{R} \to \mathbb{R}$ of class C^3. As above, the algorithm will search for critical points, i.e., to the solutions of the equation $f'(x) = 0$. If one supposes that $\overline{x}$ is a nondegenerate solution, and applies the Newton's method for this equation, they would be lead to consider the iterations

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}.$$

For this algorithm we have a quadratic convergence (as shown in the previous section), the drawback being the same as we discussed for the Newton's method. On the other hand, if $f''(x_k)$ is not positive, then the direction $-\frac{f'(x_k)}{f''(x_k)}$ is not necessarily a decreasing one, so we can attain maximal points.

We close this section with a special look to the case of convex functions. Clearly, the above algorithm is well suited to these functions, but the particular form of convexity allows the design of some powerful specific algorithms. We now introduce the proximal point algorithm. An initial form of this was published by the French mathematician Bernard Martinet in 1970, and later generalized by the American mathematician Ralph Tyrrell Rockafellar in 1976.

Let $f : \mathbb{R}^p \to \mathbb{R}$ be a convex differentiable function. Thus, for every fixed $y \in \mathbb{R}^p$, we consider the function $g_y : \mathbb{R}^p \to \mathbb{R}$,

$$g_y(x) = f(x) + \frac{1}{2} \|x - y\|^2 .$$

This new application is convex and differentiable (as a sum of functions with these properties). Moreover, g_y is strictly convex because $x \mapsto \frac{1}{2} \|x - y\|^2$ has this property (see Theorem 2.2.15 (*iv*)).

Suppose that f satisfies the coercivity condition of Proposition 3.1.8, whence f attains its global minimum on $\mathbb{R}^p$. It is easy to see that the same condition is satisfied by g_y as well, therefore there exists a global minimum point of g_y on $\mathbb{R}^p$. By the fact that g_y is strictly convex, this minimum point is unique (Proposition 3.1.24), and we denote it by $\overline{x}_y$. Furthermore, according to Theorem 3.1.22 and the differentiation rules, $\overline{x}_y$ is characterized by the relation

$$\nabla f(\overline{x}_y) + \overline{x}_y - y = 0.$$

We generate now a sequence of iterations following the next rule: $x_0 \in \mathbb{R}^p$, and for every $k \geq 0$, $x_{k+1} = \overline{x}_{x_k}$, that is,

$$\nabla f(x_{k+1}) + x_{k+1} - x_k = 0.$$

Let $\overline{x}$ be a minimum point for f (it exists according to the above assumptions). The next relation holds:

$$\|x_{k+1} - \overline{x}\|^2 = \|x_k - \overline{x}\|^2 - \|x_{k+1} - x_k\|^2 + 2 \langle x_{k+1} - \overline{x}, x_{k+1} - x_k \rangle , \ \forall k.$$

But

$$\langle x_{k+1} - \overline{x}, x_{k+1} - x_k \rangle = -\nabla f(x_{k+1})(x_{k+1} - \overline{x}) \leq 0$$

(from Theorem 2.2.10), so

$$\|x_{k+1} - \overline{x}\|^2 \leq \|x_k - \overline{x}\|^2 - \|x_{k+1} - x_k\|^2 \leq \|x_k - \overline{x}\|^2 , \ \forall k.$$

We deduce the following facts: the sequence $(\|x_k - \overline{x}\|)_k$ is decreasing, whence convergent (being positive), while the sequence $(\|x_{k+1} - x_k\|)_k$ is convergent to 0. In particular, the sequence $(x_k)_k$ is bounded. We show that (x_k) is convergent to a minimum point of f. Let $x \in \mathbb{R}^p$ arbitrary but fixed. Then

$$\nabla f(x_{k+1})(x - x_{k+1}) = \langle x_k - x_{k+1}, x - x_{k+1} \rangle \geq - \|x_k - x_{k+1}\| \, \|x - x_{k+1}\| .$$

Let $z \in \mathbb{R}^p$ be a limit point of (x_k) (its existence is ensured by the boundedness of the sequence). Passing to the limit in the above relation, using that $\|x_{k+1} - x_k\| \to 0$ and that $(\|x - x_{k+1}\|)$ is bounded, we deduce

$$\nabla f(z)(x - z) \geq 0.$$

Since x is arbitrary, we get $\nabla f(z) = 0$, that is z is a critical point, whence a minimum point. Suppose that (x_k) has at least two different limit points z_1 and z_2. According to the above stage of the proof, both are minimum points of f. With the same reasoning as in the case of $\overline{x}$, we infer that the sequences $(\|x_k - z_1\|)_k$ and $(\|x_k - z_2\|)_k$ are convergent. But

$$\|x_k - z_2\|^2 = \|x_k - z_1\|^2 + 2\langle x_k - z_1, z_1 - z_2\rangle + \|z_1 - z_2\|^2, \ \forall k.$$

Therefore, there exists

$$2\lim_k \langle x_k - z_1, z_1 - z_2\rangle = \lim_k \|x_k - z_2\|^2 - \lim_k \|x_k - z_1\|^2 - \|z_1 - z_2\|^2.$$

By the fact that z_1 is a limit point of (x_k), $\lim_k \langle x_k - z_1, z_1 - z_2\rangle$ can be only 0, so

$$\lim_k \|x_k - z_2\|^2 - \lim_k \|x_k - z_1\|^2 = \|z_1 - z_2\|^2 > 0.$$

Changing the roles of z_1 and z_2,

$$\lim_k \|x_k - z_1\|^2 - \lim_k \|x_k - z_2\|^2 = \|z_1 - z_2\|^2 > 0,$$

so we arrive at a contradiction. Thus (x_k) is convergent to a minimum point of f.

6.2.2 The Case of Constraint Problems

For constrained problems, we adopt a slightly simplified framework which allows, nevertheless, the presentation of the main ideas of two important methods of searching for extrema, namely the sequential quadratic programing and the interior point methods.

6.2.2.1 Sequential quadratic programming

The sequential quadratic programming (SQP) is one of the most effective methods used to solve nonlinear optimization problems with constraints. We will restrict our approach to equality-constrained optimization, i.e., we consider for the problem (P) defined in the second section of Chapter 3 only equalities constraints. We take then the C^1 function $h : \mathbb{R}^p \to \mathbb{R}^m$ and

$$M := \left\{x \in \mathbb{R}^p \mid h(x) = 0\right\}.$$

The underlying idea of SQP is to model the problem (P) at the current iterate x_k by a quadratic programming subproblem, and then to construct the next iteration x_{k+1} by the use of the minimizer of this subproblem.

In order to continue our discussion in the general case of nonlinear optimization problems with constraints, we present some elements about quadratic programming.

An optimization problem where the objective function is quadratic and the constraints are linear is called a quadratic program (QP). As above, we limit our analyses to the case of equality constraints, i.e., we consider the problem

$$\begin{array}{ll} (QP)\ \min_x & f(x) := \frac{1}{2}\left\langle (Qx^t)^t, x\right\rangle + \langle c, x\rangle\,, \\ \text{subject to} & Ax^t = b^t, \end{array}$$

where Q is a symmetric $p \times p$ matrix, A is a $m \times p$ matrix with $m \leq p$, x, c are vectors in $\mathbb{R}^p$, and b is a vector in $\mathbb{R}^m$. If the Hessian matrix Q is positive definite, then we speak about convex QP, and in this case the analysis is similar to the case of linear programs. When Q is an indefinite matrix, more difficulties can arise, since in this case several stationary and local minima may appear.

In what follows, we will restrict to the case of convex QPs, and we suppose that the matrix A has full row rank ($\operatorname{rank} A = m$). Then $\overline{x}$, the unique minimum point of (QP) (see Exercise 7.73), is fully characterized by the relations

$$\begin{aligned} &A\overline{x}^t = b^t \\ &\exists \mu \in \mathbb{R}^m \text{ such that } \nabla f(\overline{x}) + \mu A = 0, \end{aligned}$$

which finally give

$$\mu^t = -(AQ^{-1}A^t)^{-1}(b + AQ^{-1}c^t)$$

and

$$\overline{x}^t = -Q^{-1}c^t + Q^{-1}A^t(AQ^{-1}A^t)^{-1}(b + AQ^{-1}c^t).$$

Remark also that the first-order optimality conditions for $\overline{x}$ can be written under matricial form as follows:

$$\begin{pmatrix} Q & A^t \\ A & 0 \end{pmatrix} \begin{pmatrix} \overline{x}^t \\ \overline{\mu}^t \end{pmatrix} = \begin{pmatrix} -c^t \\ b^t \end{pmatrix}. \tag{6.2.5}$$

Rewrite (6.2.5) in a form more useful for computation: take $\overline{x} = x + y$, where x is an estimate of the solution and y is the desired step. Then (6.2.5) becomes

$$\begin{pmatrix} Q & A^t \\ A & 0 \end{pmatrix} \begin{pmatrix} y^t \\ \overline{\mu}^t \end{pmatrix} = \begin{pmatrix} d^t \\ e^t \end{pmatrix}, \tag{6.2.6}$$

where

$$e^t = -Ax^t + b^t, \quad d^t = -c^t - Qx^t, \quad y = \overline{x} - x.$$

The previous comments show that, in order to find the unique global solution of (QP), we must solve the linear system (6.2.6). A first observation is that if $p \geq 1$, the Karush-Kuhn-Tucker matrix

$$K := \begin{pmatrix} Q & A^t \\ A & 0 \end{pmatrix}$$

is always indefinite. One option is to use a triangular factorization, as the QR (Householder) factorization, or to use the so-called Schur-complement method. For details see the book (Nocedal and Wrightm, 2006).

Coming back to the general case of constrained optimization problems with equality constraints, recall that the Lagrangian of (P) is the function

$$L(x,\mu) = f(x) + \sum_{j=1}^{m} \mu_j h_j(x).$$

Denote the Jacobian matrix of the constraints by $A(x)$, i.e.,

$$A(x) = \begin{pmatrix} \nabla h_1(x) \\ \dots \\ \nabla h_m(x) \end{pmatrix}.$$

As shown by Theorem 3.2.6, the first order Karush-Kuhn-Tucker conditions for the problem (P) can be written as

$$F(x,\mu) := \begin{pmatrix} \nabla f(x) + \mu A(x) \\ h(x) \end{pmatrix} = 0. \tag{6.2.7}$$

One method is to solve the nonlinear equation (6.2.7) by the use of Newton's method. We have

$$\nabla F(x,\mu) = \begin{pmatrix} \nabla^2_{xx} L(x,\mu) & A(x)^t \\ A(x) & 0 \end{pmatrix},$$

hence the Newton step, according to (6.1.5), is

$$\begin{pmatrix} x^t_{k+1} \\ \mu^t_{k+1} \end{pmatrix} = \begin{pmatrix} x^t_k \\ \mu^t_k \end{pmatrix} - \begin{pmatrix} \nabla^2_{xx} L(x_k,\mu_k) & A(x_k)^t \\ A(x_k) & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} (\nabla f(x_k) + \mu_k A(x_k))^t \\ (h(x_k))^t \end{pmatrix}. \tag{6.2.8}$$

Of course, in order that the Karush-Kuhn-Tucker matrix

$$K(x_k,\mu_k) := \begin{pmatrix} \nabla^2_{xx} L(x_k,\mu_k) & A(x_k)^t \\ A(x_k) & 0 \end{pmatrix}$$

is nonsingular, we suppose that the constraint Jacobian $A(x)$ has full row rank, and that the matrix $\nabla^2_{xx} L(x,\mu)$ is positive definite on the tangent space of the constraints, i.e.,

$$\left\langle (\nabla^2_{xx} L(x,\mu) d^t)^t, d \right\rangle > 0, \ \forall d \neq 0 \text{ s.t. } A(x) d^t = 0. \tag{6.2.9}$$

Another way to view the iterations (6.2.8) is to consider the quadratic problem bellow at each iterate (x_k,μ_k) :

$$\begin{array}{rl} \min_y & \frac{1}{2} \left\langle (\nabla^2_{xx} L(x_k,\mu_k) y^t)^t, y \right\rangle + \left\langle \nabla f(x_k), y \right\rangle + f(x_k) \\ \text{subject to} & A(x_k) y^t + (h(x_k))^t = 0. \end{array} \tag{6.2.10}$$

Under the assumptions made, we know by the comments before that this problem has a unique solution y_k for which there is a multiplier l_k such that:

$$\begin{gathered}(\nabla^2_{xx}L(x_k,\mu_k)y_k^t)^t + \nabla f(x_k) - l_k A(x_k) = 0,\\ A(x_k)y_k^t + (h(x_k))^t = 0.\end{gathered} \tag{6.2.11}$$

Moreover, the pair (y_k, l_k) can be identified with the one of (6.2.8). To see this, denote $y'_k := x_{k+1} - x_k$ and rewrite (6.2.8) as

$$\begin{pmatrix} \nabla^2_{xx}L(x_k,\mu_k) & A(x_k)^t \\ A(x_k) & 0 \end{pmatrix}\begin{pmatrix} (y'_k)^t \\ \mu^t_{k+1} - \mu^t_k \end{pmatrix} = \begin{pmatrix} -(\nabla f(x_k) + \mu_k A(x_k))^t \\ -(h(x_k))^t \end{pmatrix}.$$

By subtracting $\mu_k A(x_k)$ in both sides of the previous relation, we get

$$\begin{pmatrix} \nabla^2_{xx}L(x_k,\mu_k) & A(x_k)^t \\ A(x_k) & 0 \end{pmatrix}\begin{pmatrix} (y'_k)^t \\ \mu^t_{k+1} \end{pmatrix} = \begin{pmatrix} -(\nabla f(x_k))^t \\ -(h(x_k))^t \end{pmatrix}.$$

Hence, by the nonsingularity of the Karush-Kuhn-Tucker matrix $K(x_k, \mu_k)$ and by relations (6.2.11), we obtain that $y_k = y'_k$ and $\mu_{k+1} = l_k$.

Hence, the new iterate (x_{k+1}, μ_{k+1}) can be defined either as solution of the quadratic program (6.2.10), or as the Newton type iterate given by (6.2.8) applied to the optimality conditions of the problem.

We close our consideration with a result about the rate of convergence. Recall that the set of critical directions is in our case

$$C(\overline{x},\overline{\mu}) = \left\{u \in \mathbb{R}^p \mid \nabla h_j(\overline{x})(u) = 0, \text{ for every } j \in \overline{1,m}.\right\}$$

Theorem 6.2.4. *Suppose $\overline{x}$ is a local solution of the problem*

$$\min f(x), \quad \textit{subject to } h(x) = 0,$$

such that f and h are twice continuously differentiable functions with Lipschitz continuous second derivatives. Moreover, suppose that the linear independence condition holds at $\overline{x}$, and that

$$\nabla^2_{xx}L(\overline{x},(\overline{\lambda},\overline{\mu}))(u,u) > 0, \quad \forall u \in C(\overline{x},\overline{\mu}) \setminus \{0\}.$$

Then, if (x_0, μ_0) is sufficiently close to $(\overline{x}, \overline{\mu})$, then the sequence generated by (6.2.8) converges quadratically to $(\overline{x}, \overline{\mu})$.

Proof The proof follows from Theorem 6.1.5, because (6.2.8) is the Newton's method applied to the nonlinear system $F(x, \mu) = 0$, where F is given by (6.2.7). □

6.2.2.2 Interior-point methods

In the approach we present here, for the problem (P) defined in the second section of Chapter 3, we consider only inequality constraints. Therefore, we take $g : \mathbb{R}^p \to \mathbb{R}^n$ and

$$M := \{x \in \mathbb{R}^p \mid g(x) \leq 0\}.$$

We denote

$$\begin{aligned}\operatorname{strict} M &:= \{x \in \mathbb{R}^p \mid g(x) < 0\} \\ &= \left\{x \in \mathbb{R}^p \mid g_i(x) < 0,\ \forall i \in \overline{1,n}\right\}.\end{aligned}$$

Let us observe that, in general, strict M does not coincide with the interior of M (it is enough to consider $g : \mathbb{R} \to \mathbb{R}^2$ with $g(x) = (-x^2, -x-1)$, since in this case strict $M = \operatorname{int} M \setminus \{0\}$).

The main idea is to transform the constrained problem into an unconstrained one through a penalization of the objective function f by an auxiliary function that contains the constraints. In fact, this also happens when one introduces the Lagrangian function, but then some parameters (λ, μ) depending on the solution were in force. At this moment, we consider the function (called the logarithmic barrier of (P)), $B(x, \mu) : \operatorname{strict} M \times (0, \infty) \to \mathbb{R}$,

$$B(x, \mu) := f(x) - \mu \sum_{i=1}^{n} \ln(-g_i(x)).$$

It is clear that this function preserves the smoothness properties of the problem data. On the other hand, if a solution $\overline{x}$ lies in strict M, then for x close to $\overline{x}$, $\lim_{\mu \to 0} B(x, \mu) = f(x)$. If $\overline{x}$ lies in $M \setminus \operatorname{strict} M$ then at least one constraint is active, so for a sequence $(x_k) \to \overline{x}$,

$$\lim_k \left(f(x_k) - \mu \sum_{i=1}^{n} \ln(-g_i(x_k)) \right) = \infty.$$

Consequently, a coercivity condition similar to that in Proposition 3.1.9 could be fulfilled (under some conditions). The idea (in both situations) is to ensure the existence of a unconstrained minimum for $B(\cdot, \mu)$ on strict M, denoted x_μ, then, for $\mu \to 0$, to show that x_μ converges towards a minimum of the problem (P). The below figure offers an intuitive image for this remark for the case of the function $f : \mathbb{R} \to \mathbb{R}$, $f(x) = e^x - x^3$ under the constrains $g_1(x) = 1 - x \leq 0$, $g_2(x) = x - 3 \leq 0$. The minimum is $\overline{x} = 3$, and the Figure 6.1 presents, besides the graph of f, the graphs of $B(x, 3^{-1})$ and $B(x, 7^{-1})$.

In order to prepare the main result, we need an additional preliminary discussion. We have already said that if we have minimum points that are close one to each other, then it is difficult to design algorithms which make the distinction between them, and in order to approach the desired point one should start from appropriate initial data. The most unpleasant situation occurs when a minimum point is not isolated in the set of local minima. Such an example for a C^2 function is given below. Let $f : \mathbb{R} \to \mathbb{R}$,

$$f(x) = \begin{cases} x^4 \left(2 + \cos \frac{1}{x}\right), & x \neq 0 \\ 0, & x = 0. \end{cases}$$

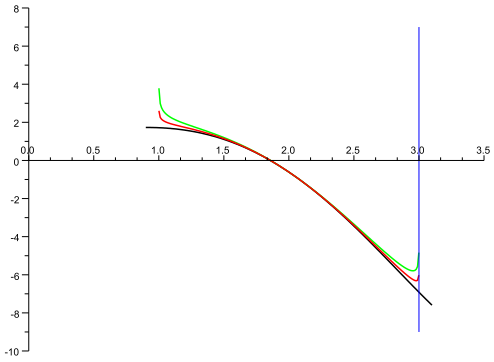

Figure 6.1: Barrier method illustration.

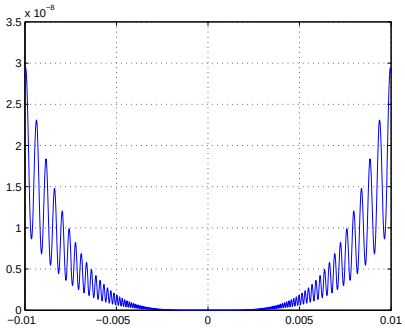

Figure 6.2: Non isolated minimum.

This function has a global minimum at 0, but there is a sequence of local minima which converges to 0 (see Figure 6.2).

It is now necessary to formulate some conditions concerning topological properties of the set of minima. We start with a definition.

Definition 6.2.5. *Let $A \subset B$. We say that A is an isolated subset of B if there exists a closed set E with $A \subset \operatorname{int} E$ and $B \cap E = A$.*

The next result holds.

Proposition 6.2.6. *Let $M \subset \mathbb{R}^p$ and $\varphi : M \to \mathbb{R}$. We denote by N a set (that we suppose to be nonempty) of a local minima of φ on M, for which the value of the function is the*

same (denoted by $\overline{\varphi}$). Suppose that N^* is an isolated compact subset of N. Then there exists a compact set C such that $N^* \subset \operatorname{int} C$ and $\varphi(x) > \overline{\varphi}$ for every $x \in (M \cap C) \setminus N^*$.

Proof According to the preceding definition, there exists a closed set E with $N^* \subset \operatorname{int} E$ and $N \cap E = N^*$. Since N^* consists only of minima, for every $x \in N^*$ there exists an open neighborhood V_x of x with

$$\varphi(x) = \overline{\varphi} \leq f(u), \ \forall u \in M \cap V_x.$$

Then the set $G := \bigcup_{x \in N^*} V_x$ is open and includes N^*, and $\varphi(u) \geq \overline{\varphi}$ for every $u \in M \cap G$. Then, by the compactness of N^* and the openness of $G \cap \operatorname{int} E$, there exists a compact set C with

$$N^* \subset \operatorname{int} C \subset C \subset G \cap \operatorname{int} E \subset G \cap E$$

(it can be shown quite simply that one can take C as $\left\{x \in \mathbb{R}^p \mid d(x, C) \leq n^{-1}\right\}$ for sufficiently large $n \in \mathbb{N}^*$). Clearly, $N^* \subset \operatorname{int} C \cap M$. Take now $x \in (M \cap C) \setminus N^*$. Since $x \in M \cap G$, one has $\varphi(x) \geq \overline{\varphi}$. On the other hand, since $C \subset E$, one has $x \in E \setminus N^*$, whence $x \notin N$. Therefore, $\varphi(x) \neq \overline{\varphi}$. Consequently, $\varphi(x) > \overline{\varphi}$ is the only possibility. □

We present now the main result of this section.

Theorem 6.2.7. *Suppose that f and g are continuous. Let N be the set (supposed to be nonempty) of local minima of f on M for which the value of the function is the same (denoted by $\overline{f}$), and let $(\mu_k) \subset (0, \infty)$ be a strictly decreasing sequence convergent to* 0. *We suppose that:*

(a) there exists an isolated compact subset N^* of N;

(b) $N^* \cap \operatorname{cl}(\operatorname{strict} M) \neq \emptyset$.

Then:

(i) there exists a compact set C such that $N^* \subset \operatorname{int} C$, and for every $x \in (M \cap C) \setminus N^*$, $f(x) > \overline{f}$;

(ii) there exist an infinity of numbers k for which there exists a global minimum $y_k \in \operatorname{strict} M \cap \operatorname{int} C$ without restrictions of $B(\cdot, \mu)$ on $\operatorname{strict} M \cap \operatorname{int} C$ with

$$B(y_k, \mu_k) = \min\{B(x, \mu_k) \mid x \in \operatorname{strict} M \cap C\};$$

(iii) every limit point of (y_k) is in N^*, and if (x_l) is a subsequence of (y_k) convergent to the underlying limit point, then

$$\lim_l f(x_l) = \overline{f} = \lim_l B(x_l, \mu_l).$$

Proof The first item, (i), follows easily from (a) and from Proposition 6.2.6. So, there exists a compact set C such that $N^* \subset \operatorname{int} C \cap M$, and the value of f at the points of N^* is the smallest one in $C \cap M$. Since the function $B(\cdot, \mu_k)$ verifies the assumptions of Proposition 3.1.9 for $D = \operatorname{strict} M$, there exists a global minimum for $B(\cdot, \mu_k)$ on $\operatorname{strict} M \cap C$.

At this moment, this minimum point, denoted by y_k, is not without restrictions (note that C is closed). Since (y_k) is bounded, there exists a subsequence of it, denoted (x_l), which has a limit, denoted by x_∞, lying in $M \cap C$. Therefore x_∞ is a feasible point. We show now that $x_\infty \in N^*$. Otherwise, from the preceding part, $f(x_\infty) > \overline{f}$. In order to arrive at a contradiction, we use (b). Take $x^* \in N^* \cap \mathrm{cl}(\mathrm{strict}\, M)$. We distinguish two situations, and in both cases we show that there exists $x_{\mathrm{int}} \in C \cap \mathrm{strict}\, M$ with $f(x_\infty) > f(x_{\mathrm{int}})$. Firstly, we consider that $x^* \in \mathrm{strict}\, M$. Then $f(x_\infty) > f(x^*)$, so we can take $x_{\mathrm{int}} = x^*$. Suppose that $x^* \in \mathrm{cl}(\mathrm{strict}\, M) \setminus \mathrm{strict}\, M$. From $N^* \subset \mathrm{int}\, C$, we deduce that $x^* \in \mathrm{int}\, C$. Since $f(x_\infty) > \overline{f} = f(x^*)$ and f is continuous, there exists a neighborhood V of x^* such that $f(x_\infty) > f(x)$ for every $x \in V$. In particular, there exists $x_{\mathrm{int}} \in C \cap \mathrm{strict}\, M$ with the desired property. Thus, in every situation, there exists $x_{\mathrm{int}} \in C \cap \mathrm{strict}\, M$ with $f(x_\infty) > f(x_{\mathrm{int}})$. Again, from the continuity of f, for every l big enough, $f(x_l) > f(x_{\mathrm{int}})$. But x_l is a global minimum on $C \cap \mathrm{strict}\, M$ for $B(\cdot, \mu_l)$, so

$$f(x_l) - \mu_l \sum_{i=1}^{n} \ln(-g_i(x_l)) \leq f(x_{\mathrm{int}}) - \mu_l \sum_{i=1}^{n} \ln(-g_i(x_{\mathrm{int}})).$$

Since $\sum_{i=1}^{n} \ln(-g_i(x_{\mathrm{int}})) \in \mathbb{R}$, passing to the limit as $l \to \infty$,

$$\lim_l \left(f(x_{\mathrm{int}}) - \mu_l \sum_{i=1}^{n} \ln(-g_i(x_{\mathrm{int}})) \right) = f(x_{\mathrm{int}}).$$

If $x_\infty \in \mathrm{strict}\, M$, as above,

$$\lim_l \left(f(x_l) - \mu_l \sum_{i=1}^{n} \ln(-g_i(x_l)) \right) = f(x_\infty),$$

whence

$$f(x_\infty) \leq f(x_{\mathrm{int}}),$$

in contradiction to the step before.

If $x_\infty \in (M \cap C) \setminus \mathrm{strict}\, M$, adding $-\mu_l \sum_{i=1}^{n} \ln(-g_i(x_{\mathrm{int}}))$ in the inequality $f(x_l) > f(x_{\mathrm{int}})$, we infer, by the use of the relation before,

$$\begin{aligned} f(x_l) - \mu_l \sum_{i=1}^{n} \ln(-g_i(x_{\mathrm{int}})) &> f(x_{\mathrm{int}}) - \mu_l \sum_{i=1}^{n} \ln(-g_i(x_{\mathrm{int}})) \\ &\geq f(x_l) - \mu_l \sum_{i=1}^{n} \ln(-g_i(x_l)), \end{aligned}$$

whence

$$-\sum_{i=1}^{n} \ln(-g_i(x_{\mathrm{int}})) > -\sum_{i=1}^{n} \ln(-g_i(x_l)).$$

For $l \to \infty$, in the left-hand side we get a real number, and in the right-hand side we get $+\infty$, which is a contradiction. We conclude that the assumption made was false,

so $f(x_\infty) = \overline{f}$, that is $x_\infty \in N^*$. We conclude that every limit point of (y_k) satisfies this property. From the fact that $x_\infty \in N^*$, we infer that $x_\infty \in \operatorname{int} C$, so, eventually, x_l belongs to $\operatorname{int} C$. This means that the geometric restriction $x \in C$ is not active, whence x_l is a minimum without constraints for $B(\cdot, \mu_l)$ on strict $M \cap \operatorname{int} C$. The second item, (ii), is proved.

The first part of (iii) is obvious, since $\lim_l f(x_l) = f(x_\infty) = \overline{f}$. It remains to show that $\overline{f} = \lim_l B(x_l, \mu_l)$.

If $x_\infty \in \operatorname{strict} M$, then the sum $\sum_{i=1}^n \ln(-g_i(x_l))$ is finite for all big l. Then

$$\lim_l B(x_l, \mu_l) = f(x_\infty) = \overline{f}.$$

Now, suppose that $x_\infty \notin \operatorname{strict} M$, so at least a constraint goes to 0 on (x_l) for $l \to \infty$. Since for every l, x_l is the minimum point for $B(\cdot, \mu_l)$ on strict $M \cap C$, we have

$$B(x_l, \mu_l) \le B(x_{l+1}, \mu_l) \text{ and } B(x_{l+1}, \mu_{l+1}) \le B(x_l, \mu_{l+1}).$$

We multiply the first inequality by $\mu_l^{-1}\mu_{l+1} \in (0, 1)$ and we add the second inequality. We get

$$f(x_{l+1})\left(1 - \frac{\mu_{l+1}}{\mu_l}\right) \le f(x_l)\left(1 - \frac{\mu_{l+1}}{\mu_l}\right),$$

so $f(x_{l+1}) \le f(x_l)$.

For every l big enough, $\sum_{i=1}^n \ln(-g_i(x_l)) < 0$, whence $B(x_l, \mu_l) > f(x_l)$. Since $f(x_{l+1}) \le f(x_l)$ and $\lim_l f(x_l) = f(x_\infty) = \overline{f}$, we infer that the sequence $(B(x_l, \mu_l))$ is lower bounded. On the other hand,

$$0 < \mu_{l+1} < \mu_l \text{ and } \sum_{i=1}^n \ln(-g_i(x_l)) < 0,$$

so for l big enough,

$$-\mu_{l+1} \sum_{i=1}^n \ln(-g_i(x_l)) < -\mu_l \sum_{i=1}^n \ln(-g_i(x_l)),$$

hence

$$B(x_l, \mu_{l+1}) < B(x_l, \mu_l).$$

Therefore,

$$B(x_{l+1}, \mu_{l+1}) \le B(x_l, \mu_{l+1}) \le B(x_l, \mu_l).$$

We deduce that the sequence $(B(x_l, \mu_l))$ is monotone, and converges towards a limit denoted by $\overline{B}$. The inequality $\overline{B} \ge \overline{f}$ is ensured by the above considerations. We suppose, by way of contradiction, that $\overline{B} > \overline{f}$, and we take $\varepsilon := 2^{-1}(\overline{B} - \overline{f}) > 0$. From the continuity of f, there exists a neighborhood V of x_∞ for which

$$f(x) < f(x_\infty) + \varepsilon = \overline{B} - \varepsilon, \ \forall x \in V.$$

There exists at least one point of $x' \in \text{strict}\, M$ in V. So,

$$B(x_l, \mu_l) \le B(x', \mu_l) = f(x') - \mu_l \sum_{i=1}^{n} \ln(-g_i(x')).$$

But $\sum_{i=1}^{n} \ln(-g_i(x')) \in \mathbb{R}$, and for every l big enough,

$$-\mu_l \sum_{i=1}^{n} \ln(-g_i(x')) < 2^{-1}\varepsilon.$$

But

$$f(x') < \overline{B} - \varepsilon,$$

so

$$B(x_l, \mu_l) < \overline{B} - \varepsilon + 2^{-1}\varepsilon = \overline{B} - 2^{-1}\varepsilon,$$

which contradicts $B(x_l, \mu_l) \to \overline{B}$. So $\overline{B} = \overline{f}$ and the proof is complete. □

The hypotheses of the above result are quite weak. It is remarkable that the problem data are supposed to be only continuous. We can say that the assumption (b) is the most demanding one, but it is quite natural. However, there exist situations when this assumption is not fulfilled. To see this, it is sufficient to consider the problem of minimizing the function $f : \mathbb{R} \to \mathbb{R}$, $f(x) = (x+1)^2$ under the constraint $g(x) \le 0$, where $g : \mathbb{R} \to \mathbb{R}^2$, $g(x) = (x(1-x), -x)$. Then $M = \{0\} \cup [1, \infty)$, and the only solution is $\overline{x} = 0$. But strict $M = (1, \infty)$, hence $N^* \cap \text{cl}(\text{strict}\, M) = \emptyset$.

The good part of the conclusion is that we established convergence to the solutions of the problem (P) without qualification conditions. However, the sequence (y_k) can be divergent: we know that it has convergent subsequences. Let us remark that

$$\nabla_x B(x, \mu) = \nabla f(x) + \sum_{i=1}^{n} \frac{\mu}{g_i(x)} \nabla g_i(x).$$

If $\overline{x}$ is a minimum point without restrictions for $B(\cdot, \mu)$, then $\nabla_x B(\overline{x}, \mu) = 0$, and if μ is small enough and $\overline{x}$ is close to the solution of the problem (P), then the Lagrange multipliers λ_i can be approximated by $\mu g_i^{-1}(\overline{x})$.

Subsequently, Theorem 6.2.7 can be combined with other hypotheses and techniques in order to obtain several enhancements of the conclusions and in order to include the equality constraints in the discussion.

6.3 Scientific Calculus Implementations

In this section we aim at illustrating the theoretical discussions above through their numerical implementation in the scientific calculus software Matlab. Thus we verify, from a practical point of view, the results we have proved before, and we split our exemplifications into two categories of codes: on one hand, there are codes which use some default functions of Matlab (which, in turn, encapsulate various numerical algorithms) and, on the other hand, we directly implement many of the studied algorithms.

1. (least squares method - linear dependence) The system obtained by the modelling the least squares method for the affine dependence $v = at + b$, has, as established before (Section 3.4), a unique solution:

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{N} t_i^2 & \sum_{i=1}^{N} t_i \\ \sum_{i=1}^{N} t_i & N \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^{N} t_i v_i \\ \sum_{i=1}^{N} v_i \end{pmatrix}.$$

Let us take the concrete example: $N = 5$, $t_1 = 0$, $t_2 = 1$, $t_3 = 2$, $t_4 = 3$, $t_5 = 4$ and $v_1 = 1$, $v_2 = 2.5$, $v_3 = 5.1$, $v_4 = 6.7$, $v_5 = 8.3$.

For the calculus of the parameters a, b we implement the following Matlab code:

```
t=[0,1,2,3,4];
v=[1,2.5,5.1,6.7,8.3];
A=[sum(t.^2) sum(t)
sum(t) 5]
B=[sum(t.*v)
sum(v) ]
U=A^(-1)*B
x=linspace(0,4.5,90);
y=U(1)*x+U(2);
plot(t,v,'ro','Linewidth',2);
hold on;
plot(x,y);
```

Then we obtain the values 1.88 and 0.96 as well as the figure below (Figure 6.3).

2. (least squares method - nonlinear dependence) A dedicated Matlab function for solving nonlinear problems coming from the application of the least squares method is the function `lsqnonlin` which can be used with the syntax `[x,resnorm]=lsqnonlin(@fun, x0)`, where `fun` is a function which depends upon the parameters of the model. Therefore, `lsqnonlin` takes as objective function the sum of the squares of the components of the vector `fun` and returns both the minimum point and the minimal value.

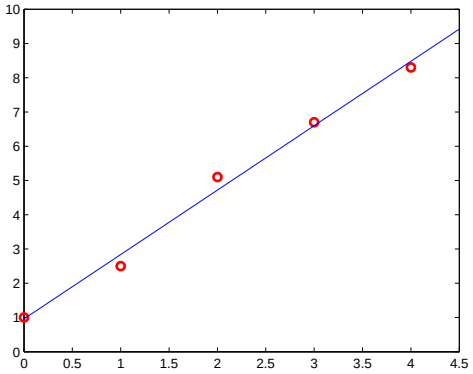

Figure 6.3: Least squares method: linear case.

Let us suppose that from the direct observation of a specific physical phenomenon at the moments t_i we get the m_i data, as in the table below.

i	1	2	3	4	5	6
t_i	0.1	0.3	0.5	0.7	0.8	0.9
m_i	0.7	1.5	4.5	22.3	94	387.9

Moreover, we suppose that one can observe a behaviour of type $(1-t)^{x_1}$ around 0, and a behaviour of type t^{-x_2} around 1. The suggested continuous model at every moment t is $f : \mathbb{R}^3 \to \mathbb{R}$,

$$f(x) = x_3(1-t)^{x_1}t^{-x_2}.$$

The function `lsqnonlin` will minimize the objective function

$$x \to \sum_{i=1}^{6} \left[m_i - x_3(1-t)^{x_1}t^{-x_2}\right]^2.$$

The program consists of a function file with the code

```
function z=fun(p)
measures=[0.1 0.7;0.3 1.5;0.5 4.5;0.7 22.3; 0.8 94; 0.9 387.9];
z=measures(:,2)-p(3)*(1-measures(:,1)).^p(1)./measures(:,1).^p(2)
```

and the main file as follows:

```
p0=[1 1 1];
[p,difference]=lsqnonlin(@fun,p0)
measures=[0.1 0.7;0.3 1.5;0.5 4.5;0.7 22.3; 0.8 94; 0.9 387.9];
```

which generates

```
p = -0.9368 -6.7935 91.8653
difference =27.7334
```

while the additional lines

```
plot(measures(:,1),measures(:,2),'o','Linewidth',2)
hold on;
model= '91.8653*(1-x)^(-0.9368)/x^(-6.7935)';
fplot(model,[0 0.95],'-r')
```

give, on the same figure, the discrete (measured) model and the continuous model (see the figure below).

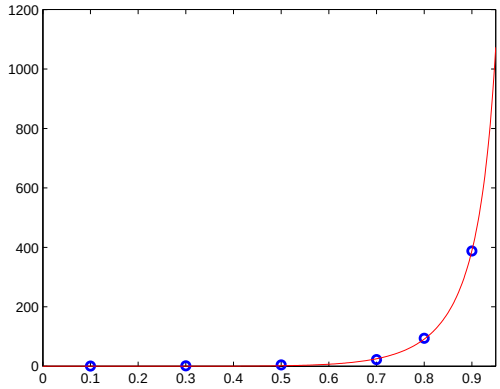

Figure 6.4: Least squares method: nonlinear case.

3. (fixed points - basic approximations) We want to approximate the solution of the equation $\cos x = x$, $x \in [0, 1]$. Clearly, a solution of this equation is a fixed point of the restriction of the function cos to the interval $[0, 1]$. The cos function is a contraction, since $\sup_{x\in[0,1]} \left|\cos' x\right| = \sin 1 < 1$. Moreover, every calculator gives us the approximate value of the contraction constant: $\sin 1 \simeq 0.84147$, so

$$|\cos x - \cos y| \leq 0.8415\,|x - y|\,.$$

Thus, according to the Banach Principle, there exists a unique solution (denoted $\overline{x}$) of the mentioned equation which can be approximated arbitrarily well by the sequence of the Picard iterations starting from every initial date $x_0 \in [0, 1]$. We intend to investigate how accurately the solution has been approximated after a given number of iterations. One may ask how many iterations are needed to obtain the value of $\overline{x}$ with

an error smaller than $\frac{1}{1000}$. Answering this question is now possible in view of the estimations concerning the speed of convergence of Picard approximations in the Banach Principle. We recall that:

$$|x_n - \overline{x}| \leq |x_1 - x_0| \frac{\lambda^n}{1-\lambda} \tag{6.3.1}$$

$$|x_n - \overline{x}| \leq \frac{\lambda}{1-\lambda} |x_n - x_{n-1}| \tag{6.3.2}$$

This discussion helps us to understand these inequalities and to obtain the desired approximation of $\overline{x}$ starting from $x_0 = 0$. Taking into account (6.3.1) and (6.3.2), we have

$$|x_n - \overline{x}| \leq \frac{0.8415^n}{1-0.8415} \text{ and } |x_n - \overline{x}| \leq \frac{0.8415}{1-0.8415} |x_n - x_{n-1}|.$$

In the Matlab programs given below, one sees that the second estimation is better than the first one, since the value $\frac{0.8415^n}{1-0.8415}$ is less than 0.001 starting from $n = 51$, while the right-side member in the second part is under 0.001 faster (for $n = 22$). For instance, for $n = 22$ in the second relation, we get

$$|x_{22} - \overline{x}| < 0.0009,$$

so $\overline{x} \in (x_{22} - 0.0009, x_{22} + 0.0009)$, that is $\overline{x}$ is between 0.7381 and 0.7399.
The Matlab programs are as follow :

```
lambda=0.8415;
c=1/(1-lambda)
i=0;
while c*lambda>0.001
i=i+1; c=c*lambda;
end
```

which gives

```
-> disp(i+1); disp(c*lambda);
51.
0.0009500
```

and, respectively,

```
u=0; i=0;
while (0.8415/(1-0.8415))*abs((u-cos(u)))>0.001
    i=i+1; u=cos(u);
end
disp(i+1); disp((0.8415/(1-0.8415))*abs((u-cos(u))))
```

which gives

```
-> disp(i+1); disp((0.8415/(1-0.8415))*abs((u-cos(u))))
22.
0.0008820.
```

Again, the approximation process is faster if one starts form an initial value close to $\overline{x}$. For instance, taking $x_0 = 0.7$, the estimations

$$|x_n - \overline{x}| \leq \frac{0.8415^n}{1 - 0.8415} \left|\cos(0.7) - 0.7\right| \text{ and } |x_n - \overline{x}| \leq \frac{0.8415}{1 - 0.8415} |x_n - x_{n-1}|$$

give an error less than 0.001 for $n = 35$, and $n = 16$, respectively. These new values can be verified easily, by making the obvious modifications in the above programs.

4. (speed of convergence of Picard iterations: the case $f'(\overline{x}) \in (0, 1)$) Let us consider the function $f : [0, \infty) \to [0, \infty)$ given by $f(x) = \frac{1}{1+x^2}$. We have seen that this is a contraction, has a unique fixed point which is the unique positive solution of the equation $x^3 + x - 1 = 0$, and the sequence of the Picard iterations satisfies:

$$\frac{x_{n+1} - \overline{x}}{x_n - \overline{x}} \overset{n\to\infty}{\to} f'(\overline{x}) = \frac{-2\overline{x}}{(1 + \overline{x}^2)^2} = -2\overline{x}^3 \in (0, 1).$$

Let us study the speed of convergence of this sequence by means of a Matlab program. The stopping criterion is the attainment of a maximum number of iterations (1000), or the situation where the absolute value of the difference between two consecutive iterations is under an admissible tolerance (10^{-7}).

```
funct='1/(1+x^2)'
tol=1e-7; maxiter=1000;
n=0; x=1; x_old=0;
%Picard
while abs(x-x_old)>tol & n<maxiter
      x_old=x; x=eval(funct); n=n+1;
end
%endPicard
disp(x);
disp(n);
```

The displayed results are:

```
->disp(u); 0.6823278 ->disp(n); 35
```

so the algorithm stopped after 35 iterations and found the approximate value of the solution as being 0.6823278, starting from the initial data $x_0 = 1$. The speed of convergence, which is a relatively good one, is due to the fact that $\left|f'(\overline{x})\right|$ is smaller than 1.

5. (speed of convergence of Picard iterations: the case $f'(\overline{x}) = 1$) For the restriction of the function $\sin x$ to the interval $[0, 1]$, which satisfies the assumptions in the Picard Theorem with $\overline{x} = 0$ as the unique fixed point, the speed of convergence dramatically changes, as every Picard iteration (x_k) satisfies

$$\frac{x_{k+1} - \overline{x}}{x_k - \overline{x}} \overset{k\to\infty}{\to} f'(\overline{x}) = 1$$

and we expect that the progress made in approaching the solution from a iteration to another is very small. In the above program, we change f and we get the results:

```
->disp(u);
0.0545930
->disp(n);
1000
```

which means that after 1000 iterations we get a quite unsatisfactory approximation of the fixed point. The situation changes insignificantly if one starts with the value $x_0 = 0.1$, closer to $\overline{x}$:

```
->disp(u); 0.0480222 ->disp(n); 1000.
```

6. (speed of convergence of Picard iterations: the case $f'(\overline{x}) = 0$) Consider the case of the function $f : [\sqrt{2}, \infty) \to [\sqrt{2}, \infty)$ given by

$$f(x) = \frac{x}{2} + \frac{1}{x}.$$

It is easy to see that f is well defined (the means inequality). Moreover,

$$\left|f'(x)\right| = \left|\frac{1}{2} - \frac{1}{x^2}\right| \leq \frac{1}{2},$$

so f is a contraction and its unique fixed point is $\overline{x} = \sqrt{2}$. One observes that $f'(\overline{x}) = 0$, so, for every nonstationary Picard iteration,

$$\frac{x_{k+1} - \overline{x}}{(x_k - \overline{x})^2} = \frac{1}{2x_k} \to \frac{1}{2\sqrt{2}} = f''(\overline{x}),$$

so we have quadratic convergence, and we expect a very good speed of convergence. We repeat the above program for the new function and we get:

```
->disp(u); 1.4142136 ->disp(n); 5
```

so the algorithm stops after only 5 iterations and we have a very good approximation of the fixed point $\overline{x} = \sqrt{2}$.

7. (the Aitken acceleration methods) We come back to the function sin (restricted to the interval $[0, 1]$) for which the convergence of the Picard iterations sequence is slow. We want to test the two Aitken acceleration methods: the weak and the strong ones (which actually work as well for the case $\left|f'(\overline{x})\right| = 1$). Briefly, for the weak method, besides a Picard sequence associated to f, one considers the sequence

$$y_k = x_k - \frac{(f(x_k) - x_k)^2}{f(f(x_k)) - 2f(x_k) + x_k}$$

which converges faster, however, without a modification of the order of convergence. For the strong method, we work directly with the sequence

$$x_{k+1} = x_k - \frac{(f(x_k) - x_k)^2}{f(f(x_k)) - 2f(x_k) + x_k}.$$

We have the code below:

```
funct='sin(x)'
functcomp='sin(sin(x))'
tol=1e-7;
maxiter=20000;
y=1;x=1;y_old=0;n=0;
%Aitken (weak)
while abs(y-y_old)>tol & n<maxiter
 y_old=y; x=eval(funct);
 y=x-((eval(funct)-x)^2)/(eval(functcomp)-2*eval(funct)+x);
n=n+1;
end
disp(x); disp(y); disp(n);
```

The result is the following:

```
 0.0122
 0.0082
20000
```

which means that for the initial data $x_0 = 1$, after 20000 Picard iterations we are not able to approximate the fixed point $\overline{x} = 0$ with an error smaller that 10^{-2} (we get 0.0122449), while this happens after 20000 weak Aitken iterations (the value is 0.0081631).

For the strong Aitken method, we do not change the Picard method, and for the code we made the following modifications: we introduce a new constant `tol1=1e-4`, and eliminate the variable `w`, and the strong Aitken method looks like:

```
funct='sin(x)'
functcomp='sin(sin(x))'
tol1=1e-4; maxiter=1000;
n=0; x=1; x_old=0;
while abs(x-x_old)>tol1 & n<maxiter
 x_old=x;
 x=x-((eval(funct)-x)^2)/(eval(functcomp)-2*eval(funct)+x);
 n=n+1;
end
disp(x);
disp(n);
```

with the result:

```
 1.8779e-004
 21
```

which is clear indication of the huge progress for the speed of convergence.

8. (particular acceleration of the Picard iterations) There are particular cases where the Picard iterations can be accelerated by means of an auxiliary function. For

instance, consider $f : \mathbb{R} \to \mathbb{R}$, $f(x) = x^3 + 4x^2 + x - 10$. By the basic methods of mathematical analysis, the equation $f(x) = x$ has only one real solution situated in the interval $[1, 2]$. Clearly, f is not a contraction. However, the application $g : [1, 2] \to \mathbb{R}$,

$$g(x) = \frac{2x^3 + 4x^2 + 10}{3x^2 + 8x}$$

satisfies the fact that the equality $g(x) = x$ is equivalent to $f(x) = x$. But

$$g'(x) = \frac{(6x^2 + 8x)(3x^2 + 8x) - (6x + 8)(2x^3 + 4x^2 + 10)}{(3x^2 + 8x)^2}$$
$$= \frac{(6x + 8)(x^3 + 4x^2 - 10)}{(3x^2 + 8x)^2}.$$

Since $\overline{x}$ is the solution of the equation $x^3 + 4x^2 - 10 = 0$, we conclude that $g'(\overline{x}) = 0$. This means that around $\overline{x}$, g is a contraction and, moreover, the convergence of the associated Picard iterations is quadratic. The next program shows that after only 9 steps, the Picard iterations of f blow up, while using g we get a good approximation of the solution after only 5 iterations.

```
tol=1e-7;maxiter=1000;
u=1; u_old=0; n=0; t=1; t_old=0; p=0;
%Picard g
while abs(u-u_old)>tol & n<maxiter
     u_old=u; u=(2*u^3+4*u^2+10)/(3*u^2+8*u); n=n+1;
end
%Picard f
while abs(t-t_old)>tol & p<maxiter
     t_old=t; t=t^3+4*t^2+t-10; p=p+1;
end
disp(u); disp(n); disp(t); disp(p);
```

The result is:

```
-> disp(u); disp(n); disp(t); disp(p);
1.36523
5.
Nan
9.
```

9. (Newton method - one root) We test the Newton method for the function $f : \mathbb{R} \to \mathbb{R}$ given by the relation

$$f(x) = x + e^x + \frac{10}{1 + x^2} - 5$$

which has a solution of order 1 in $(-2, 0)$, as one can also observe in the figure below:

Starting with the initial data $x_0 = 1.5$ we converge rapidly to this solution, as shown in the next code:

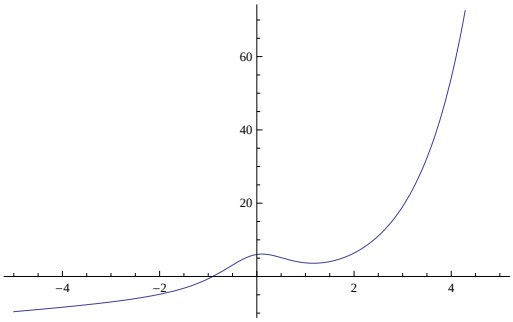

Figure 6.5: The graph of $x + e^x + \frac{10}{1+x^2} - 5$.

```
f_el='x+exp(x)+10/(1+x^2)-5';
syms x
g=diff(f_el);tol=1e-5;maxiter=1000;
x=1.5;x_old=0;n=1;
%Newton
while abs(x-x_old)>tol & n<maxiter
    x_old=x;x=x-eval(f_el)/eval(g);n=n+1;
end
x
n
eval(f_el)
```

which returns:

```
x = -0.9046
n = 35
ans = 8.8818e-016
```

If one starts with the initial date $u = -1.5$, then one gets the approximation after only 5 iterations.

10. (distance to a generalized ellipsoid) Let have a look again at the nonlinear equation whose solution gives the number needed to compute the projection of a point on a generalized ellipsoid. Let us implement the Newton method for approximation of the square of the equation

$$\sum_{i=1}^{p} \frac{a_i^2 v_i^2}{\left(a_i^2 + \lambda\right)^2} = 1,$$

for $p = 5$, $a_i = 6 - i$, $v_i = 9$, $i \in \overline{1,5}$. The code:

```
f_el='2500/((25+x)^2)+1600/((16+x)^2)+900/((9+x)^2)
+400/((4+x)^2)+100/((1+x)^2)-1';
syms x
```

```
g=diff(f_el);tol=1e-5;maxiter=1000;
x=97;x_old=0;n=1;
%Newton
while abs(x-x_old)>tol & n<maxiter
x_old=x;x=x-eval(f_el)/eval(g);n=n+1;
end
x
n
eval(f_el)
```

returns:

```
x =57.5719; n =9; ans =-1.1102e-016
```

11. (Newton method- several roots) Let us consider $f : \mathbb{R} \to \mathbb{R}$, $f(x) = e^x - 2x^2$. It is not very difficult to see that f has three real roots, among which one is negative and two are positive. We apply the Newton method (as in the previous examples) for various initial data.

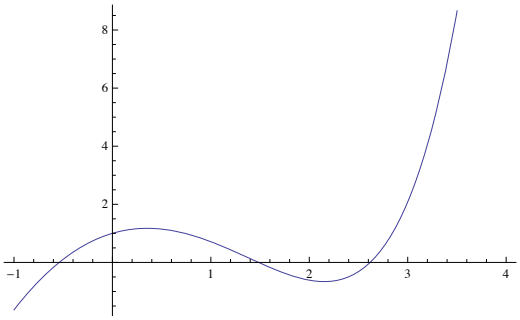

Figure 6.6: The graph of $e^x - 2x^2$ on $[-1, 4]$.

If we start with $u = 0$ we obtain:

```
x = -0.5398; n = 7.
```

Therefore, starting from 0 we find the approximation of the negative root after 7 iterations.

Starting from $u = 1$, we get:

```
x = 1.4880; n = 5
```

and for $u = 3$, we get:

```
x = 2.6179; n = 6.
```

12. (Newton method for fixed points) Besides Picard iterations which, in general, have linear convergence, we can use the Newton's method for approximating fixed points with quadratic orders of convergence. For instance, in the above case of

the fixed point of cos in [0, 1], according to the theory, it is sufficient to approximate the solution of the equation $\cos x = x$, and this means that we should consider the Picard iterations associated to

$$g(x) = x + \frac{\cos x - x}{1 + \sin x}.$$

The next Matlab program proves the advantages of this new approach:

```
f_el='x+(cos(x)-x)/(1+sin(x))';
syms x
tol=1e-5;maxiter=1000;
x=1;x_old=0;n=1;
%Newton
while abs(x-x_old)>tol & n<maxiter
    x_old=x;x=eval(f_el);n=n+1;
end
%Picard
t=1;t_old=0; p=0;
while abs(t-t_old)>tol & p<maxiter
     t_old=t; t=cos(t); p=p+1;
end
x
n
t
p
```

and the results are:

```
x = 0.7391
n = 5
t = 0.7391
p = 29
```

13. (proximal point algorithm) We implement now the proximal point algorithm. Let $f : \mathbb{R} \to \mathbb{R}$,

$$f(x) = \frac{x^4}{4} + \frac{x^2}{2} - 3x + 1.$$

This is a convex function since $f''(x) = 3x^2 + 1 > 0$ for every $x \in \mathbb{R}$. Its graph is below (Figure 6.7). In order to approximate the minimum point we use the proximal point algorithm in the following code:

we define in a function file:

```
function F = prox_fun(x,y)
     F = x.^3+2.*x-3-y;
 end
```

and in a M-file we use the default function `fsolve` in its parametric version:

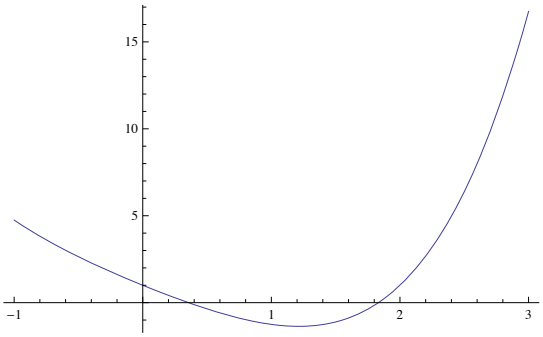

Figure 6.7: The graph of $\frac{x^4}{4} + \frac{x^2}{2} - 3x + 1$.

```
maxiter=100;n=0;y=-10;
while n<maxiter
     y = fsolve(@(x) prox_fun(x,y),[-1;y(1)]);
     n=n+1;
end
disp(y(1));
```

We get the approximate value $\overline{x} = 1.213411662762243$.

14. (the algorithm of steepest descent direction) Let us consider the steepest descent method. Usually, the direction is chosen as

$$p_k = -\frac{\nabla f(x_k)}{\left\|\nabla f(x_k)\right\|}$$

if $\left\|\nabla f(x_k)\right\| \neq 0$ and, concerning the step α_k, we use only the Armijo condition

$$f(x_k + \alpha_k p_k) < f(x_k) + c_1 \alpha_k \nabla f(x_k)(p_k).$$

Therefore, we test, at every step, if the condition is fulfilled and, contrary, we multiply α_k by a factor less than 1, in order to obtain a smaller step.

For illustration, we chose the function $f : \mathbb{R} \to \mathbb{R}$, $f(x) = e^x - 2x^2$. This function has a local minimum point close to 2.

We have the code:

```
funct=@(x) exp(x)-2*x^2;
derivative=@ (x)exp(x)-4*x;
tol=1e-7; maxiter=20;u=5; u_old=1; n=0; factor=0.5;
c1=0.01; alpha=1;
 while abs(derivative(u))>tol
 u_old=u;
 u=u-alpha*derivative(u)/abs(derivative(u));
 while funct(u)>=funct(u_old)+c1*alpha*abs(derivative(u_old))
```

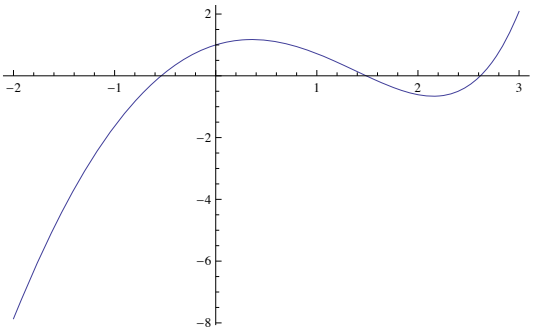

Figure 6.8: The graph of $e^x - 2x^2$ on $[-2, 3]$.

```
 alpha=factor*alpha;
 u=u-alpha*derivative(u)/abs(derivative(u));
 end
n=n+1;
 end
 u
n
 alpha
 derivative(u)
```

and the results are

```
u = 2.153292357921600
n = 15
alpha = 5.960464477539063e − 008
ans = −2.854974923138798e − 008
```

So, the algorithm stops after it obtains a convenient approximation of the minimum. The evolution of the decreasing of the gradient absolute value is also interesting.

15. (the algorithm of steepest descent direction: several variables) The next example refers to the approximation of the minimum point for the Rosenbrock function (see Exercise 7.51). In code below, we implement the steepest descent method for this function, with the chosen step following an heuristic rule. The graphic representations we get give us informations about the construction if the iterates. We have the code:

```
t1=linspace(-0.6,1.3,20);
t2=linspace(-0.4,1.3,20);
function z=r(x, y)
    z=100.0*(y-x^2)^2 + (1-x)^2
z=feval(t1,t2,r);contour(t1,t2,z,40,flag=[2, 2 0]);
function z=g_r(x)
```

```
    z=[-2*(1-x(1))-400*x(1)*(x(2)-x(1)^2),200*(x(2)-x(1))]
function z=r(x)
    z=100.0*(x(2)-x(1)^2)^2 + (1-x(1))^2
maxiter=50;u=[-0.4 0.6];n=1;v=u;alpha=0.25;
for i=1:maxiter
    n=n+1;
    u=u-alpha*g_r(u)/norm(g_r(u));v=[v;u];
    alpha=0.25/log(n);
end
plot(v(:,1),v(:,2),'-');plot(1,1,'r*')
```

We get the next picture:

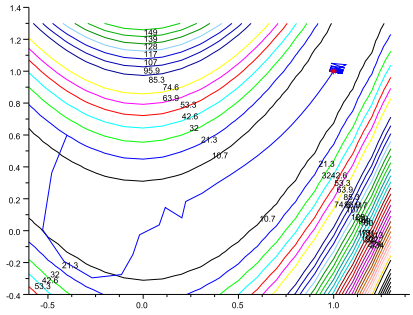

Figure 6.9: The descent method for Rosenbrock function.

Observe that the iterations oscillate around the point (1, 1) which, due to Exercise 7.51, is the minimum point of the function. A detail of the previous picture convinces us of this.

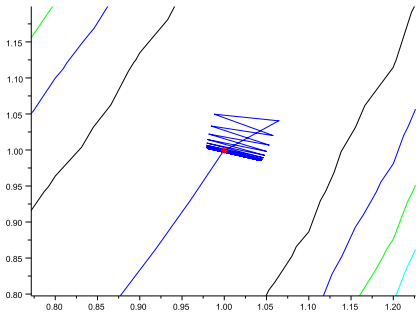

Figure 6.10: Detail: the oscillation of the iterations.

Supplemental details can be obtained by displaying at every step the value of the iteration, the distance to the minimum and the norm of the gradient.

16. (the QP method) We consider the algorithm for the function $f(x) = \frac{1}{2}\langle (Ax^t)^t, x\rangle + \langle b, x\rangle$ from Example 6.2.3. We saw that the speed of this algorithm depends of the ratio of the biggest and the smallest eigenvalue of A. We give a generic code for the method described in Example 6.2.3 for the situation of a matrix A of the form $\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$ (having hence the eigenvalues λ_1, λ_2), and for $b = 0$.

```
t1=linspace(-0.3,0.3,20);t2=linspace(-0.3,0.3,20);
a=20;b=1;
function z=r(x, y)
    z=a*x^2 + b*y^2
z=feval(t1,t2,r);
contour(t1,t2,z,10,flag=[2, 2 0]);
function z=g_r(x)
    z=[2*a*x(1),2*b*x(2)]
function z=r(x)
    z=a*x(1)^2 +b* x(2)^2
maxiter=10;u=[-0.1 0.3];v=u;
for i=1:maxiter
    u=u-g_r(u)*norm(g_r(u))^2/(2*r(g_r(u)));v=[v;u];
end
plot(v(:,1),v(:,2),'-o');plot(0,0,'r*')
```

In the case considered here we have $c = 20$, the level lines of the function being ellipses with relatively big eccentricity.

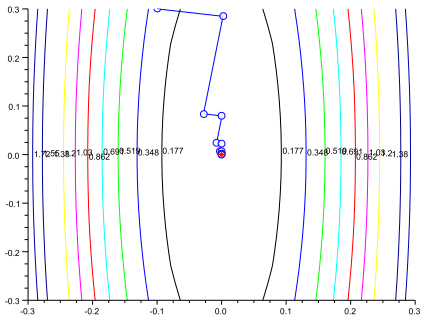

Figure 6.11: The QP method.

We obtain the same phenomenon of oscillation of the iterations, according to the next picture.

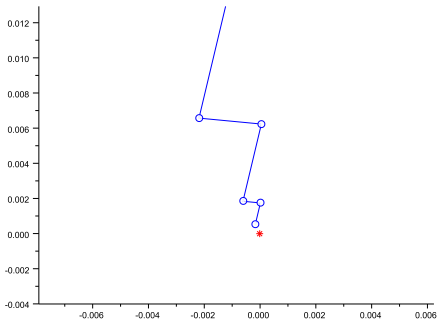

Figure 6.12: The QP method: detail.

17. (the SQP method) Consider the functions $f, h : \mathbb{R}^2 \to \mathbb{R}$ given by

$$f(x) = e^{x_1 x_2} - \frac{1}{2}\left(x_1^3 + x_2^3 + 1\right)^2, \quad h(x) = x_1^2 + x_2^2 - 5.$$

We implement the algorithm described in (6.2.8) to find the solution of the problem of minimizing f with the restriction $h(x) = 0$. Following the steps described at the SQP method, we generate the code:

```
    f='exp(x*y)-1/2*(x^3+y^3+1)^2'
    h='x^2+y^2-5'
    L='exp(x*y)-1/2*(x^3+y^3+1)+l*(x^2+y^2-5)'
    syms x y l
    df=[diff(f,x);diff(f,y)]
    dL=[diff(diff(L,x),x), diff(diff(L,x),y); diff(diff(L,x),y),
diff(diff(L,y),y)]
    A=[diff(h,x) diff(h,y)]
    At=[diff(h,x);diff(h,y)]
    F=[df-l*At;h]
    dF=[dL -At;A 0]
    x=1;y=2;l=1;
    v=[x;y;l];
    i=0;
    while(norm(eval(F))>10^-7)
     i=i+1
     v=v-inv(eval(dF))*eval(F)
```

```
 x=v(1);y=v(2);l=v(3);
end
v
```

Taking as initial value $(x_{10}, x_{20}, \mu_0) = (1, 2, 1)$, the code generates after 39 iterations the solution $(\overline{x}_1, \overline{x}_2) = (1.5811388, 1.5811388)$, $\overline{\mu} = -15.0304612$. Starting from $(x_{10}, x_{20}, \mu_0) = (-\sqrt{2}, \sqrt{3}, 1)$, the algorithm generates the same solution after 49 steps.

16. (the barrier method) Let $f, g : \mathbb{R}^2 \to \mathbb{R}$,

$$f(x) = x_1^2 + \frac{x_2^2}{3} + x_1x_2 + x_1,\ g(x) = x_1 + x_2 - 1.$$

Consider the optimization problem of minimizing f with the restriction $g(x) \leq 0$. We have shown that this problem has the solution $\overline{x} = (-2, 3)$. We verify that the minimal points of the barrier functions obtained for different values of the parameter μ approximate this solution. This can be done by considering the unconstrained minimization problems given by the barrier functions.
A selection of the obtained values in this way is given in the table below:

μ	1	0.0625	0.0123457	0.0004165	0.0001
$\overline{x}_1$	−1.2928932	−1.8232233	−1.9214326	−1.9855692	−1.9929289
$\overline{x}_2$	0.8786797	2.4696699	2.7642977	−0.9983152	2.9787868

Therefore, we have a good approximation of the actual solution since μ is small.